

CONDICIONAL: IF

PROBLEMA: A partir del monto de un crédito y la cantidad de cuotas se desea calcular el monto de cada cuota.

Datos de entrada y de salida?

Algoritmo ValorDeCuota
 DE: monto, cantCuotas (enteros)
 DS: cuota (real)

cuota ← monto/cantCuotas

CUIDADO

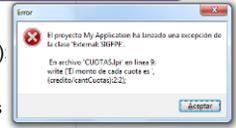
Necesitamos un condicional

CONDICIONAL: IF

```
program cuotas;
var credito: real;
    cantCuotas: integer;
begin
write ('Ingrese el monto del crédito ');
readln (credito);
write ('Ingrese la cantidad de cuotas
readln (cantCuotas);
write ('El monto de cada cuota es ', credito/cantCuotas);
end.
```

Si en cantCuotas se ingresa 0: **ERROR EN EJECUCION!**

Necesitamos un condicional



CONDICIONAL: IF

```
program cuotas;
var credito: real;
    cantCuotas: integer;
begin
...
```

Escribir una nueva versión donde quede almacenado en el dato "monto" el valor del dato de salida.

```
if (cantCuotas <= 0)
then
    writeln ('La cantidad de cuotas debe ser mayor a cero')
else
    write ('El monto de cada cuota es ', credito/cantCuotas);
end.
```

Evitamos la posibilidad de que surja un error en ejecución!

Es importante **validar** los datos para prevenir errores.

CONDICIONAL: CASE

Se utiliza cuando una o varias instrucciones **mutuamente excluyentes** dependen de una **expresión** de un tipo **escalar**.

Se evalúa la **expresión escalar** y de acuerdo al valor computado se selecciona la instrucción que se va a ejecutar.

Escalar: es un tipo no estructurado en el que está definida una relación de orden. *Integer* y *char* son escalares (también *boolean*).

CONDICIONAL: CASE

Escribir un segmento de programa que muestre el peaje a pagar según la cantidad de ruedas del vehículo:

	Peaje
• 2 ruedas	\$ 0
• 4 ruedas	\$ 5
• 6 ruedas	\$ 8
• 8 ruedas	\$ 12

- ¿Cuáles son los datos de entrada?
- ¿Cuáles son los datos de salida?
- ¿Cuáles pueden ser los casos de prueba?



```
readln(cantRuedas);
if (cantRuedas = 2)
then peaje := 0
else
if (cantRuedas = 4)
then peaje := 5
else
if (cantRuedas = 6)
then peaje := 8
else
if (cantRuedas = 8)
then peaje := 12;
```

CONDICIONAL: CASE

```

readln(cantRuedas);
case cantRuedas of
  2 : peaje := 0;
  4 : peaje := 5;
  6 : peaje := 8;
  8 : peaje := 12;
end;
writeln('Peaje ', peaje);

```

Todos del mismo tipo.



peaje puede quedar sin inicializar!

CONDICIONAL: CASE

Otra versión

```

program peajes;
var cantRuedas, peaje: integer;
begin
  writeln('Ingrese la cantidad de ruedas: ');
  readln(cantRuedas);

  if ((cantRuedas <> 2) and (cantRuedas <> 4) and (cantRuedas <> 6) and (cantRuedas <> 8))
  then
    writeln('La cantidad de ruedas no es valida.')
  else
    begin
      case cantRuedas of
        2 : peaje:=0;
        4 : peaje:=5;
        6 : peaje:=8;
        8 : peaje:=12
      end;
      writeln('El valor del peaje es: ', peaje);
    end;
  end;
end.

```

sentencia simple para el "then"

sentencia compuesta en el "else"

CONDICIONAL: IF

Escribir un programa que permita leer una cantidad de cuotas y mostrar el interés que se aplica.

- * si la cantidad de cuotas es 2 o 3 el interés es del 5%
- * si son entre 4 y 12 cuotas el interés es del 6%
- * si es otro valor mostrar un mensaje de error.

Proponga casos de prueba para verificar la solución.

CONDICIONAL: IF

Casos:

2 o 3 cuotas	Mostrar 5%
Entre 4 y 12 cuotas	Mostrar 6%
Otro valor	Mostrar mensaje de error

CONDICIONAL: IF

```

program interes;
var cuotas : integer;
begin
  write ('Ingrese la cantidad de cuotas: ');
  readln (cuotas);
  if ((cuotas = 2) or (cuotas = 3))
  then
    writeln ('El interés es del 5%')
  else
    if ((cuotas >= 4) and (cuotas <= 12))
    then
      writeln ('El interés es del 6%')
    else
      writeln ('Error en la cant. de cuotas!');
    end;
  end;
end.

```

Escribir una nueva versión donde quede almacenado en el dato "interes" el valor del dato de salida.

CONDICIONAL: IF

((cuotas = 2) or (cuotas = 3))

La expresión es verdadera si uno de los dos operandos computa true, es decir, la cantidad de cuotas es 2 o es 3.

((cuotas >= 4) and (cuotas <= 12))

La expresión es verdadera si ambos operadores relacionales computan true, es decir, si la cantidad de cuotas es 4, 5, 6, 7, 8, 9, 10, 11 o 12.

CONDICIONAL: CASE

Otra versión

```

program interes;
var cuotas : integer;
begin
  write ('Ingrese la cantidad de cuotas: ');
  readln (cuotas);

  case cuotas of
    2, 3:   writeln ('El interés es del 5%');
    4..12: writeln ('El interés es del 6%')

  else
    writeln ('Error en la cant. de cuotas!');
  end;
end.

```

Pascal permite establecer **subrangos de valores de tipo integer o char.**

Valor inicial .. Valor final

CONDICIONAL: IF

Una obra social cubre los gastos de farmacia de acuerdo a la categoría del afiliado

A → 80% cubierto por la obra social

B → 60% cubierto por la obra social

C → 50% cubierto por la obra social

Escriba un programa que lea el monto gastado por un afiliado en farmacia, su categoría y compute el monto a cargo de la obra social y el monto a cargo del afiliado.

CONDICIONAL: IF

El problema incluye tres datos constantes, cada uno correspondiente a los porcentajes de cobertura según la obra social.

Podemos declarar tres constantes A, B y C cada una de las cuales mantiene la cobertura de una categoría.

Dos datos de entrada:

la categoría del afiliado y el monto gastado.

Dos datos de salida:

el monto a cargo de la OS y el monto a cargo del afiliado.

CONDICIONAL: IF

```

program ObraSocial;
const  A = 0.80; B = 0.60; C = 0.50;
var    categoria: char;
       monto, porc, OS, Afil: real;
begin
  ...
end.

```

DATO AUXILIAR

CONDICIONAL: IF

```

program ObraSocial;
...
begin
  write ('Ingrese categoría y monto ');
  readln (categoria, monto);
  if (categoria = 'A')
  then porc := A
  else
    if (categoria = 'B')
    then porc := B;
    else porc := C;
  OS = monto * porc;
  Afil = monto * (1-porc);
  writeln ('A cargo OS: ', OS, ' Afiliado: ', Afil);
end.

```



Que pasa si en la variable *categoria* el usuario ingresa una 'Z'?

CONDICIONAL: IF

Los VALORES de las CONSTANTES A, B y C son números reales.

El condicional permite asignarle a la variable *PORC* el valor de la constante que corresponde de acuerdo al caso.

El VALOR de la VARIABLE *categoria* es un caracter 'A', 'B' o 'C'. Los caracteres se encierran entre apóstrofes.

Aunque se trata de un ejercicio simple, existen distintas alternativas para resolverlo.

A, B y C son nombres de constantes. 'A', 'B' y 'C' son valores de tipo CARACTER.

CONDICIONAL: IF

```

1 program OSociales;
2 const A = 0.80; B = 0.60; C = 0.50;
3 var categoria:char;
4     monto,OS, Afil: real;
5
6 Datos de entrada
7 begin
8     write ('Ingrese categoria y monto ');
9     readln (categoria, monto);
10    if (categoria = 'A')
11    then
12        begin
13            OS := monto * A;
14            Afil := monto * (1-A);
15        end
16    else
17        if (categoria = 'B')
18        then
19            begin
20                OS := monto * B;
21                Afil := monto * (1-B);
22            end
23        else
24            begin
25                OS := monto * C;
26                Afil := monto * (1-C);
27            end;
28
29    writeln ("A cargo OS: ", OS:5:2, " -- Afiliado: ", Afil:5:2);
30    readln();
31 end.
    
```



Otra versión

CONDICIONAL: IF

```

program ObraSocial;
var categoria: char;
...
begin
write ('Ingrese categoria y monto ');
readln (categoria, monto);
if ((categoria <> 'A') and
(categoria <> 'B') and
(categoria <> 'C'))
then
writeln ('Error en la categoria')
else
begin
...
end;
end.
    
```

Validación de la categoría

CONDICIONAL: IF

Si $((\text{categoria} \neq 'A') \text{ and } (\text{categoria} \neq 'B') \text{ and } (\text{categoria} \neq 'C'))$
 (es distinta de todas las validas)

not $((\text{categoria} = 'A') \text{ or } (\text{categoria} = 'B') \text{ or } (\text{categoria} = 'C'))$
 (no es ninguna de las validas)

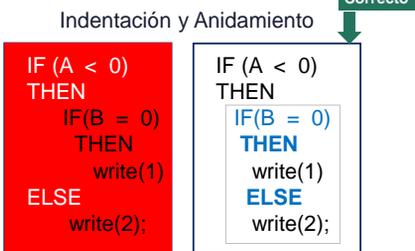
Detección de casos de CATEGORÍA INVÁLIDA (expresiones que son verdaderas si la categoría ingresada no es válida)

```

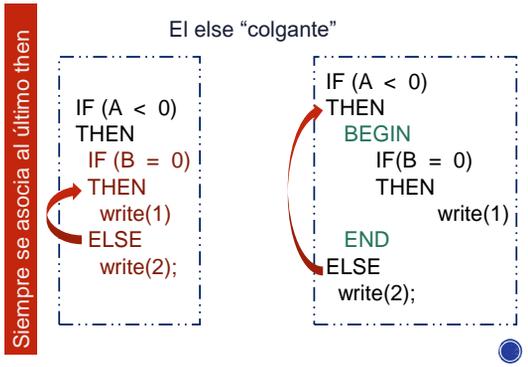
begin
write ('Ingrese categoria y monto ');
readln (categoria, monto);
case categoria of
'A': porc := A;
'B': porc := B;
'C': porc := C;
else
porc:=0
end;
if (porc=0)
then writeln('LA CATEGORIA NO ES VALIDA')
else
begin
OS = monto * porc;
Afil = monto * (1-porc);
writeln ('A cargo OS: ', OS, ' Afiliado: ', Afil);
end;
end.
    
```

Otra versión CASE

CONDICIONAL: IF



CONDICIONAL: IF



CUIDADO! La indentación afecta a la legibilidad pero **no modifica el significado del programa.**
 El else queda ligado SIEMPRE al último THEN

CONDICIONAL: CASE

Escribir un programa que lea un carácter y muestre un cartel indicando si se trata de una letra mayúscula, minúscula, un dígito u otro símbolo.

```

1 program caseSimb;
. var ch: char;
. begin
.   writeln('Ingrese un simbolo del teclado: ');
.   readln(ch);
.   case ch of
.     'a'..'z': writeln('Es una letra minuscula');
.     'A'..'Z': writeln('Es una letra mayuscula');
.     '0'..'9': writeln('Es un digito');
10  else writeln('Es un simbolo distinto de letra y digito');
.   end;
.   readln;
. end.
    
```

Pascal permite establecer subrangos de valores de tipo integer o char.

CONDICIONAL: CASE



PROBLEMA: Dada una nota numérica, mostrar la letra correspondiente.

```

program pepe; var x: integer; begin
writeln('Ingrese la calificacion del alumno: ');readln(x);
case x of 90..100: writeln('A');80..89: writeln('B'); 70..79:
writeln('C');60..69:writeln('D');0..59:writeln('E');end;readln;end.
    
```



CONDICIONAL: CASE



PROBLEMA: Dada una nota numérica, mostrar la letra correspondiente.

```

VAR Calificacion : Integer;

BEGIN
Write('Ingrese la Calificacion del alumno :');
Readln(Calificacion);
CASE Calificacion OF
  90..100 : Writeln('A');
  80..89  : Writeln('B');
  70..79  : Writeln('C');
  60..69  : Writeln('D');
  0..59   : Writeln('E')
END;
END.
    
```



Claridad en nuestro código

- ✖ Es importante que un algoritmo (y luego el programa) sea claro, y que no presente ambigüedades para quien lo lea.
- ✖ Para lograr esto hay que tener en cuenta una serie de pautas de "buen estilo" para la realización de algoritmos:

- Líneas para agrupar en bloques
- Indentación
- Uso de nombres representativos



*Para estudiar
en casa*



CONDICIONAL: IF

Otro ejemplo

Un comercio aplica un descuento del 10% a las ventas que superan los \$250. Escribir un programa que lea el monto bruto de una venta y muestre el monto luego de aplicar el descuento correspondiente.

Monto Bruto	Monto final (neto)
<= 250	Monto Bruto
> 250	90% de Monto Bruto

¿Cuáles son los datos de entrada?

¿Cuáles son los datos de salida?



CONDICIONAL: IF

Algoritmo venta
DE: bruto
DS: neto
COMIENZO

SI (bruto <=250)
ENTONCES

neto := bruto

SINO

neto := bruto * 0.9;

FIN

CONDICIONAL: IF

```
program venta;
const porc = 0.9;
var bruto, neto: real; // el neto es el bruto * porcentaje
begin
  write ('Ingrese el Monto Bruto: ');
  readln (bruto);
  if (bruto <=250)
  then neto := bruto
  else neto := bruto * porc;
  writeln ('El monto neto es ', neto);
end.
```

CONDICIONAL: CASE

Categorías a las que pertenecen los signos del zodiaco:

Signo	Categoría
1. Aries	Fuego
2. Tauro	Tierra
3. Géminis	Aire
4. Cáncer	Agua
5. Leo	Fuego
6. Virgo	Tierra
7. Libra	Aire
8. Escorpio	Agua
9. Sagitario	Fuego
10. Capricornio	Tierra
11. Acuario	Aire
12. Piscis	Agua

Se quiere escribir un programa que:

- 1) Muestre el listado de los signos con un número y categoría asociados.
- 2) Pida por teclado un número (dato entero) asociado a un signo.
- 3) Muestre por pantalla la categoría a la que pertenece el signo del zodiaco seleccionado.

Nota: Si el número introducido por el usuario, no está asociado a ningún signo del zodiaco, se mostrará el mensaje:

"ERROR: <núm> no está asociado a ningún signo."



CONDICIONAL: CASE

En la pantalla:

```
Listado de signos del zodiaco:
1. Aries
2. Tauro
3. Géminis
4. Cáncer
5. Leo
6. Virgo
7. Libra
8. Escorpio
9. Sagitario
10. Capricornio
12. Acuario
13. Piscis

Introduzca numero de signo: 7
Es un signo de Aire.
```

```
Listado de signos del zodiaco:
1. Aries
2. Tauro
3. Géminis
4. Cáncer
5. Leo
6. Virgo
7. Libra
8. Escorpio
9. Sagitario
10. Capricornio
12. Acuario
13. Piscis

Introduzca numero de signo: 15
ERROR: 15 no esta asociado a ningún signo.
```



Ejercicio

Escriba un programa que lea una letra y muestre la que sigue en forma circular.

Es decir, la que sigue a la 'A' es la 'B', la que sigue a la 'Z' es la 'A'.

Si el usuario ingresa un carácter que no es una letra el programa debe mostrar un mensaje de error.

Elija casos de prueba adecuados para verificar su solución.



Ejercicio

Escribir un programa en Pascal que dada una calificación en valor alfabético (A, B, C, D o E) indique su equivalente en valor numérico (en este caso será 10, 9, 8, 7 o 6).

Use un condicional múltiple (CASE).

