

Tarea: Técnicas de Programación y pseudocódigo

Captulo II: Selección



CFPI: A Carballeira

Módulo: Programación

Contenido

Organización selectiva o decisión.....	3
Condición Si-Entonces (if)	3
Condición Si-entonces-sino (if-else).....	4
Practicando.....	5
Buenas prácticas en el uso de SI: Descuentos de un almacén	10
Buenas prácticas en el uso de SI: Condicionales con varias preguntas.....	12
Condicionales con casos o en cascada	14
Ejercicios empleando SI.....	17
Selección Múltiple Según (switch).....	20
Buenas prácticas en el uso de SEGÚN.....	21
Ejemplo - Selección de vocales	21
Ejercicios estructura SEGUN	23
Anexo: Uso de operadores lógicos	25
Utilizando lógica "positiva"	25
Utilizando lógica "negativa"	26
Entonces surge la pregunta: ¿Cual hay que usar?	27

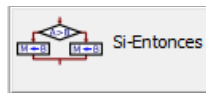
Diagramas de flujo y pseudocódigo

Organización selectiva o decisión

Hasta ahora vimos solo código que se ejecutaba de manera Secuencial... es decir no había ninguna Bifurcación, por lo tanto solo existía un camino.

Algunas veces es necesario establecer una condición en la resolución de un problema. Esa condición después de ser evaluada divide la resolución del problema en dos posibles vías o caminos:

- a) si el resultado de la condición es verdadero, se toma el camino del verdadero,
- b) si dicho resultado es falso, se toma el camino del falso.



Condición Si-Entonces (if)

La ejecución de estas sentencias depende del valor de la condición lógica. Al evaluar la expresión lógica, devuelve un Verdadero o un Falso. Si la expresión lógica es Verdadera se realizan las instrucciones que están bajo la indicación. La sintaxis sería:

```
Si <expresión que se evalúa> Entonces
instrucciones..
....
FinSi
```

Veamos un Ejemplo:

Proceso if0

definir a, b como entero;

leer a;

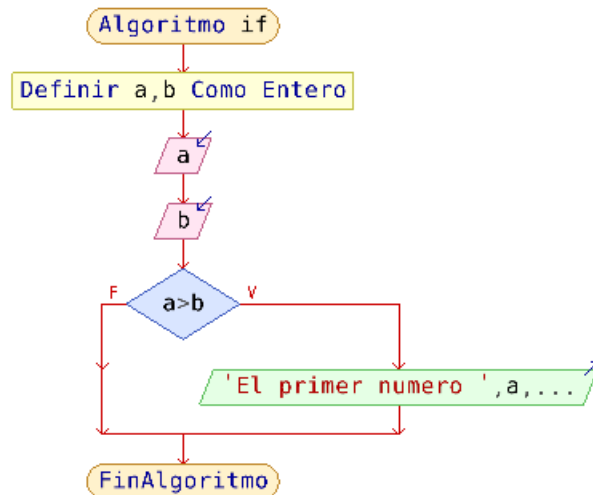
leer b;

Si a>b Entonces

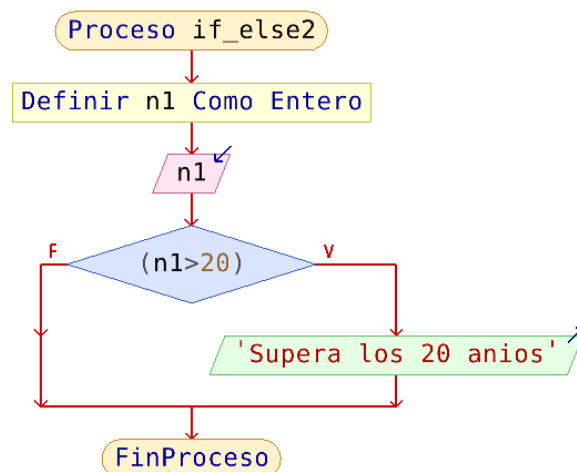
Escribir "El primer número ", a, " es mayor que el Segundo ", b, " ingresado";

FinSi

FinProceso



Ejemplo: Dada la edad de un alumno mostrar con cartel aclaratorio si la misma supera los 20 años.

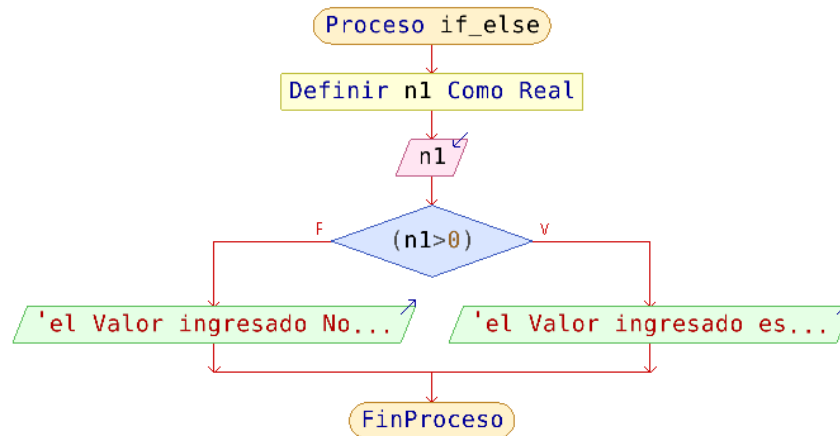


Condición Si-entonces-sino (if-else)

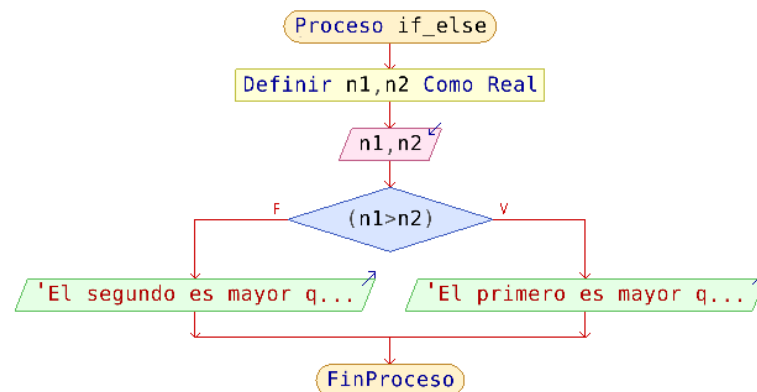
Este tipo de estructura organiza la ejecución de las acciones después de la evaluación de una expresión.

Los resultados posibles de la expresión establecida podrán ser: **verdadero (V) ó falso (F)**, por lo tanto las acciones siguientes a ser ejecutadas tienen **dos posibles caminos**, **PERO SIEMPRE debe existir una acción al menos por el verdadero**.

- Si la expresión es verdadera se ejecutarán la/s acciones que están en el camino del verdadero
- Si la expresión es falsa se ejecutarán la/s acciones que están en el camino del falso.



Ejemplo: Dados dos números reales distintos mostrar el mayor.



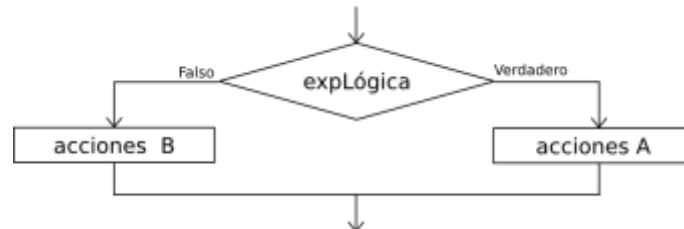
Practicando...

En casi cualquier secuencia de instrucciones para un ordenador, será vital poder comprobar si se cumple alguna condición. Una primera forma básica de comprobar condiciones es con la orden "Si". Su uso básico sería

```
Si condicion Entonces
    pasos_a_dar_si_es_verdadero
SiNo
    pasos_a_dar_si_es_falso
FinSi
```

El bloque "SiNo" es opcional: podemos optar por no indicar lo que queremos que se haga cuando no se cumpla la condición.

Para ayudarnos a planificar el comportamiento de una secuencia de instrucciones, se suele usar como ayuda los llamados "diagramas de flujo". En estos diagramas, una condición se representa como un rombo, del que salen dos flechas: una para la secuencia de acciones a realizar si se cumple la condición y otra para cuando no se cumple:



Así, si en PSeInt hacemos clic en el icono de la parte derecha que representa la condición SI-ENTONCES, aparece un esqueleto de programa casi completo, para que hagamos los cambios que nos interesen:

PSeInt

Archivo Editar Configurar Ejecutar Ayuda

Ejemplo004.psc

```
1 Algoritmo Ejemplo004
2
3   Escribir "Dime un numero"
4   Leer primerNumero
5   Escribir "Dime otro numero"
6   Leer segundoNumero
7
8   Si expresion_logica Entonces
9       acciones_por_verdadero
10
11   SiNo
12       acciones_por_falso
13   Fin Si
14
15 FinAlgoritmo
```

Comandos

- Escribir: "Hola !"
- Leer: Dato1
- Asignar: A ← B + 1
- Si-Entonces: M ← A, N ← B
- Según: M ← A, N ← B, O ← C
- Mientras: M ← A, N ← B
- Repetir: M ← A, N ← B
- Para: M ← A, N ← B
- Función: y ← f(x)

Ayuda Rápida

SI - ENTONCES - SINO

- {expresion_logica}: complete aquí la condición a evaluar para decidir que grupo de acciones se deben ejecutar.
- {acciones_por_verdadero}: complete aquí las acciones que se deben ejecutar si la condición ({expresion_logica}) resulta verdadera.

El pseudocódigo es correcto. Presione F9 para ejecutarlo.

Sólo tenemos que escribir la condición que realmente nos interesa, y la serie de pasos que se deben dar si se cumple y si no se cumple, de modo que nuestro programa podría quedar así:

Algoritmo Ejemplo004

```
Escribir "Dime un numero"

Leer primerNumero

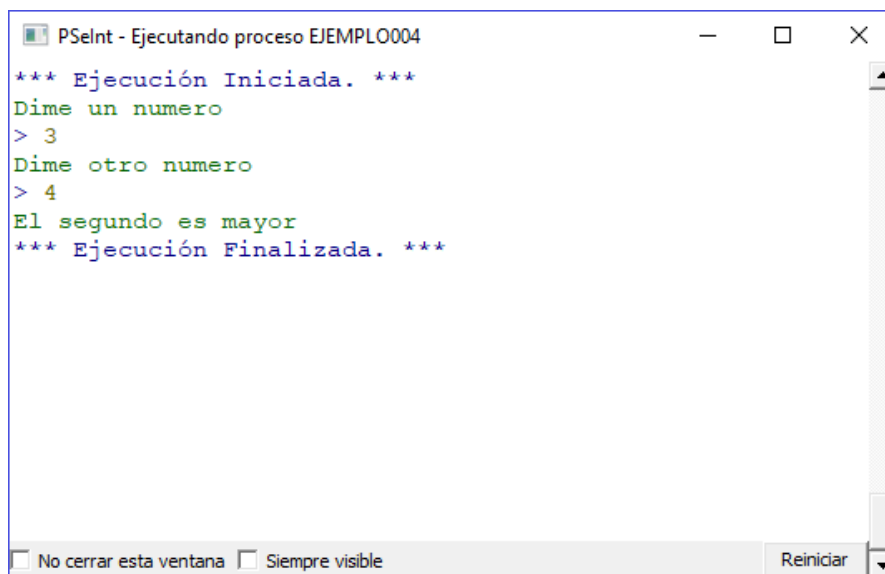
Escribir "Dime otro numero"

Leer segundoNumero

Si primerNumero > segundoNumero Entonces
    Escribir "El primero es mayor"
Sino
    Escribir "El segundo es mayor"
FinSi

FinAlgoritmo
```

Su resultado sería éste:



```
PSeInt - Ejecutando proceso EJEMPLO004
*** Ejecución Iniciada. ***
Dime un numero
> 3
Dime otro numero
> 4
El segundo es mayor
*** Ejecución Finalizada. ***

☐ No cerrar esta ventana ☐ Siempre visible 
```

(Nota: si lo piensas con un poco de detenimiento -o lo pruebas-, te darás cuenta de que ese planteamiento no es correcto del todo: si introduces dos números iguales, te dirá que el segundo es el mayor; lo solucionaremos un poco más adelante, en el ejemplo 4c).

En ese ejemplo hemos comparado si un valor es mayor que el otro (>). Los operadores de comparación que tenemos disponibles son:

Operador relacional	Significado	Ejemplo
>	Mayor que	3>2
<	Menor que	2<3
=	Igual que	3=3
<=	Menor o igual que	5<=5
>=	Mayor o igual que	6>=5

Ejercicio de repaso propuesto 4.1: Crea un programa que pida dos números al usuario y responda si son iguales o no lo son.

También es posible comprobar varias condiciones a la vez, para hacer construcciones más complejas, como: "si a es mayor que b y b es mayor que c", o como "si a es igual a 1 o b es igual a 1 ". Los operadores lógicos que de los que disponemos son:

Operador lógico	Significado	Ejemplo
& ó Y	Conjunción (y).	(7>4) & (2=1) //falso
ó O	Disyunción (o).	(1=1 2=1) //verdadero
~ ó NO	Negación (no).	~(2<5) //falso

Un ejemplo de su uso sería:

Algoritmo Ejemplo004b

```
Escribir "Dime un numero"
```

```
Leer primerNumero
```

```
Si primerNumero > 0   o   primerNumero = 0 Entonces
```

```
    Escribir "Es mayor o igual que cero"
```



```
Sino
    Escribir "Es negativo"
FinSi

FinAlgoritmo
```

Además, podemos encadenar varias condiciones una detrás de otra. Como hemos comentado, el ejemplo 4 fallaba si los dos números son iguales. Por eso, un planteamiento más detallado (y más correcto) sería emplear dos comprobaciones "si", de la siguiente manera:

```
Algoritmo Ejemplo004c

    Escribir "Dime un numero"
    Leer primerNumero
    Escribir "Dime otro numero"
    Leer segundoNumero

    Si primerNumero > segundoNumero Entonces
        Escribir "El primero es mayor"
    Sino
        Si primerNumero < segundoNumero Entonces
            Escribir "El segundo es mayor"
        Sino
            Escribir "Son iguales"
        FinSi
    FinSi

FinAlgoritmo
```

Ejercicio de repaso propuesto 4.2: Haz una variante del ejemplo 004b, usando el operador " \geq ".

Ejercicio de repaso propuesto 4.3: Crea un programa que pida un número al usuario y diga si es positivo, negativo o cero.

Ejercicio de repaso propuesto 4.4: Haz un programa que pida al usuario dos números y diga cuántos de ellos son positivos.

Ejercicio de repaso propuesto 4.5: Crea un programa que pida al usuario dos números y muestre su división si el segundo no es cero, o un mensaje de aviso en caso contrario.

Ejercicio de repaso propuesto 4.6: Prepara un programa que pida al usuario tres números y diga cuál es el mayor de los tres.

Buenas prácticas en el uso de SI: Descuentos de un almacén

Un almacén aplica un descuento del 10% a quienes han comprado mas de 100 euros en productos.

Realice un algoritmo que permita realizar esta operación directamente en la caja registradora.

Desarrollo:

Proceso ofertas01

Definir valor,descuento,pagar **Como Real**;

Leer valor;

Si valor>100 **Entonces**

 descuento = 0.10;

SiNo

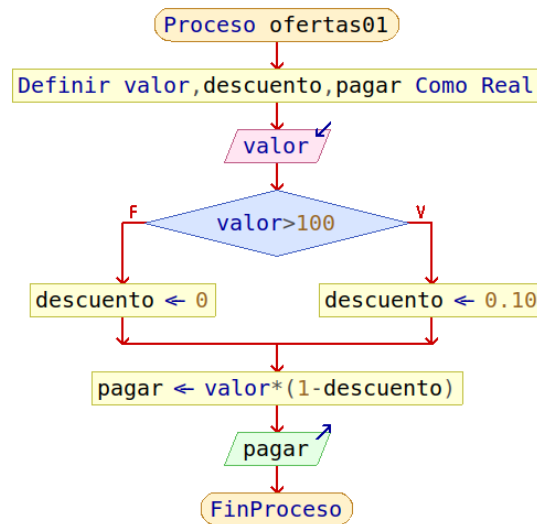
 descuento = 0;

FinSi

pagar = valor*(1-descuento);

Escribir pagar;

FinProceso



Mejor solución:

Otra forma de expresar el algoritmo es inicializar a 0 el descuento y solo modificarlo si se cumple con la condición de que el cliente adquiriera un valor superior a 100 dólares:

Proceso ofertas02

Definir valor, descuento, pagar **Como Real**;

descuento = 0;

Leer valor;

Si valor > 100 **Entonces**

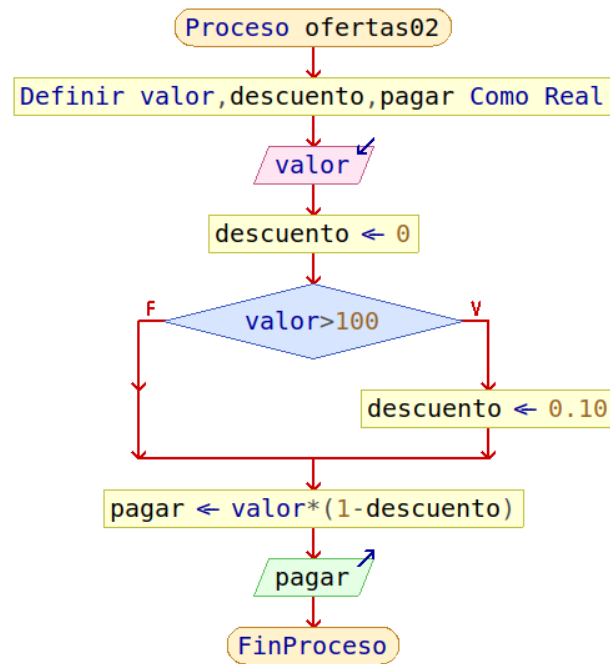
 descuento = 0.10;

FinSi

pagar = valor * (1 - descuento);

Escribir pagar;

FinProceso



Importante: Cuando la condicional modifique el valor de una o varias variables solamente en caso de que se cumpla la condición, la acción se debe ejecutar por el lado Verdadero.

Debe evitarse plantear condiciones lógicas que salgan solamente por el Falso y no tengan acción alguna por el camino Verdadero.

Buenas prácticas en el uso de SI: Condicionales con varias preguntas

Cuando se debe incluir más de una pregunta en un condicional, se utilizan los operadores lógicos.

En diagramas de flujo o pseudocódigo, las expresiones usan las mismas conectivas y que la notación de lógica matemática.

Algunos preeren usar la notación de lenguajes estructurados de programación y otros utilizan la forma simple: “Y”, “O”.

Conectiva	Símbolo	Expresión	Operador
Conjunción	^, &&	Y	AND
Disyunción	v,	O	OR
Negación	!	NO	NOT

Ejemplo: Descuento en Almacén con 2 condiciones

Un almacén aplica el descuento del 10 % para compras mayores a 8 euros y que incluyan más de 10 artículos.

Desarrollo:

Para el ejercicio se requieren dos variables de ingreso: valor y cantidad.

El descuento se aplica con la lógica $(valor > 8)$ y $(cantidad > 10)$, que se traduce en la siguiente expresión: $(valor > 8) \ \&\& \ (cantidad > 10)$.

Para el procedimiento se continúa de igual forma que en el ejercicio anterior:

Proceso ejemplo02

Definir valor,descuento,pagar **Como Real**;

Definir cantidad **Como Entero**;

Leer valor;

Leer cantidad;

Si $(valor > 8)$ **Y** $(cantidad > 10)$ **Entonces**

 descuento = 0.10;

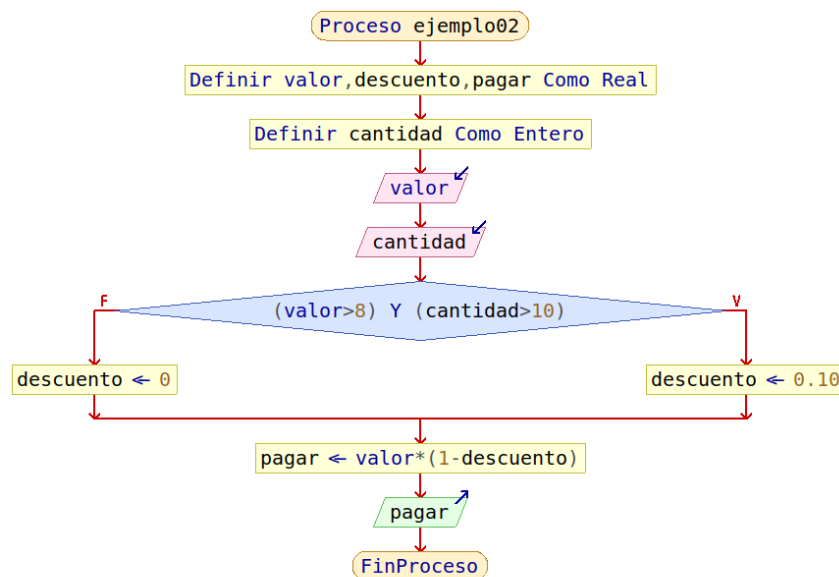
SiNo

 descuento = 0;

FinSi

pagar = $valor * (1 - descuento)$;

Escribir pagar;

FinProceso

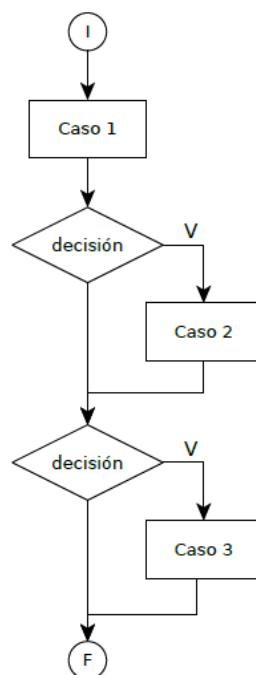
Pregunta: ¿Cómo mejorarías este algoritmo?... Inicializando descuento al principio.

Condicionales con casos o en cascada

Existen problemas en los que una variable puede tomar diferentes valores conforme al cumplimiento de diferentes casos.

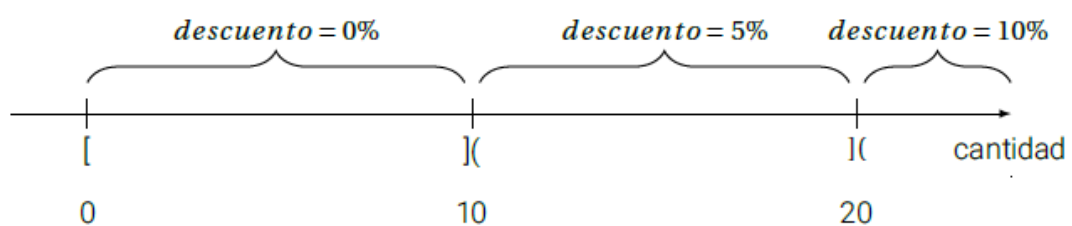
Por ejemplo, cuando se realizan ventas por volumen (por cantidades grandes de producto), el descuento depende de la cantidad del producto que se vende.

Cuando **el condicional puede tener varios valores**, se puede abordar el problema, dividiendo la condición en casos, donde cada caso es una pregunta. Se **empieza por el caso más simple y se modifica, va analizando las condiciones utilizando preguntas en cascada**:



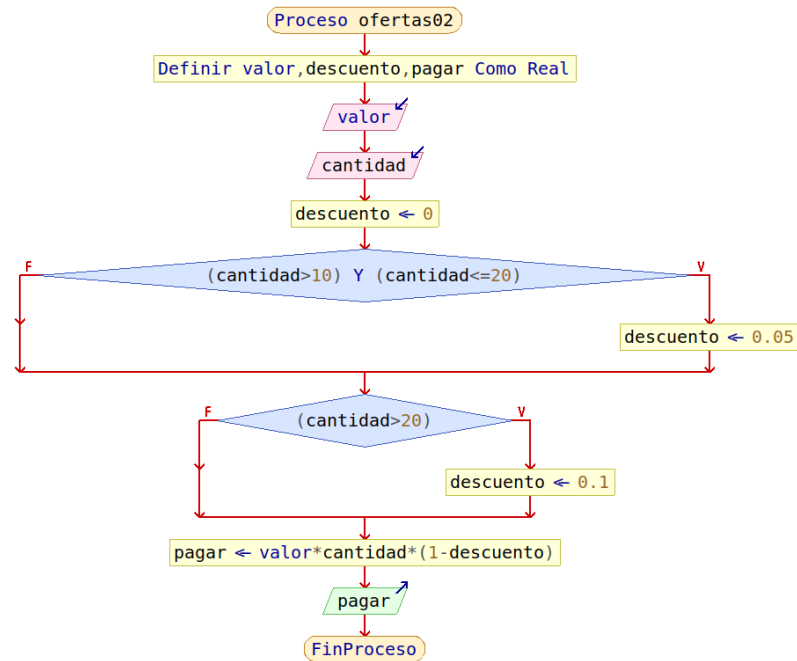
Ejemplo: Descuento en Almacén con más de 2 condiciones

Un almacén rebaja 10% del precio de un producto si se adquieren más de 20 unidades y 5% si adquieren entre 10 y 20 unidades (inclusive), no hay descuento para cantidades menores o iguales a 10 unidades. Con el precio unitario del producto y la cantidad adquirida, realice un algoritmo para mostrar el valor a pagar



Desarrollo:

Una forma de simplificar la resolución del problema, es dividirlo en casos. Siguiendo la lógica que se plantea en la figura, el programa calcula el descuento en función de la cantidad de artículos. De este modo, la acción de "Aplicar los descuentos" se traduce en evaluar cada rango de cantidad y escribirlos uno a uno como condicionales.

**Proceso** ofertas02

Definir valor,descuento,pagar **Como Real**;

Leer valor;

Leer cantidad;

descuento <- 0;

Si (cantidad>10) **Y** (cantidad<=20) **Entonces**

descuento <- 0.05;

FinSi

Si (cantidad>20) **Entonces**

descuento <- 0.1;

FinSi

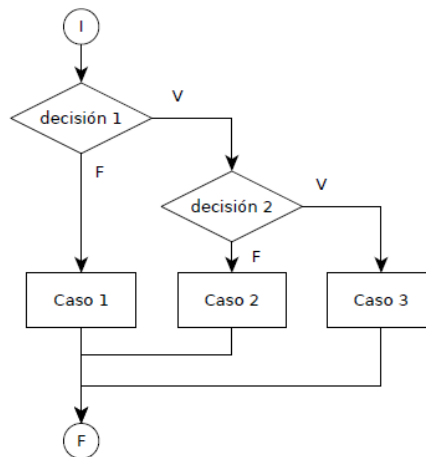
pagar <- valor*cantidad*(1-descuento);

Escribir pagar;

FinProceso

Mejor solución

Otra forma de resolver el problema del ejemplo anterior, consiste en generar un camino por cada caso para asignar el valor a la variable de salida. De esta forma, cada camino planteado es parte de un condicional.



Como se puede observar en el gráfico en el mejor de los casos sólo evaluamos una opción, sin embargo con la solución anterior independientemente del caso todas las condiciones eran evaluadas.

Proceso ofertas02

Definir valor,descuento,pagar,cantidad **Como Real**;

Leer valor;

Leer cantidad;

descuento = 0;

Si (cantidad>10) **Entonces**

Si (cantidad>20) **Entonces** descuento = 0.1;

SiNo descuento = 0.05;

FinSi

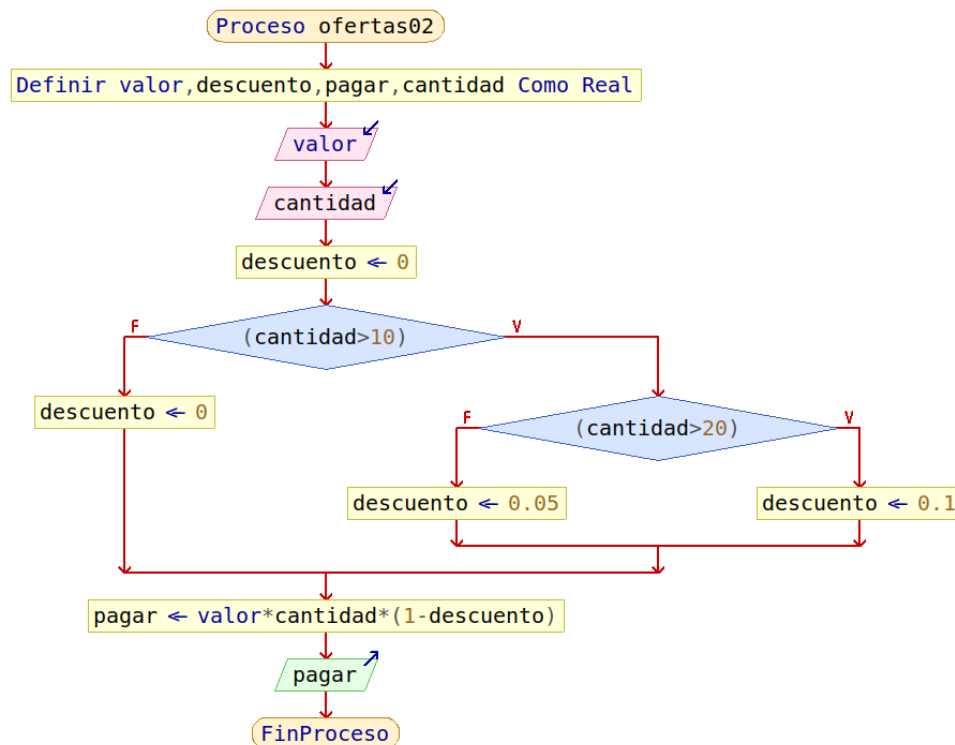
SiNo descuento = 0;

FinSi

pagar = valor*cantidad*(1-descuento);

Escribir pagar;

FinProceso



Condicionales con casos vs. en árbol: Conclusión

Comparando los diagramas de flujo, es posible notar que el hilo del programa es diferente en ambas soluciones. El programa hace lo mismo, pero de diferente manera. En un caso las preguntas se realizan una a continuación de otra, siempre. En el condicional por árbol, solo se realizan las preguntas necesarias para el cada caso.

Ejercicios empleando SI...:

1. Escriba un algoritmo que determine si un número ingresado es positivo, negativo o cero.
2. Escriba un algoritmo que permita leer dos valores numéricos A y B. Si A es mayor que B debe realizar la suma de ambos, caso contrario, se hace el producto.
3. Escriba un algoritmo que permita ingresar valores numéricos A, B, C, D, E y decir si su promedio es mayor o igual a 10.
4. Hacer un programa que permita ingresar dos números y el símbolo de la operación (+, -, *, /). obteniéndose el correspondiente resultado. Si el símbolo no es correcto deberá imprimir un mensaje que indique "Error en símbolo".
5. Escriba un algoritmo capaz de encontrar el máximo de dos valores (M y N)
6. Escriba un algoritmo capaz de encontrar el máximo de tres valores (R, S y T).

7. Escribir un algoritmo que permita ingresar tres números distintos entre sí y los muestre en pantalla ordenados de mayor a menor y de menor a mayor.
8. Se ingresa por teclado un número positivo de uno o dos dígitos (1..99). Mostrar un mensaje indicando cuantos dígitos tiene el número ingresado.
9. Se tiene como datos 4 valores numéricos enteros distintos. Calcular y mostrar la suma del mayor número y el menor número de los 4 dados.
10. Desarrollar un algoritmo que permita realizar la suma de dos horas distintas (HH, MM, SS). Tener en cuenta que si la hora supera 24 se debe contar un día.
11. Leer los lados de un rectángulo y el radio de un círculo. Determinar si las áreas de ambas figuras son o no iguales.
12. Diseñe un algoritmo que permita ingresar la hora actual del día en tres variables, HH, MM y SS y determine cuantas horas, minutos y segundos restan para las 23:59:00.
13. Diseñe un algoritmo que permita ingresar dos valores X e Y, luego determine qué porcentaje es X de Y.
14. Un año es bisiesto si es divisible por 4 y no es por 100, o si es divisible por 400. Desarrolle un algoritmo donde se lea o ingrese desde la entrada estándar un valor entero correspondiente a un año cualquiera y determine si es o fue bisiesto.
15. Leer la nota de un alumno (numérica) y mostrar un mensaje diciendo si la calificación es suspenso (<5), aprobado (≥ 5 y < 7), bien (≥ 7 y < 9), notable (≥ 9 y < 10) o sobresaliente ($= 10$). Mostrar un mensaje si la nota es incorrecta.
16. El promedio de prácticas de un curso se calcula en base a cuatro notas de las cuales se elimina la nota menor y se promedian las tres notas más altas. Diseñe un algoritmo que determine la nota eliminada y el promedio de un estudiante.
17. Diseñe un algoritmo que lea tres longitudes y determine si forman o no un triángulo. Si es un triángulo determine de que tipo de triángulo se trata entre: equilátero (si tiene tres lados iguales), isósceles (si tiene dos lados iguales) o escaleno (si tiene tres lados desiguales). Considere que para formar un triángulo se requiere que: "el lado mayor sea menor que la suma de los otros dos lados".
18. Ingresar un número C y determinar si pertenece al intervalo [A, B] con A y B ingresados por teclado.
19. Una compañía dedicada al alquiler de automóviles cobra \$30 hasta un máximo de 300 km de distancia recorrida. Para más de 300 km y hasta 1000 km, cobra \$30 más un monto adicional de \$ 0.15 por cada kilómetro en exceso sobre 300. Para más de 1000 km cobra \$30 más un monto adicional de \$ 0.10 por cada kilómetro en exceso sobre 1000. Los precios ya incluyen el 18% del impuesto general a las ventas, IGV. Diseñe un

algoritmo que, ingresando la distancia recorrida, determine el monto a pagar por el alquiler de un vehículo y el monto incluido del impuesto. Basándose en el año de fabricación y el peso del automóvil en una fábrica se determina la tarifa de registro según la siguiente tabla:

Año (modelo)	Peso (lb.)	Categoría de Peso	Tarifa de Registro
1970 o anterior	Menos de 2.700	1	\$ 11.600
	2.700 a 3.800	2	\$ 23.200
	Más de 3.800	3	\$ 34.800
1971 a 1979	Menos de 2.700	4	\$ 13.000
	2.700 a 3.800	5	\$ 26.000
	Más de 3.800	6	\$ 39.000
1980 o posterior	Menos de 3.500	7	\$ 12.000
	3.500 o más	8	\$ 46.000

El programa debe ingresar el año y el peso del auto e imprimir la categoría y la tarifa.

20. El cuadrante en el cual se ubica una línea dibujada desde el origen está determinado por el ángulo que la línea forma con el eje positivo de x en la siguiente forma:

Ángulo del eje positivo de x	Cuadrante
Entre 0 y 90 grados	I
Entre 90 y 180 grados	II
Entre 180 y 270 grados	III
Entre 270 y 360 grados	IV

El programa debe ingresar el ángulo de la línea como dato de entrada y desplegar el cuadrante apropiado, según la tabla dada. Si el ángulo es exactamente 0, 90, 180 o 270 grados, la línea resultante no se ubica en ningún cuadrante sino en cada uno de los ejes respectivos.

21. Dados dos números reales mostrar el mayor.
22. Dados cuatro números reales mostrar el mayor.
23. Teniendo como dato una temperatura en grados Celsius, indicar el estado físico del agua a esa temperatura y a 760 mm de presión atmosférica: sólido (menor ó igual a 0 °C), líquido (mayor a 0 °C y menor ó igual a 100 °C) y gaseoso (mayor a 100 °C).
24. Siendo los datos tres números enteros, determinar si la suma de cualquier pareja de ellos es igual al número restante. Si se cumple esta condición, escribir "Iguales", en caso contrario escribir "Distintos".
25. Dados tres números enteros distintos ingresados por teclado, determinar y mostrar el número que ocupa la posición del centro, al ordenar los mismos de menor a mayor.
26. Siendo los datos tres enteros válidos que representan una fecha: dd, mm, aa, se pide obtener y mostrar la fecha del día siguiente.
27. Ingresar un número de mes válido y un año. Mostrar la cantidad de días que puede tener el mismo, considerando que el año puede ser bisiesto.

Selección Múltiple Según (switch)

La ejecución de esta sentencia depende del valor de una variable, que va seguido del comando Segun, esto evalúa la variable y analiza con FinSegun.

Esta instrucción permite ejecutar opcionalmente varias acciones posibles, al ejecutarse, se evalúa el contenido de la variable y se ejecuta la secuencia de instrucciones asociada con dicho valor que se detallan luego del comando Hacer.

Es importante destacar que NO HAY COMPARACIÓN u EXPRESIÓN DE RELACIÓN, dentro luego de Segun, si no que solamente evalúa una variable.

Cada opción está formada por uno o más constantes separados por comas, dos puntos y una secuencia de instrucciones.

Si una opción incluye varios números, la secuencia de instrucciones asociada se debe ejecutar cuando el valor de la variable es uno de esos números.

Se debe agregar una opción final, denominada De Otro Modo, cuya secuencia de instrucciones asociada se ejecutará sólo si el valor almacenado en la variable no coincide con ninguna de las opciones anteriores.



Esta sentencia se utiliza normalmente para la creación de Menús.

Proceso switch

Definir a Como Caracter;

Escribir "Ingrese una opción: ";

Escribir "1-Primer Alternativa";

Escribir "2-Segunda Alternativa";

Escribir "S-Salir";

Leer a;

Segun(a)

'1': Escribir "Ud. eligió 1er. opción";

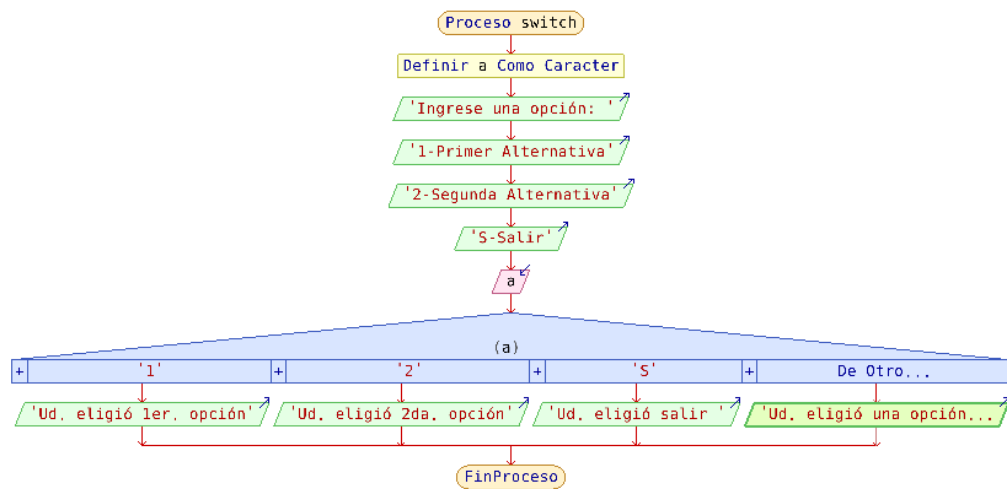
'2': Escribir "Ud. eligió 2da. opción";

'S': Escribir "Ud. eligió salir ";

De otro Modo: Escribir "Ud. eligió una opción no valida";

FinSegun

FinProceso



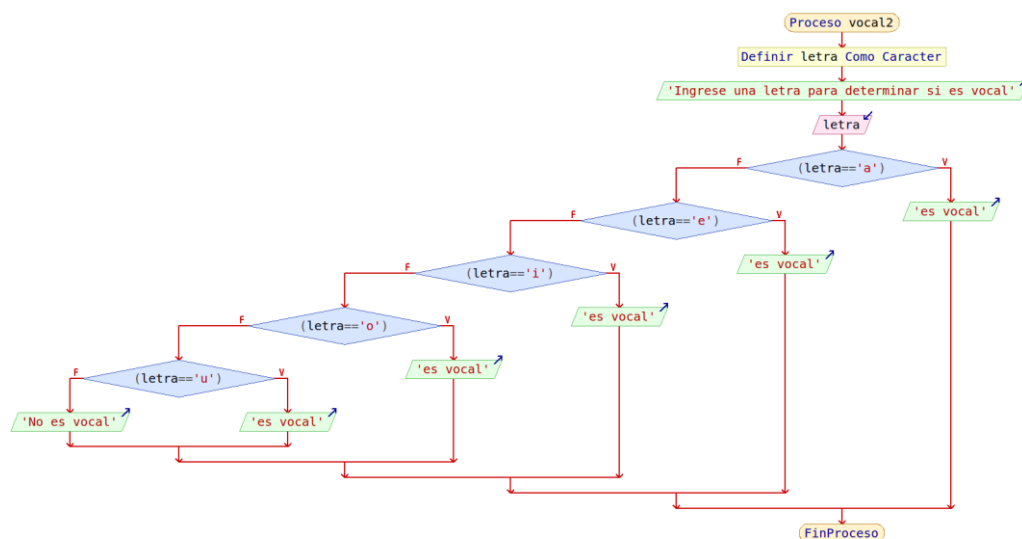
Buenas prácticas en el uso de SEGÚN

Las estructuras de selección múltiple se utilizan para realizar selecciones entre mas de dos opciones, aliviando la necesidad realizar condiciones múltiples dentro de una misma pregunta. A continuación se presentan ejemplos que muestran los diferentes escenarios

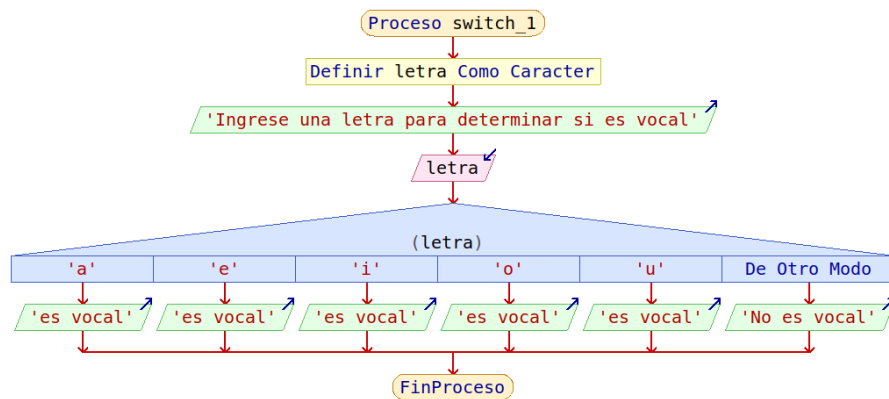
Ejemplo - Selección de vocales

En este ejemplo se muestra un programa que permite ingresar una letra por teclado e identificar las vocales ingresadas

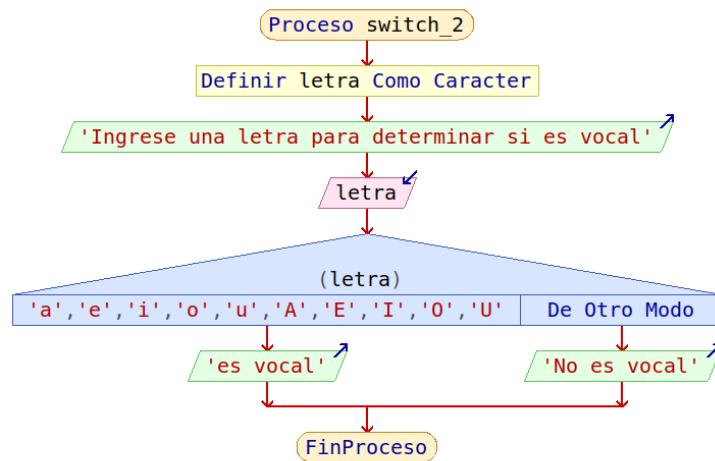
Solución 1:



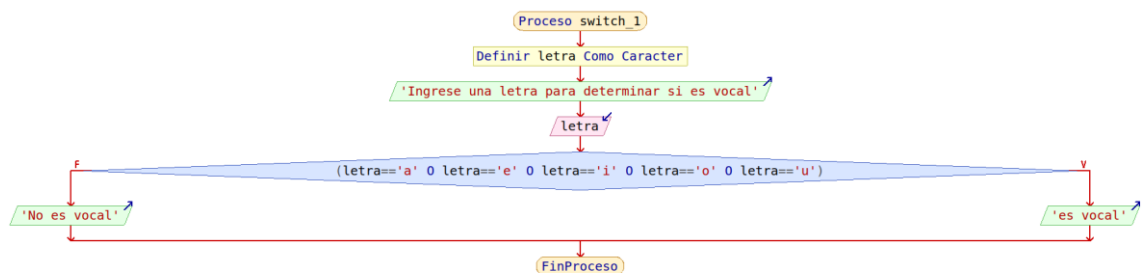
Solución 2



Solución 3:



Solución 4:



¿Cuál crees que es mejor?... Las dos últimas son equivalentes aunque cuando hay pocas ramas suele emplearse el SI. Es por eso que el SEGÚN se emplea casi de forma exclusiva en menús

Solución 3:

Proceso vocal1

Definir letra **Como Caracter**;

Escribir "Ingrese una letra para determinar si es vocal";

Leer letra;

Si (letra=="a" **O** letra=="e" **O** letra=="i" **O** letra=="o" **O** letra=="u")

Entonces Escribir "es vocal";

SiNo

Escribir "No es vocal";

FinSi

FinProcesoProceso vocal1

Solución4:

Proceso switch_2

Definir letra **Como Caracter**;

Escribir "Ingrese una letra para determinar si es vocal";

Leer letra;

Segun (letra) **Hacer**

"a","e","i","o","u","A","E","I","O","U": **Escribir** "es vocal";

De Otro Modo: **Escribir** "No es vocal";

FinSegun

FinProceso

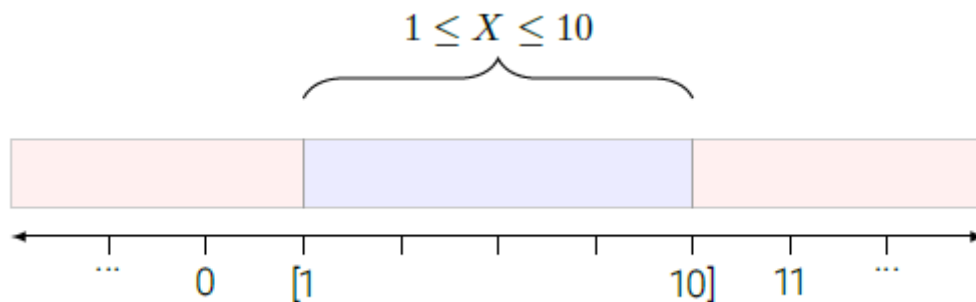
Ejercicios estructura SEGUN

1. Ingresar un número válido de día. Mostrar el nombre del día de la semana correspondiente
2. Realizar un programa que permita ingresar un número que corresponde con un número de mes. El programa deberá imprimir el nombre del mes ingresado. Por ejemplo: si se ingresa un 4, debe imprimir "Abril".
3. Realizar un programa que permita ingresar la nota (número entero) de un alumno. En función de la misma debe imprimir la calificación correspondiente según la escala que se presenta:
 - Nota menor que 5 : "Suspenso"

- Nota igual a 5: "Regular"
 - Nota igual a 6: "Bien"
 - Nota igual a 7 y 8: "Notable"
 - Nota igual a 9: "Excelente"
 - Nota igual a 10: "Sobresaliente"
4. Hacer un programa que permita ingresar dos números y el símbolo de la operación (+, -, *, /), obteniéndose el correspondiente resultado. Si el símbolo no es correcto deberá imprimir un mensaje que indique "Error en símbolo".

Anexo: Uso de operadores lógicos

Se quieren validar números entre 1 y 10 inclusive, según el intervalo de la gura



El ordenador no entiende las expresiones matemáticas tal cual las entendemos los humanos. Por ello, se debe descomponer la condición $1 \leq X \leq 10$, de una forma en que el compilador lo pueda interpretar.

Existen dos formas:

- Utilizando lógica "positiva"
- Utilizando lógica "negativa"

Utilizando lógica "positiva"

En esta forma la pregunta se escribe siguiendo la lógica tal cual como nos preguntaríamos nosotros:

- "¿El número SE ENCUENTRA dentro del rango?"
- "¿El número ES mayor a 1 Y también ES menor a 10?"

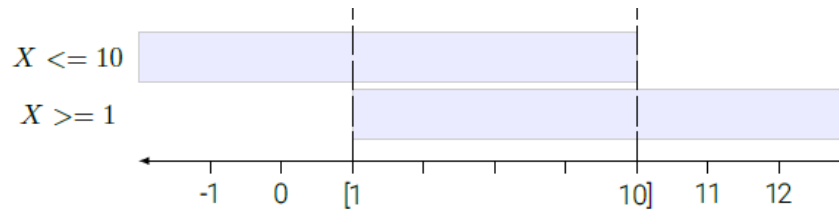
Se debe descomponer el rango $1 \leq X \leq 10$ de una forma en que la computadora lo pueda procesar.

Eso se logra desarmando la desigualdad en dos términos:

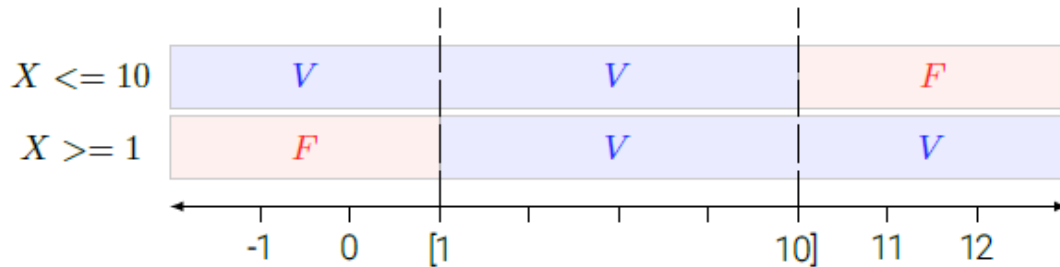
- $X \geq 1$ (1.1)
- $X \leq 10$ (1.2)

Ambas condiciones constituyen las condiciones parciales que deben ser validadas en simultáneo para determinar la pertenencia de un número al rango establecido.

En forma visual:



Analizando como se cumplen las condiciones parciales en cada segmento del rango:



¿Qué operador escojo Y (AND) o el operador O (OR)?... En este caso selecciono el operador **Y** ya que es la operación que retorna verdadero cuando todas las condiciones evaluadas son verdaderas.

$X \leq 10$	$X \geq 1$	¿en intervalo?	AND	OR
F	F	No se da	F	F
F	V	NO	F	V
V	F	NO	F	V
V	V	SI	V	V

Solución:

$$(X \geq 1) \text{ Y } (X \leq 10)$$

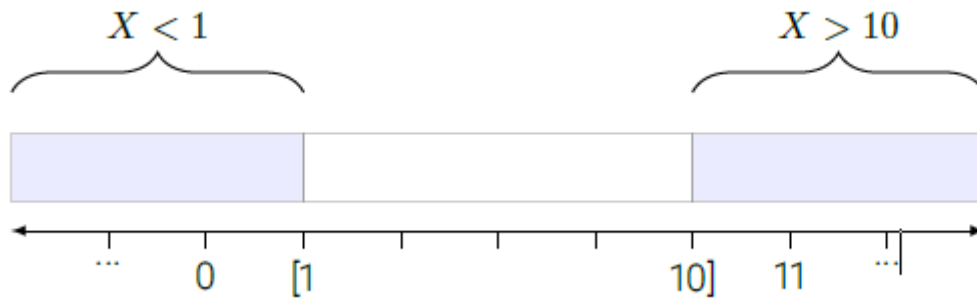
Utilizando lógica "negativa"

En esta forma la pregunta se escribe siguiendo la lógica inversa a como nos preguntaríamos normalmente:

"¿Es cierto que el número NO SE ENCUENTRA FUERA del rango?"

"¿Es cierto que el número NO ES menor a 1 y TAMPOCO ES mayor que 10?"

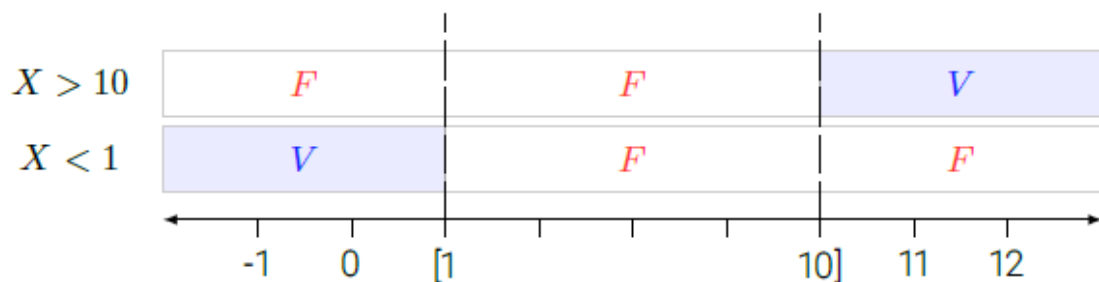
Visualmente se puede representar así:



En este caso se debe descomponer el rango $1 < X < 10$ de una forma en que la computadora lo pueda procesar. Eso se logra desarmando las desigualdades en dos términos:

- $X < 1$
- $X > 10$

Ambas condiciones constituyen las condiciones parciales que deben ser validadas para determinar la pertenencia del número al rango establecido. En forma visual:



¿Qué operador escojo Y (AND) o el operador O (OR)?... En este caso selecciono el operador **O** ya que es la operación que retorna verdadero cuando alguna de las condiciones evaluadas son verdaderas para luego negarlo y quedarme con el rango que necesito.

Solución:

$$\text{NO } ((X < 1) \text{ O } (X > 10))$$

Entonces surge la pregunta: ¿Cuál hay que usar?.

Cualquiera de ellas sirve, depende el contexto del programa el uso de una forma u otra.

Se sugiere manejar ambos tipos de razonamiento.