

Universidad Autónoma de Entre Ríos
Facultad de Ciencia y Tecnología
Sede: Oro Verde



FUNDAMENTOS DE PROGRAMACIÓN

RESUMEN DE CONTENIDOS N°3
**Estructuras de Control: Secuencia y
Selección**

Estructuras de Control: Secuencia y Selección

Introducción

Hasta aquí, se han desarrollado programas empleando tres acciones primitivas fundamentales: asignación, lectura y escritura. Las mismas han permitido resolver problemas sencillos, con la característica además, de que las acciones que lo formaban se ejecutaban secuencialmente, en el orden en que aparecían, es decir: una a continuación de otra.

Pero en el diseño de algoritmos, para la resolución de problemas más complejos, es necesario contar con herramientas que nos permitan modificar el orden lineal de ejecución de las acciones, donde el ejecutante pueda tomar decisiones y determinar qué acción realizar en el momento de ejecución del algoritmo.

Para ello, la diagramación estructurada nos brinda recursos propios. En rigor, esta metodología establece que todo algoritmo –por más complejo que sea – puede elaborarse mediante el uso de sólo tres estructuras lógicas de control:

- **Secuencia**
- **Selección**
- **Iteración**

Estas **estructuras** presentan la característica de tener un único punto de entrada y un único punto de salida. Un programa definido en base a estas estructuras, es más fácil de entender, y permite, por lo tanto, detectar los errores de lógica más rápidamente.

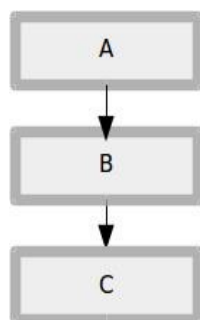
A continuación, se explicará cada una de las estructuras de control citadas, en algorítmica básica, y luego su equivalente en C++.

ESTRUCTURAS LÓGICAS DE CONTROL

SECUENCIA

Es un conjunto de acciones que se ejecutan en el mismo orden en que aparecen en el algoritmo, una a continuación de la otra.

La representación gráfica en un diagrama de flujo es:



Los bloques A, B, C representan una acción elemental o un módulo; cada uno de ellos, con un único punto de entrada y un único punto de salida.

En pseudocódigo o lenguaje de programación, la secuencia se especifica indicando las acciones en el orden en que deben ser ejecutadas, separadas entre sí por el signo de puntuación ";", (escritas en el mismo renglón o en diferente):

```
A; B;  
C;
```

SELECCIÓN

La necesidad de contar con una estructura de **selección** que nos permita tomar decisiones en un algoritmo, se planteó en los primeros problemas, en los algoritmos no computacionales, donde se usaban condiciones que permitían al ejecutante decidir qué grupo de acciones ejecutar y cuáles no.

Ejemplo:

SI se tiene soda
ENTONCES agregar en la taza media cucharadita de soda
SINO agregar en la taza media cucharadita de agua fría
FINSI

Las estructuras de **selección** otorgan al ejecutante, la posibilidad de alterar el flujo lineal de control de las acciones de un algoritmo. Permiten tomar decisiones y elegir un camino a seguir en base a ciertos valores específicos del algoritmo, ya sean datos que se ingresan o resultados de cálculos que se realizan.

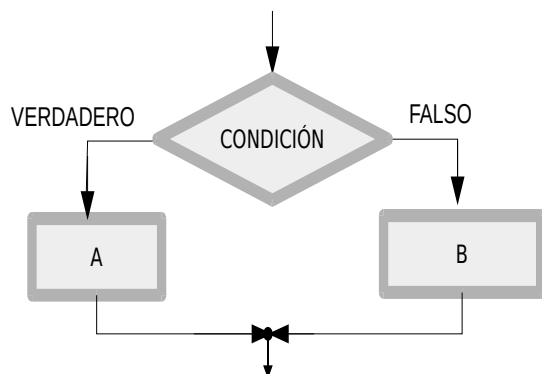
Observamos 2 tipos de estructuras de selección:

- condicional o de decisión
- selección múltiple

Estructura condicional o de decisión

Esta estructura permite elegir el camino a seguir según el valor de verdad de una condición. La estructura **condicional** permite al ejecutante o procesador, elegir las acciones a seguir entre dos alternativas específicas, de acuerdo al valor de una condición en el momento de ejecución del algoritmo o programa.

La representación de esta estructura en un diagrama es:



donde **condición**, es cualquier elemento del ambiente del algoritmo que arroje un resultado lógico (**Verdadero** o **Falso**; **True** o **False**), es decir que el elemento **condición** puede ser una **variable** o **expresión lógica**.

Tanto **A** como **B** representan una acción primitiva o un conjunto de acciones.

La presencia de esta estructura en un programa, le indica al procesador que debe:

- 1) Evaluar la condición planteada

- a) si la condición toma un valor **Verdadero (True)**, ejecutar el bloque A; y finaliza la estructura condicional indicada con el punto.
- b) si la condición toma un valor **Falso (False)**, ejecutar el bloque B; y finaliza la estructura condicional indicada con el punto.

2) En ambos casos, luego continúa la ejecución del algoritmo con la acción posterior al punto.

La sintaxis en pseudocódigo sería:

SI (condición) ENTONCES A SINO B FINSI

donde las palabras claves **SI** y **FINSI** indican, respectivamente, el comienzo y fin de la estructura de selección.

De acuerdo al valor de la condición, el procesador ejecuta A (alternativa verdadera) o B (alternativa falsa) y luego continúa con la acción que sigue al **FINSI**.

A esta estructura se la considera como una unidad, con un **único punto de entrada y un único punto de salida**. Su punto de entrada es la evaluación de la condición y su punto de salida, luego de haberse ejecutado el camino correspondiente, es la acción que sigue al punto (en el diagrama) o al **FINSI** (en pseudocódigo).

Ejemplo Práctico 1:

Una compañía de turismo ha definido una política de promoción de sus empresas para lo cual ha fijado descuentos para sus clientes en función de los viajes anteriores realizados.

El valor del descuento es del 20% para aquellos clientes que, considerando los viajes realizados en el último año y el que desean realizar, han recorrido más de 3000 Kms. y del 5% para aquellos que no cumplen dicho requisito.

Se desea realizar un algoritmo que calcule el importe a pagar por un cliente, si se ingresa su nombre y apellido, el total de kilómetros recorridos hasta el momento, y los datos del viaje a contratar: precio, destino, kilómetros a recorrer. Informar los datos ingresados, el monto del descuento y el monto a pagar por el cliente.

Ambiente:

Variable	Num.	Carac.	Lóg.	Clase	Significado
NYA		X		Simple	Nombre y Apellido del Cliente
TACUM	X			Simple	Kilómetros acumulados hasta la fecha
PRECIO	X			Simple	Precio del viaje
KMS	X			Simple	Kms. a recorrer
DEST		X		Simple	Destino del Viaje
TOTAL	X			Simple	Kilómetros totales incluidos los del viaje actual

DESC	X			Simple	Descuento a realizar
IMP	X			Simple	Importe Total

Algoritmo: En Pseudocódigo

ALGORITMO TURISMO

LEER NYA, TACUM, PRECIO, DESTINO, KMS;

TOTAL = TACUM + KMS ;

SI (TOTAL > 3000)

ENTONCES DESC = PRECIO * 0.20;

SINO DESC= PRECIO * 0.05;

FINSI

IMP = PRECIO - DESC;

ESCRIBIR ' Cliente: ', NYA, ' Acumulados: ', TACUM;

ESCRIBIR ' Datos del viaje: \$' , PRECIO, KMS, 'Kms.', DESTINO;

ESCRIBIR 'Descuento: \$', DESC, 'Importe: \$', IMP;

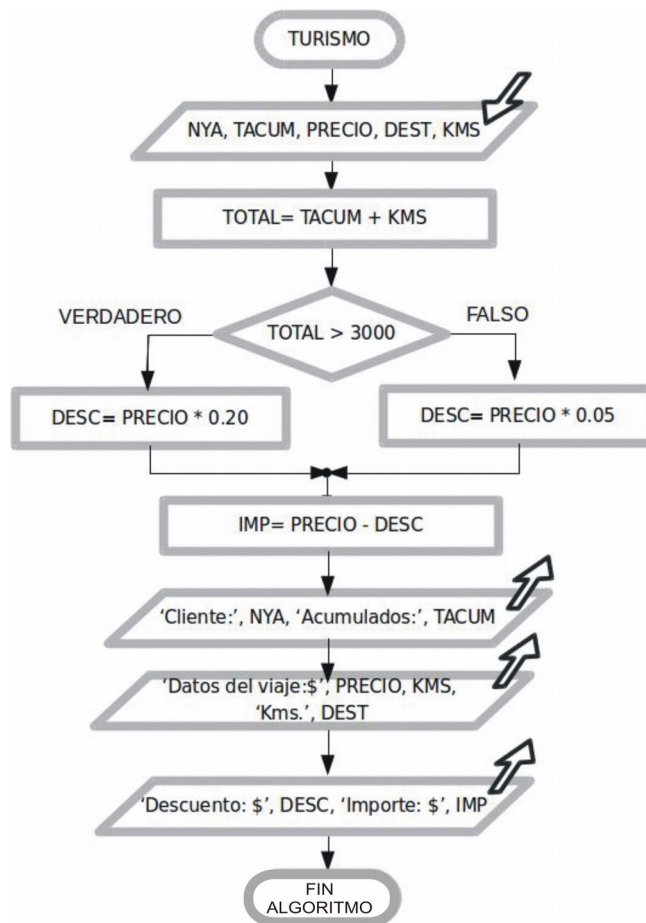
FINALGORITMO



Observación: no se coloca punto y coma (;):

- En el planteo de la condición **SI (...)**
- después de la palabra clave **FINSI**

Algoritmo: En Diagrama de Flujo



Seguimiento:

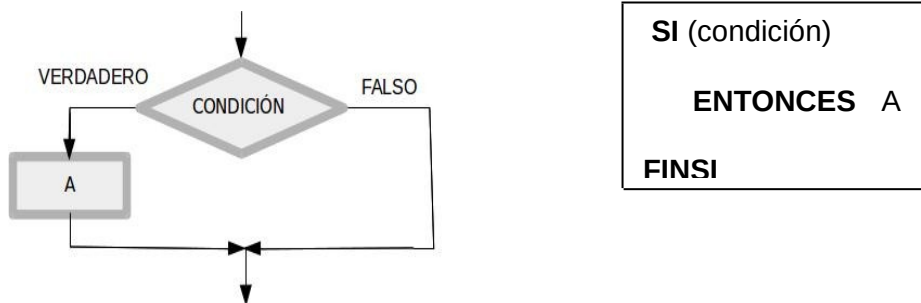
Lote de prueba: 'Juan Pérez', 3100, 4000, 'Misiones', 800

NYA	KMREC	PRECIO	DESTINO	KMS	TOTAL	DESC	IMP
'Juan Pérez'	3100	4000	'Misiones'	800	3900	800	3200

Salida por pantalla
Cliente: Juan Pérez Acumulados: 3100 Datos del Viaje: \$4000 800 Kms. Misiones Descuento: \$800 Importe: \$3200

Además, la estructura condicional brinda la posibilidad de plantear que por la alternativa falsa no se especifiquen acciones a ejecutar. De esta manera, en el condicional, la salida por **VERDADERO** deberá presentar **al menos una acción** a ejecutar, mientras que por el camino de la condición **FALSA** no se plantea ninguna acción. Esta variante de la estructura condicional se representa en diagrama de la siguiente manera:

En pseudocódigo, la sintaxis equivalente a este caso es:



A continuación, mostramos el enunciado del **Ejemplo Práctico 1**, con modificaciones, para observar el cambio en la estructura condicional:

Una compañía de turismo ha definido una política de promoción de sus empresas para la cual ha fijado un descuento del 20% para aquellos clientes que, considerando los viajes realizados en el último año y el que desean realizar, superen los 3000 Kms. de recorrido.

Se desea realizar un algoritmo que calcule el importe a pagar por un cliente, si se ingresa su nombre y apellido, el total de Kms. recorridos hasta el momento, y los datos del viaje a contratar: precio, destino, Kms. a recorrer. Informar los datos ingresados, el monto del descuento y el monto a pagar por el cliente.

Ejemplo Práctico 2:

Realizar un algoritmo que ingrese como datos, el precio unitario y la cantidad de unidades vendidas de un artículo e informe el monto neto de venta, sabiendo que para cantidades superiores a 500, se efectúa un descuento del 20 %.

Ambiente:

Variable	Num.	Carac.	Lóg.	Clase	Significado
PRECIO	X			Simple	Precio de venta del artículo
CANT	X			Simple	Cantidad de artículos vendidos
MONTO	X			Simple	Monto de la venta
CUESTION			X	Simple	Valor de la condición $Cant > 500$

Algoritmo: En Pseudocódigo

ALGORITMO VENTA

LEER PRECIO, CANT;

MONTO= PRECIO * CANT;

CUESTION= CANT > 500;

SI (CUESTION)

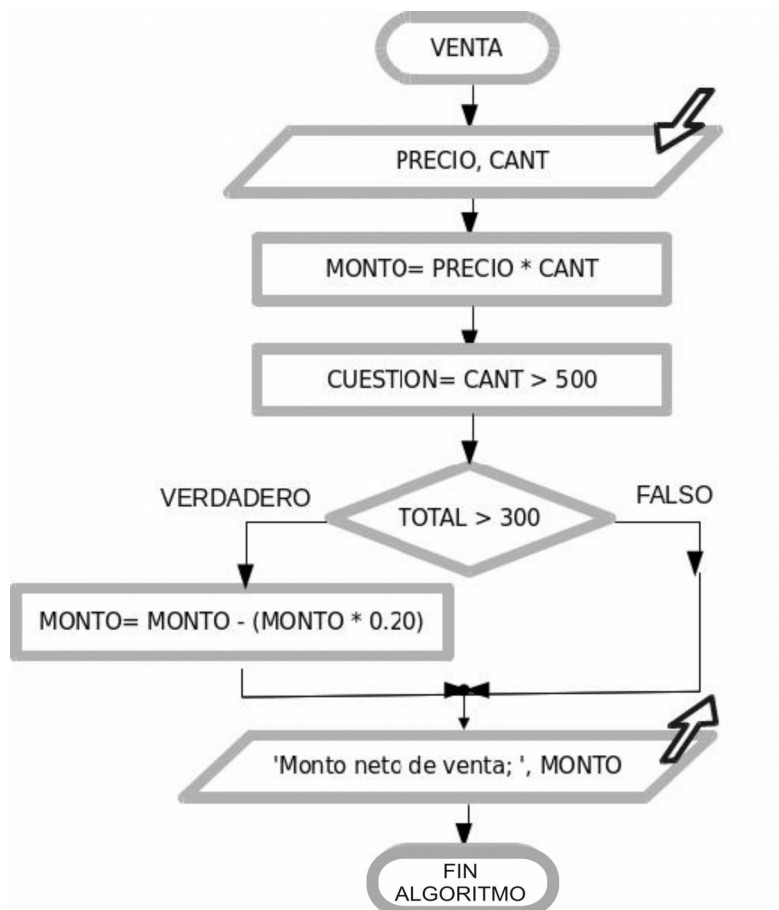
 ENTONCES MONTO= MONTO – (0.20 * MONTO);

FINSI

ESCRIBIR 'Monto neto venta:', MONTO;

FINALGORITMO

Algoritmo: En Diagrama de Flujo



Estructuras condicionales anidadas

En una estructura condicional, tanto la alternativa verdadera como la falsa pueden presentar a su vez, otras estructuras condicionales, dando lugar a esquemas como el siguiente:

En pseudocódigo:

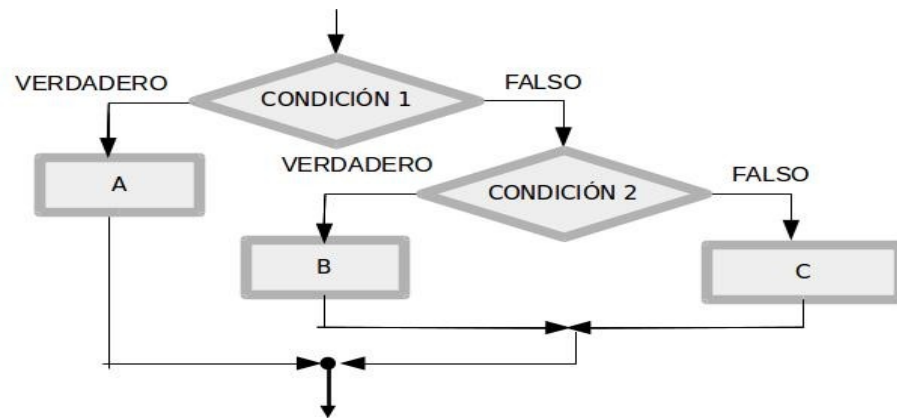
```

SI (condición1)
  ENTONCES A
  SINO SI (condición2)
    ENTONCES B
    SINO C
  FINSI
FINSI
  
```

En este esquema hablamos de condicionales anidados, con un condicional externo - el primero que se plantea (evalúa la condición1) - y condicionales internos (en este caso, aparece uno solo que evalúa la condición2).

Este esquema puede complicarse, si en A, B, y/o C planteamos otros condicionales internos, pero debe tenerse en cuenta que cada condicional debe corresponder al esquema **SI-FINSI**.

En diagrama de flujo:



Ejemplo Práctico 3:

Una compañía de turismo ha definido una política de promoción de sus empresas para lo cual ha fijado descuentos para sus clientes en función de los viajes anteriores realizados. El valor del descuento es del 20% para aquellos clientes que, considerando los viajes realizados en el último año y el que desean realizar, han recorrido más de 3500 Kms., del 15 % si han sumado más de 2000 Kms., del 5% para aquellos que han superado los 500 kms. Se desea realizar un algoritmo que calcule el importe a pagar por un cliente, si se ingresa su nombre y apellido, el total de Kms. recorridos hasta el momento, y los datos del viaje a contratar: precio, destino, Kms. a recorrer. Informar los datos ingresados, el monto del descuento y el monto a pagar por el cliente.

Ambiente:

En este caso el ambiente es el mismo que el realizado para el Ejemplo 1.

Algoritmo: En Pseudocódigo

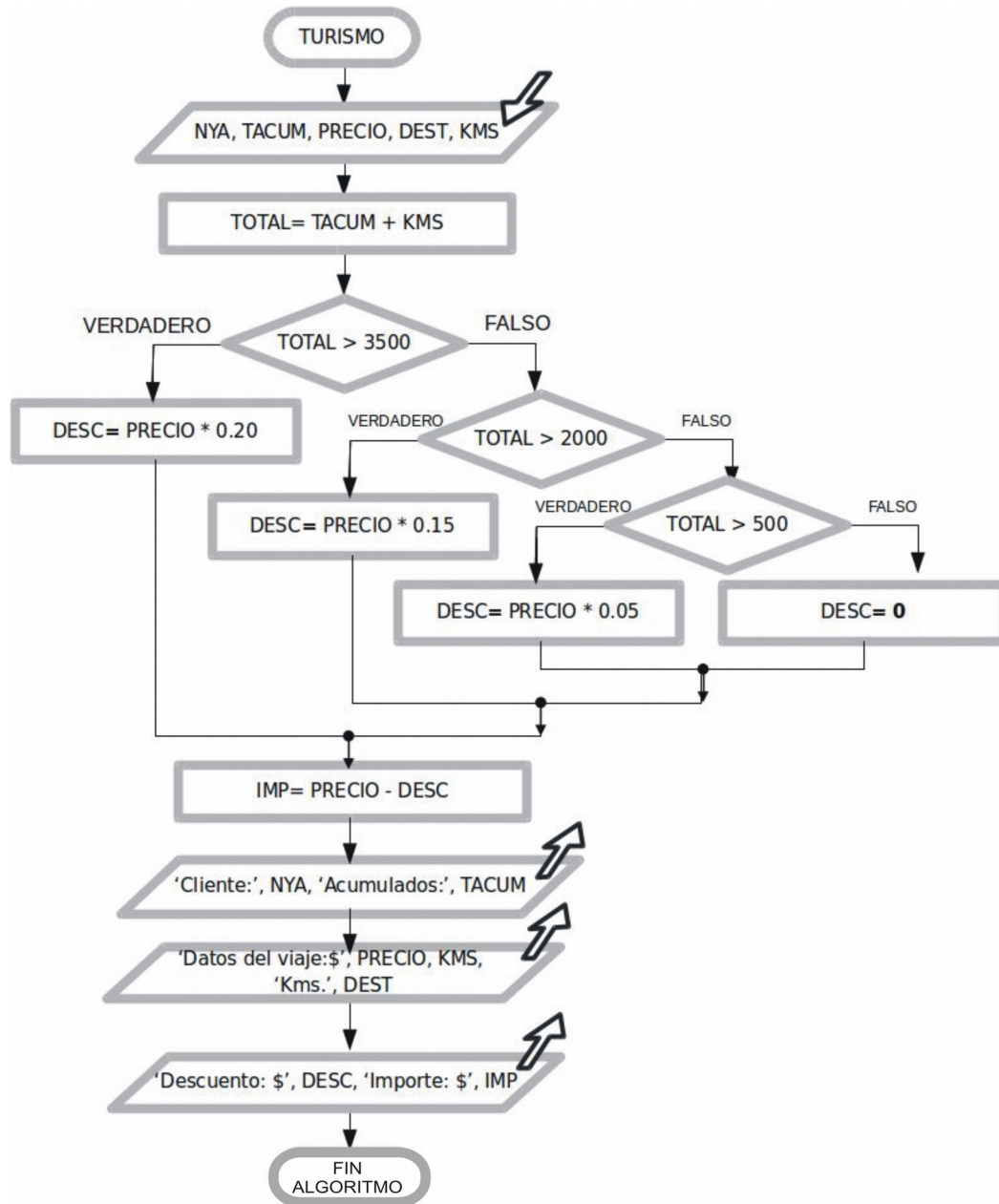
ALGORITMO TURISMO

```

LEER NYA, TACUM, PRECIO, DEST, KMS;
TOTAL= TACUM + KMS ;
SI (TOTAL > 3500 )
    ENTONCES DESC= PRECIO * 0.2;
    SINO SI (TOTAL > 2000)
        ENTONCES DESC= PRECIO * 0.15 ;
        SINO SI (TOTAL > 500)
            ENTONCES DESC= PRECIO * 0.05;
            SINO      DESC= 0;
        FINSI
    FINSI
FINSI

IMP= PRECIO - DESC;
ESCRIBIR ' Cliente: ', NYA, ' Acumulados: ', TACUM;
ESCRIBIR ' Datos del viaje: $' , PRECIO, KMS, 'Kms.', DEST;
ESCRIBIR 'Descuento: $ ', DESC, 'Importe: $', IMP;
  
```

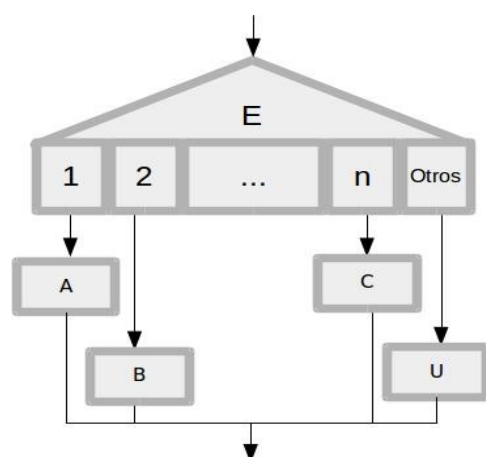
Algoritmo: En Diagrama de Flujo



Estructura de selección múltiple

Esta estructura es una generalización de la estructura condicional y brinda la posibilidad de elegir entre más de dos alternativas de acuerdo al valor de una variable de control. En pseudocódigo y en algunos lenguajes de programación, esta variable de control debe ser siempre del tipo numérica.

En diagrama de flujo se representa de la siguiente manera:



donde **E** es la variable de control, **numérica**, que toma todos los valores enteros positivos, acotados dentro de un rango preestablecido, generalmente de 1 a N.

A, B, C, U representan una acción elemental.

En presencia de esta estructura el procesador deberá:

- observar el valor de la variable E
- decidir el camino a ejecutar de acuerdo al valor de E:
- Si E toma el valor 1, realizará la acción A
- Si E toma el valor 2, ejecutará la acción B y así sucesivamente para cada uno de los valores posibles de E que se hayan indicado.
- Si E toma valores fuera del rango 1 a N, deberá ejecutar la acción U, indicada en "otros".
- en todos los casos, ejecutada la acción asociada, deberá salir de la estructura y continuar con la acción siguiente especificada después del punto.

La estructura de decisión múltiple, mantiene las características vistas en las anteriores, de tener **un único punto de entrada y un único punto de salida**.

La sintaxis equivalente en pseudocódigo es la siguiente:

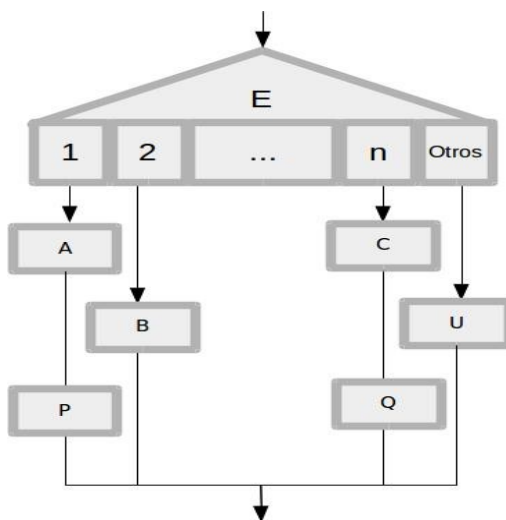
```

SEGUN E HACER
    1: A
    2: B
    . .
    . .
    n: R
    DEOTROMODO: U
FINSEGUN
  
```

El comienzo y el fin de la estructura están identificados por las palabras claves **SEGÚN** y **FINSEGÚN**, respectivamente. Ejecutada la acción asociada al valor de la variable **E**, el ejecutante finaliza la estructura y continúa con la acción siguiente a la palabra clave **FINSEGÚN**.

Puede suceder que para alguno de los caminos indicados se deba ejecutar más de una acción.

En este caso en el diagrama de flujo las acciones a realizar se graficarán de esta forma:



En pseudocódigo, la sintaxis correspondiente es:

```

SEGUN E HACER
  1: COMENZAR
      A;
      P;
      FINCOM
  2: B;
  . .
  . .
  n: COMENZAR
      C;
      Q;
      FINCOM
  DEOTROMODO: U;
FINSEGUN
  
```



Observación: no se coloca punto y coma (;):

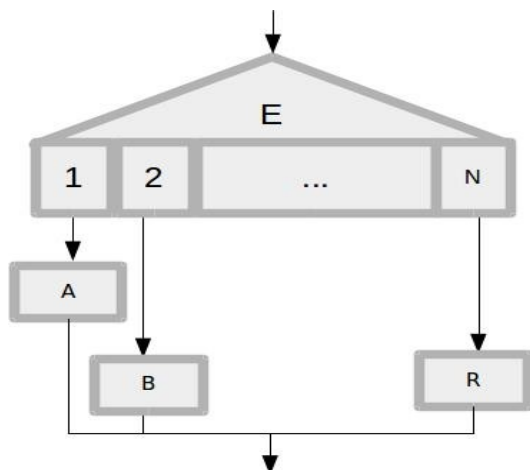
- en la acción **SEGUN HACER**
- después de la palabra clave **FINSEGUN**

Otros formatos de la estructura de selección múltiple:

- a) Si cuando la variable de control toma valores distintos a los especificados no existe acción a realizar, esta estructura permite omitir la rama correspondiente a "otros" (o **DEOTROMODO** en el pseudocódigo).

Esta alternativa nos queda:

En diagrama:



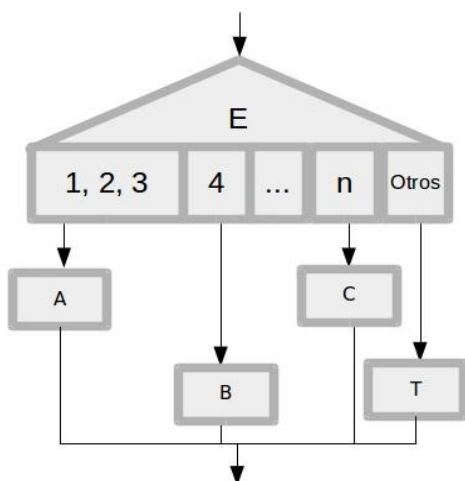
En pseudocódigo:

```

SEGUN E HACER
    1: A
    2: B
    . .
    . .
    n: R
FINSEGUN
  
```

- b) Cuando para distintos valores de la variable de control se deban ejecutar la/s misma/s acción/es, la estructura permite agrupar dichos valores indicando el camino de acciones a realizar, por única vez.

En diagrama:



En pseudocódigo:

```

SEGUN E HACER
    1, 2, 3: A
    4: B
    ...
    ...
    n: R
    DEOTROMODO: T
FINSEGUN
  
```

En este caso se consideró que para los valores de **E** igual a 1, 2 ó 3 se debe ejecutar la acción A.

Ejemplo Práctico 4:

Se desea calcular el sueldo de un empleado, conociendo como datos su legajo, nombre y apellido, categoría (1, 2, 3, ó 4) y sueldo básico. Se le paga además, una bonificación que depende de la categoría. Para la categoría 1 y 3, la bonificación es del 30 % del sueldo básico, para la categoría 2, del 20 % y para la categoría 4, del 15 %. Informar legajo, sueldo básico, bonificación y sueldo a cobrar. (En el dato categoría, sólo pueden venir los valores indicados).

Ambiente:

Variable	Num.	Carac.	Lóg.	Clase	Significado
LEG	X			Simple	Legajo del empleado
NYA		X		Simple	Nombre y apellido del empleado
CAT	X			Simple	Categoría del empleado
SB	X			Simple	Sueldo básico del empleado
BON	X			Simple	Bonificación que cobra
TOT	X			Simple	Sueldo neto a cobrar

Algoritmo: En Pseudocódigo

ALGORITMO SUELDO

LEER LEG, NYA, CAT, SB;

SEGUN CAT HACER

1, 3: BON= 0.30 * SB;

2: BON= 0.20 * SB;

4: BON= 0.15 * SB;

FINSEGUN

TOT= SB + BON;

ESCRIBIR NYA, ' ', LEG, ' ', SB, ' ', BON;

ESCRIBIR 'Sueldo a cobrar:', TOT;

FINALGORITMO



NOTA: En el ejemplo las acciones a realizar para 2 valores de la variable CAT, son las mismas. En estos casos, es posible agrupar dichos valores e indicar una única vez las acciones a realizar.

Algoritmo: En Diagrama de Flujo

