



aprenderaprogramar.com

Transformación de tablas de decisión en pseudocódigo (if ... then) (1). Diagramas de flujo. (CU00148A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel I

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 47 del Curso Bases de la programación Nivel I

24

TRANSFORMACIÓN DE TABLAS DE DECISIÓN EN PSEUDOCÓDIGO Y DIAGRAMAS DE FLUJO.

Habíamos definido a las tablas de decisión como tablas donde se organizan los condicionantes y las acciones a realizar con el fin de lograr más fáciles estructuraciones de los problemas. Con nuestros conocimientos actuales, podríamos decir que las tablas de decisión nos sirven para plantear problemas con múltiples alternativas de actuación y un alto grado de anidamiento de estructuras *Si ... Entonces – Si/No*. El cerebro humano trabaja bien con anidamientos simples pero los anidamientos múltiples resultan difíciles de seguir si no se utiliza un método que nos evite perdernos.

¿Cómo enfocar un problema de anidamientos múltiples?

El primer paso es identificar al problema como adecuado para tratarlo con una tabla de decisión y construir la misma. El siguiente paso sería transformar esa tabla en pseudocódigo o diagrama de flujo. Vamos a ver cómo hacerlo. Citaremos tres opciones:

- a) A ojo de buen cubero.
- b) Métodos intuitivos.
- c) Métodos matemáticos.

Actuar a ojo de buen cubero sería realizar una transformación directa de la tabla en pseudocódigo. Sólo es viable con tablas muy sencillas. Con “Métodos intuitivos” nos referimos a técnicas útiles sin demostración teórica profunda y que aceptamos “porque funcionan”. Por último tendríamos un análisis matemático de la tabla para su transformación en pseudocódigo. El resultado de un método “intuitivo” debe verificarse con cuidado, pues no tenemos certeza de un correcto funcionamiento para el 100% de los casos. Vamos a explicar un método aproximado o intuitivo.

TRANSFORMACIÓN DE UNA TABLA DE DECISIÓN EN PSEUDOCÓDIGO (SI ... ENTONCES)

La forma de transformar una tabla de decisión en pseudocódigo que vamos a proponer no tiene otro fundamento que el conocido y recurrente “divide y vencerás”. Los pasos a dar son:

- Dividir progresivamente la tabla.
- Dibujar el diagrama de flujo asociado.
- Escribir el pseudocódigo asociado.

Comenzamos eliminando los enunciados de las condiciones así como la parte de acciones (lista de acciones y acciones a realizar). Nos queda algo así:

C1	
C2	
C3	
C4	

Supongamos que partimos de la siguiente tabla:

TABLA ORIGEN						
	1-2-3-8	4-6-7-12	5-11	9-13	10-14	15-16
C1	Sí	—	No	Sí	No	No
C2	Sí	No	Sí	No	—	—
C3	—	—	—	—	Sí	No
C4	—	Sí	Sí	No	No	No

Conviene verificar que la tabla está bien construida y libre de errores, que pueden existir si las simplificaciones no están bien hechas. El objeto a dividir es la “tabla”. El criterio de división es el siguiente:

- a) Se divide por la fila con menor número de indiferencias.
 - a.1) En caso de empate, se divide por la fila que contiene menor número de indiferencias en la vertical de sus indiferencias.
 - a.1.1) En caso de empate, se divide por la fila que el programador estime oportuna, atendiendo a su conocimiento del problema, y si no hay criterio a seguir siempre la opción “menor” o “sí”.

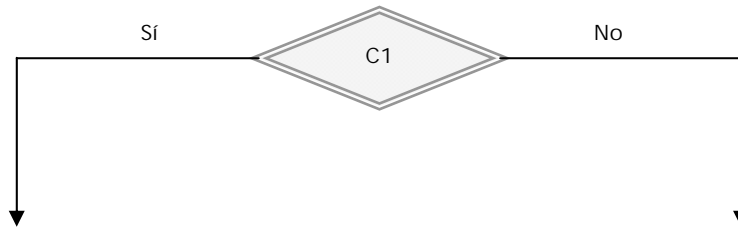
En nuestro ejemplo tenemos:

C1: 1 indiferencia.
 C2: 2 indiferencias.
 C3: 4 indiferencias.
 C4: 1 indiferencia.

Elegiremos entre C1 y C4. El siguiente criterio nos lleva a:

C1: 2 indiferencias en la vertical 4-6-7-12.
 C4: 2 indiferencias en la vertical 1-2-3-8.

Persiste el empate. Elegiremos la opción menor: C1.
C1 divide a la tabla. Gráficamente:



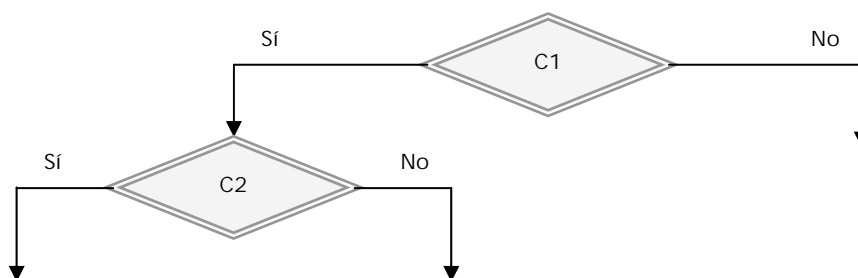
Comenzaremos a construir una rama. Elegimos la opción "Sí". Construimos una tabla C1 – Sí:

TABLA C1 - Sí			
	1-2-3-8	4-6-7- 12	9-13
C1	Sí	Sí	Sí
C2	Sí	No	No
C3	—	—	—
C4	—	Sí	No

Una indiferencia equivale a dos columnas. Por eso la columna 4-6-7-12 aparecerá en esta tabla desdoblada con valor Sí en C1. La fila que ha servido para dividir la marcaremos como sombreada. Dejaremos que "figure" sin intervenir en procesos posteriores. Volvemos a aplicar los criterios para una nueva división, ya sin intervención de C1.

C2: 0 indiferencias.
C3: 3 indiferencias.
C4: 1 indiferencia.

C2 divide a C1 – Sí. Gráficamente:



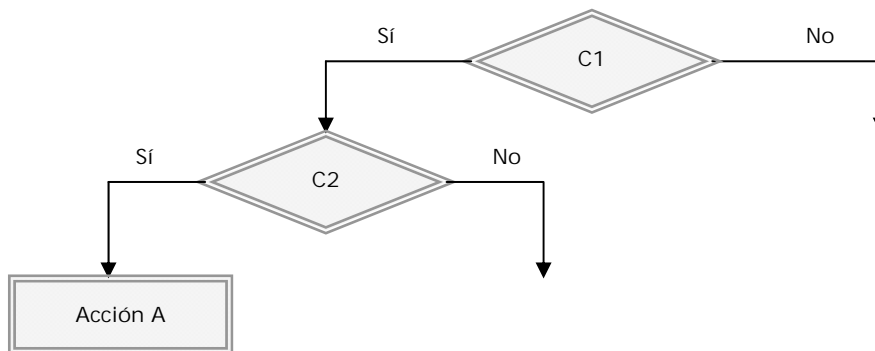
Construimos la tabla C1 – Sí – C2 – Sí.

TABLA C1 – Sí – C2 – Sí	
	1-2-3-8
C1	Sí
C2	Sí
C3	—
C4	—

Al llegar a una sola columna estamos en el nivel de acción. Podemos escribir:

C1 – Sí – C2 – Sí = Acción A.

Gráficamente:



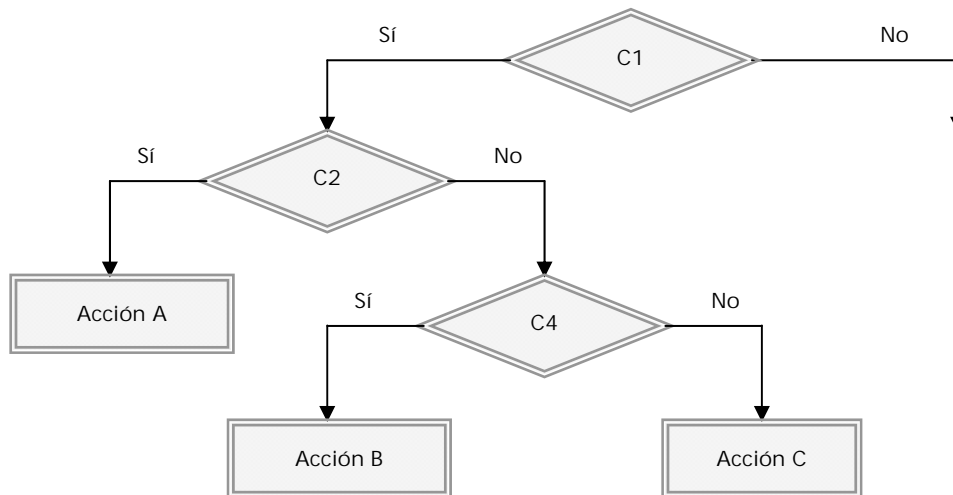
Hemos llegado a un punto terminal. Ahora regresamos al punto abierto más próximo. En este caso C1 – Sí – C2 – No. Construimos la tabla:

TABLA C1 – Sí – C2 – No		
	4-6-7-12	9-13
C1	Sí	Sí
C2	No	No
C3	—	—
C4	Sí	No

La tabla presenta una única fila variante que es C4. Estamos en un punto casi terminal. C4 divide a C1 – Sí – C2 – No y da lugar a dos acciones.

C1 – Sí – C2 – No – C4 – Sí = Acción B.

C1 – Sí – C2 – No – C4 – No = Acción C.



Hemos completado una rama del árbol. Pasamos a atacar la otra.

Próxima entrega: CU00149A

Acceso al curso completo en [aprenderaprogramar.com](http://www.aprenderaprogramar.com) --> Cursos, o en la dirección siguiente:

http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=28&Itemid=59