



aprenderaprogramar.com

# Ejercicio resuelto programa de librería II. Diagramas de flujo. (CU00223A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel II

Fecha revisión: 2024

Autor: Mario R. Rancel

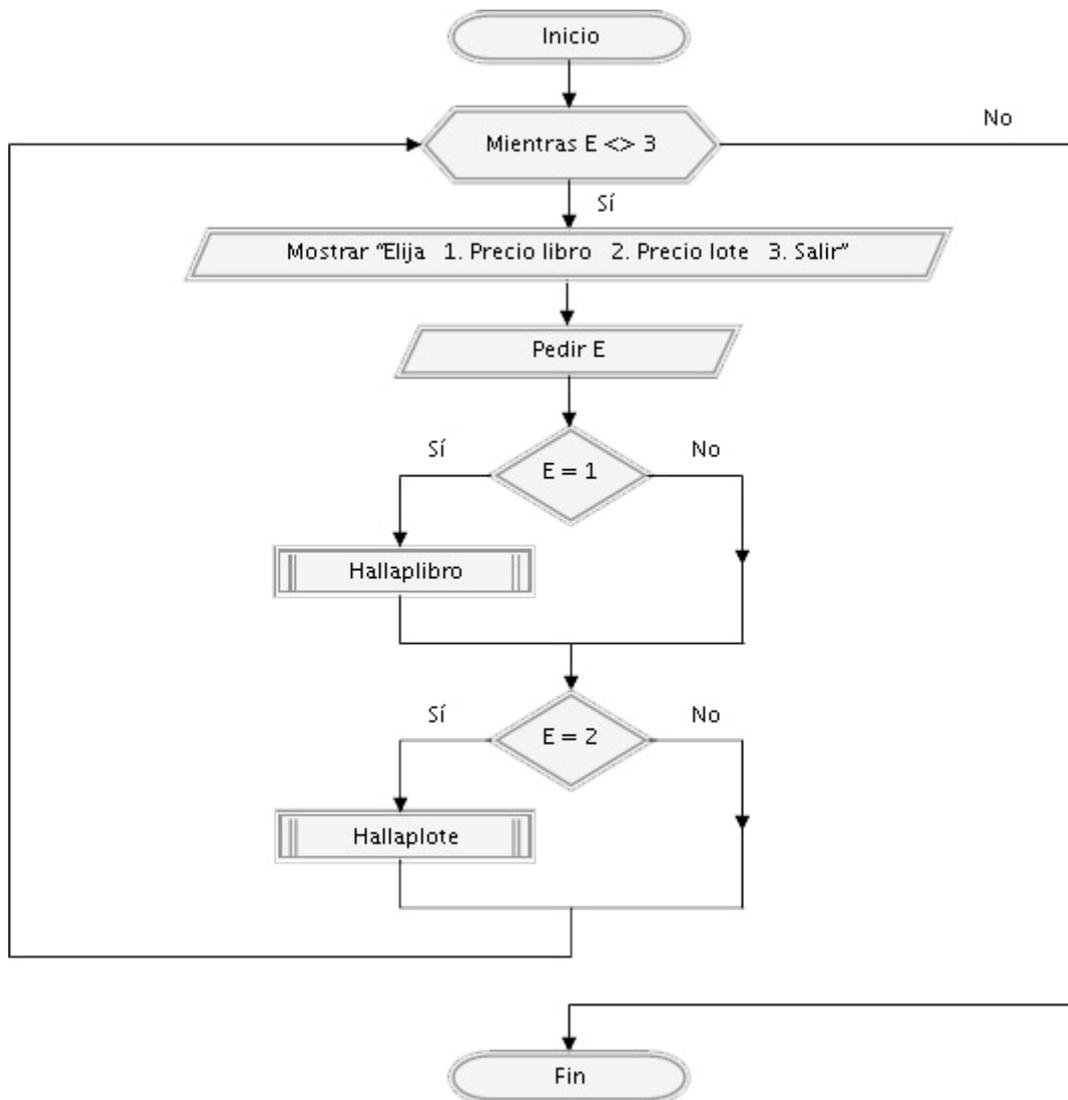
Resumen: Entrega nº 22 del Curso Bases de la programación Nivel II

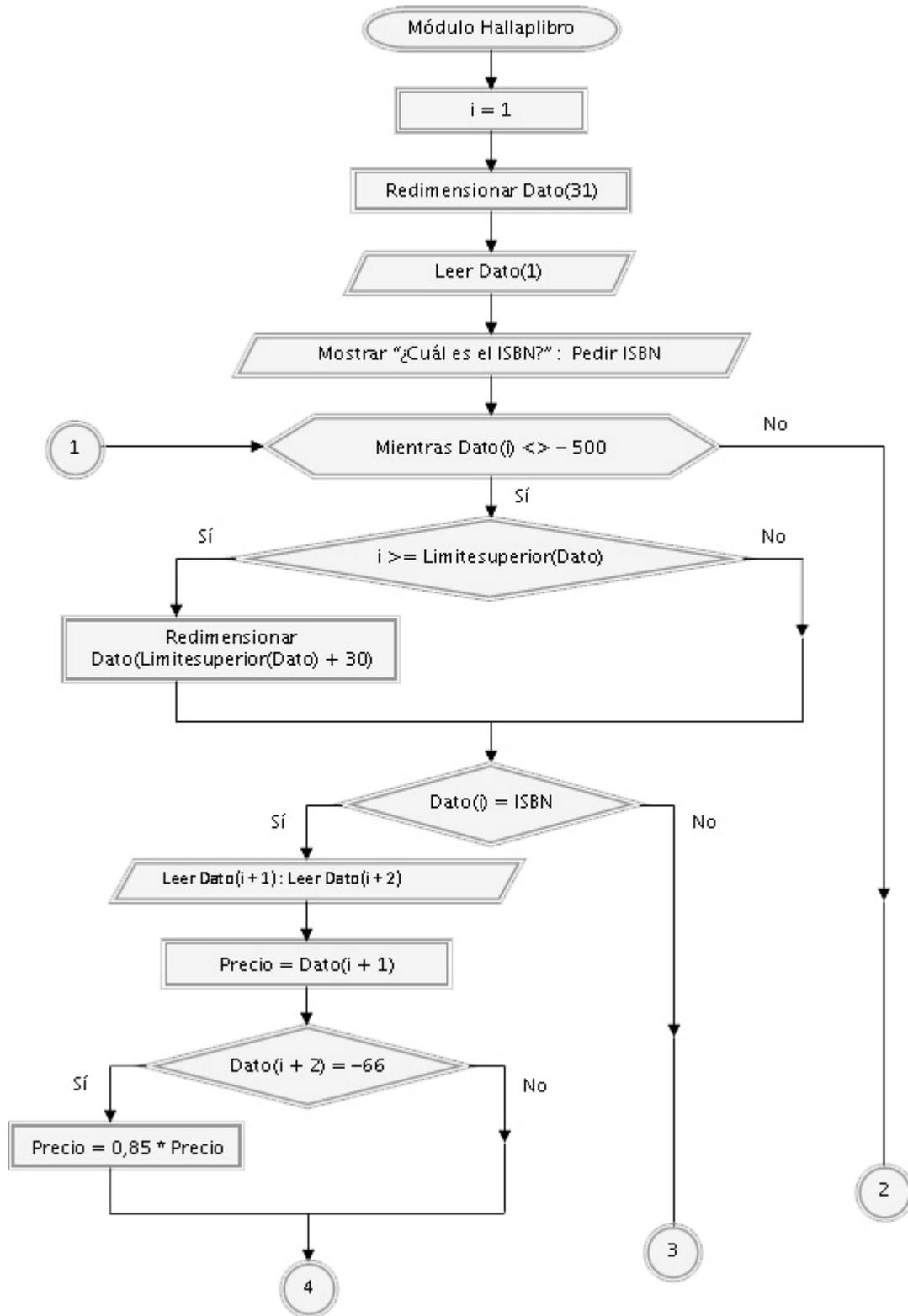
24

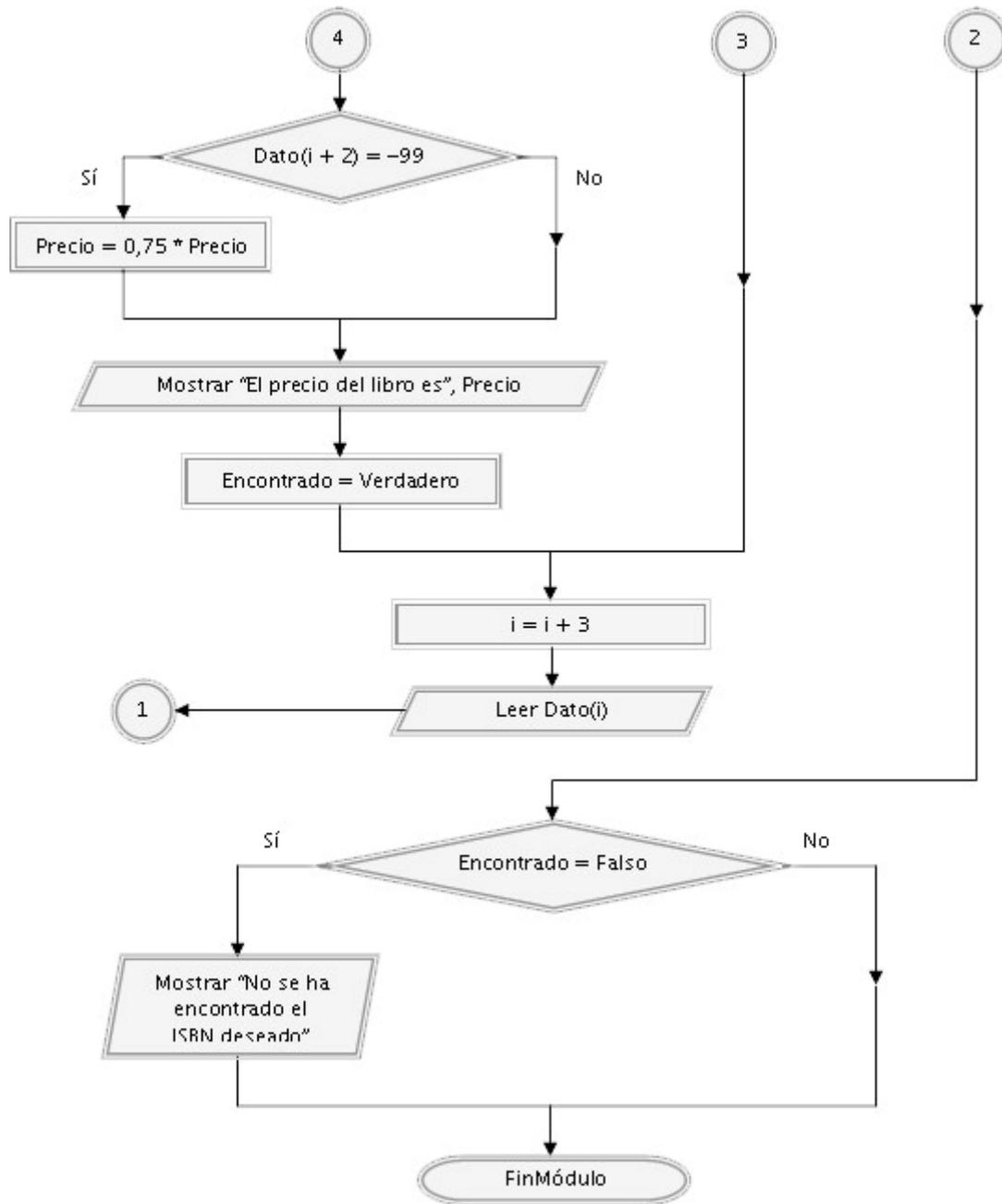
### EJERCICIO PROGRAMA GESTIÓN DE LIBRERÍA EN PSEUDOCÓDIGO. PARTE 2.

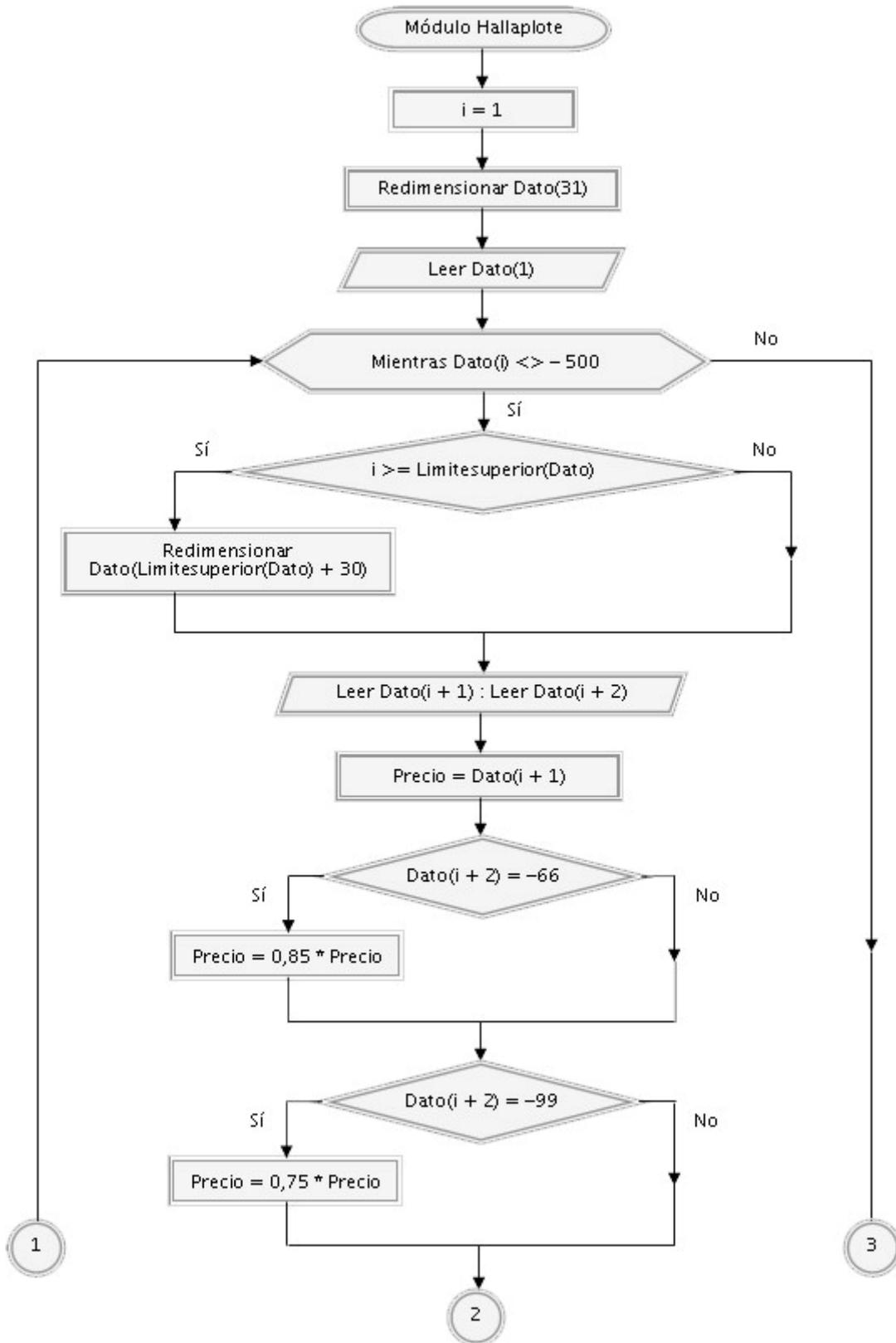
Segunda parte del algoritmo que resuelve el programa del ejercicio anterior. Recordar que nos estamos basando en conceptos como:

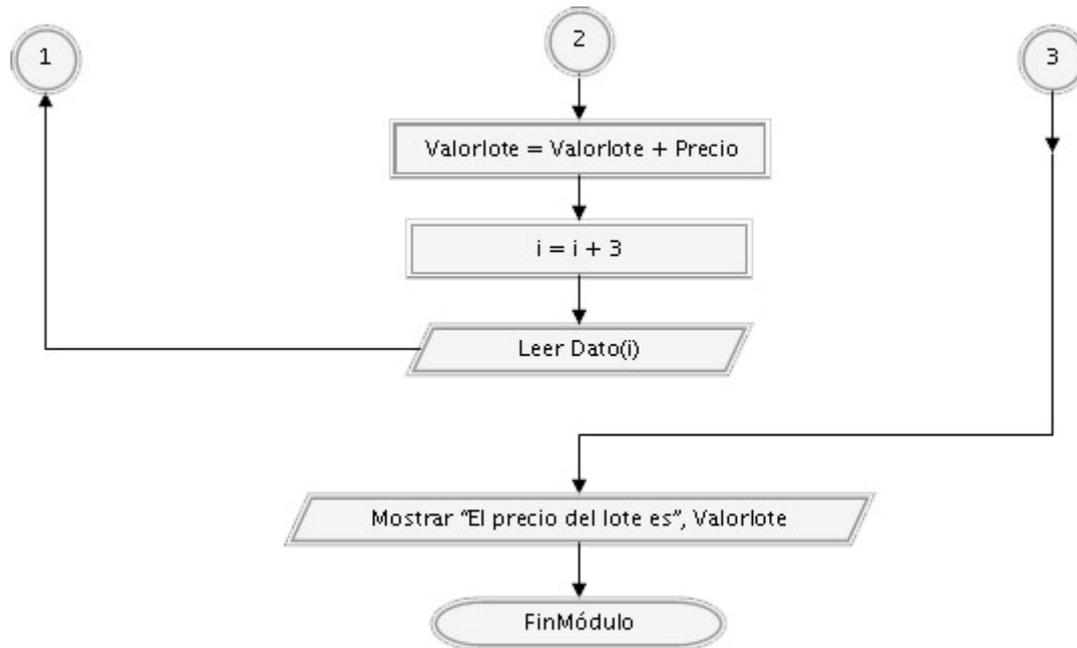
- Programación por módulos.
- Declaración de variables, utilizando la asignación por paquetes o grupos para los arrays indefinidos.
- Salida del programa controlada por el usuario.











**Comentarios:** El programa consta de un algoritmo principal y dos módulos. Se ha conservado la estructura del programa originario (ejercicio número 30) introduciendo únicamente lo propuesto, a saber: módulos, declaración de variables con arrays indefinidos tratados por paquetes y salida del programa controlada por el usuario. El programa tiene distintos aspectos mejorables en los que se ha preferido no entrar.

Veamos algunas cuestiones relevantes. Observamos que todas las variables excepto la de control del módulo principal están declaradas como variables locales. Esto supone:

- Independencia de los módulos.
- Producto cerrado de los módulos.
- Todas las variables se inicializan cuando se accede al módulo, adquiriendo valor cero o vacío: tenemos garantizado, por un lado, que lo que ocurre en otro módulo o en el algoritmo principal no nos va a afectar en ese módulo. Y por otro, nos olvidamos de posibles valores previos de variables que pudieran trastocar, por ejemplo, el funcionamiento de acumuladores.

En cuanto a la asignación de contenido al array Dato tenemos:

- Se hace una definición inicial de tamaño de 31 datos. La razón para ello es que el señalero de finalización de datos se encuentra en una posición  $1 + 3 * n$ , con  $n$  número de registros compuestos de ISBN / Precio / Descuento. Hemos escogido  $1 + 3 * 10$  pero podríamos haber elegido  $1 + 3 * 5 = 16$  ó  $1 + 3 * 15 = 46$ . Posteriormente se amplía el array cada vez que pasados 30 elementos se comprueba que siguen existiendo datos.
- No existe un ajuste final entre número de datos válidos y número de elementos del array. ¿Por qué? Habíamos dicho que normalmente es de interés redimensionar el array para que coincidan el número de datos válidos con el número de elementos del array. En este caso no se ha hecho simplemente porque no hace falta. No buscamos ser puristas sino efectivos. Entonces, ¿Para qué

molestarnos en hacer algo que no hace falta? En el módulo Hallaplibro vamos extrayendo datos y redimensionando el array en bloques de 30 unidades. Una vez extraídos todos los datos, nos da igual que en el array existan una serie de datos nulos pues no nos afectan para nada. Por ejemplo, si el señalero de cierre está en el dato 7, tendremos que Dato(8), Dato(9), ..., Dato(31) valen cero pero esto no nos afecta negativamente. Incluso, por tratarse de una variable local, no vamos a seguir arrastrando esos datos nulos todo el programa, sino que nos hemos deshecho de ellos nada más salir del módulo y volver al algoritmo principal.

En el módulo Hallaprote la situación es análoga: una vez se llega al final de datos válidos (reconocido a través del señalero – 500) se sale del proceso. Por tanto la existencia de una cola de datos nulos no nos afecta y por ser variable local no supone un lastre para el programa. En definitiva, se ha actuado así en función de las circunstancias específicas. No se trata de estar continuamente cumpliendo unas normas o criterios, sino de tener un exhaustivo conocimiento y control del programa y actuar según las circunstancias.

Una mejora interesante para este algoritmo, dado que ya conocemos medios para hacerlo, consistiría en detener la extracción de datos (proceso innecesario) después de que se haya encontrado el ISBN buscado y mostrado el precio del libro asociado.

**Próxima entrega: CU00224A**

**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=36&Itemid=60](http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=36&Itemid=60)