



aprenderaprogramar.com

**Ejercicio final de señaleros  
resuelto con pseudocódigo.**

**Organización previa de variables.**

**Diagramas de flujo. (CU00171A)**

**Sección: Cursos**

**Categoría: Curso Bases de la programación Nivel I**

**Fecha revisión: 2024**

**Autor: Mario R. Rancel**

**Resumen: Entrega nº 70 del Curso Bases de la programación Nivel I**

24

## EJERCICIO RESUELTO

Una librería dispone de un archivo donde se guarda información relativa a un lote de libros estructurada de la siguiente manera:

Dato(1) = [Número de ISBN, 10 dígitos]

Dato(2) = [Precio de venta al público, en euros]

Dato(3) = [Señalero, – 33 Libro de menos de un año de antigüedad  
– 66 Libro de uno a tres años de antigüedad  
– 99 Libro de más de tres años de antigüedad]

Dato(4) = [Número de ISBN]

·  
·  
·

Dato(n – 1) = [Señalero – 33, – 66, – 99]

Dato(n) = [Señalero – 500, Fin del lote]

Se desea desarrollar un programa que permita:

- Consultar el precio de un libro dado su ISBN.
- Conocer el precio del lote de libros en conjunto.

Teniendo en cuenta lo siguiente: si el libro tiene menos de un año de antigüedad su precio coincide con el registrado. Si tiene entre uno y tres años deberá aplicarse un descuento del 15%. Si tiene más de tres años deberá aplicarse una rebaja del 25%.

## SOLUCIÓN

### Organización previa de variables:

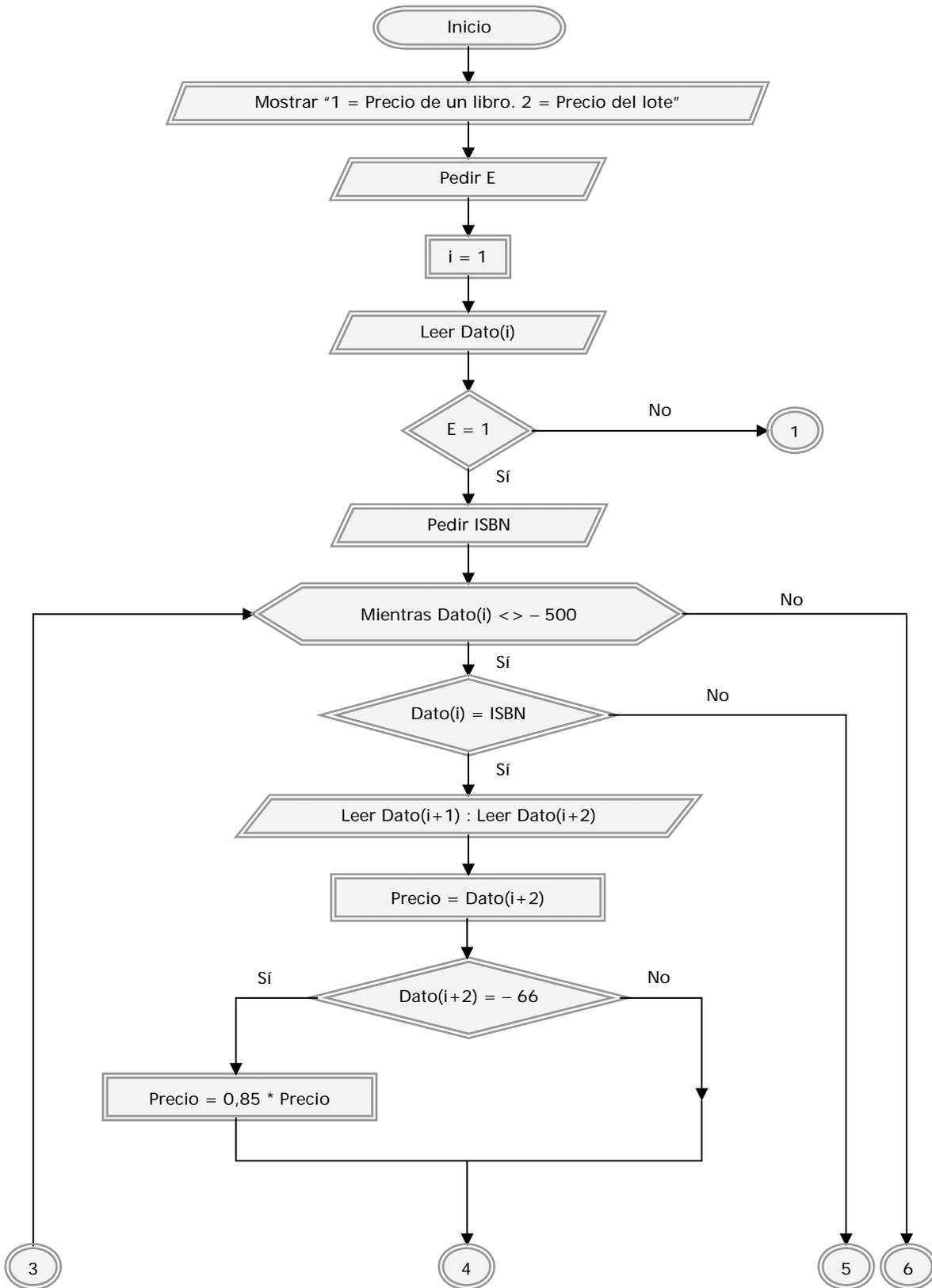
Variable	Descripción - Uso
E	Señalero para la toma de decisiones según el usuario elija conocer precio de un libro o del lote
i	Contador para extracción de datos
Dato(i)	Array de variables que contiene los datos
ISBN	Almacena el valor de ISBN actual
Precio	Almacena el valor del precio actual
Encontrado	Interruptor que informa si el ISBN pedido por el usuario fue encontrado o no (booleana)
Valorlote	Acumulador donde se realiza el sumatorio de precios de libros del lote

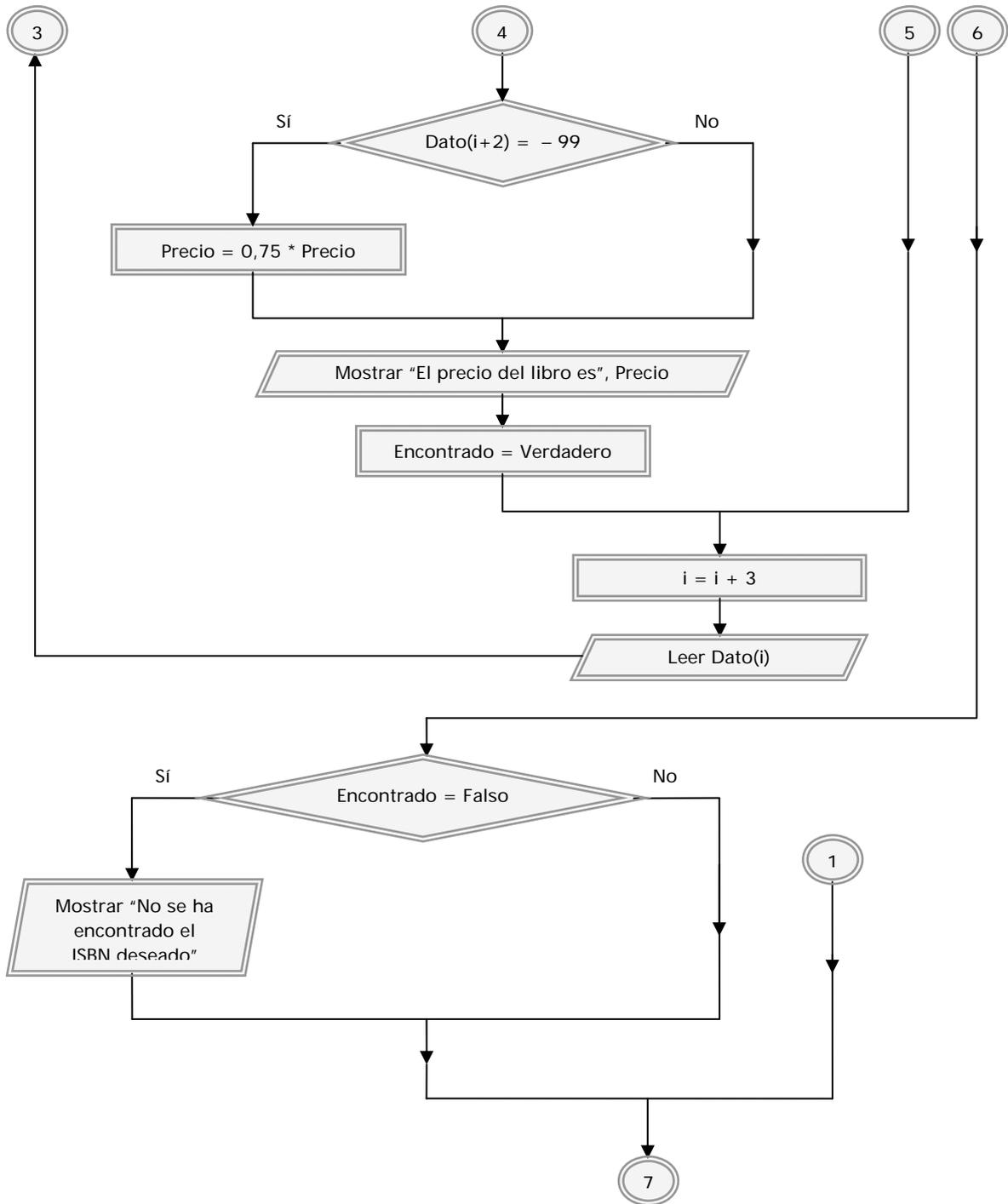
**Pseudocódigo:**

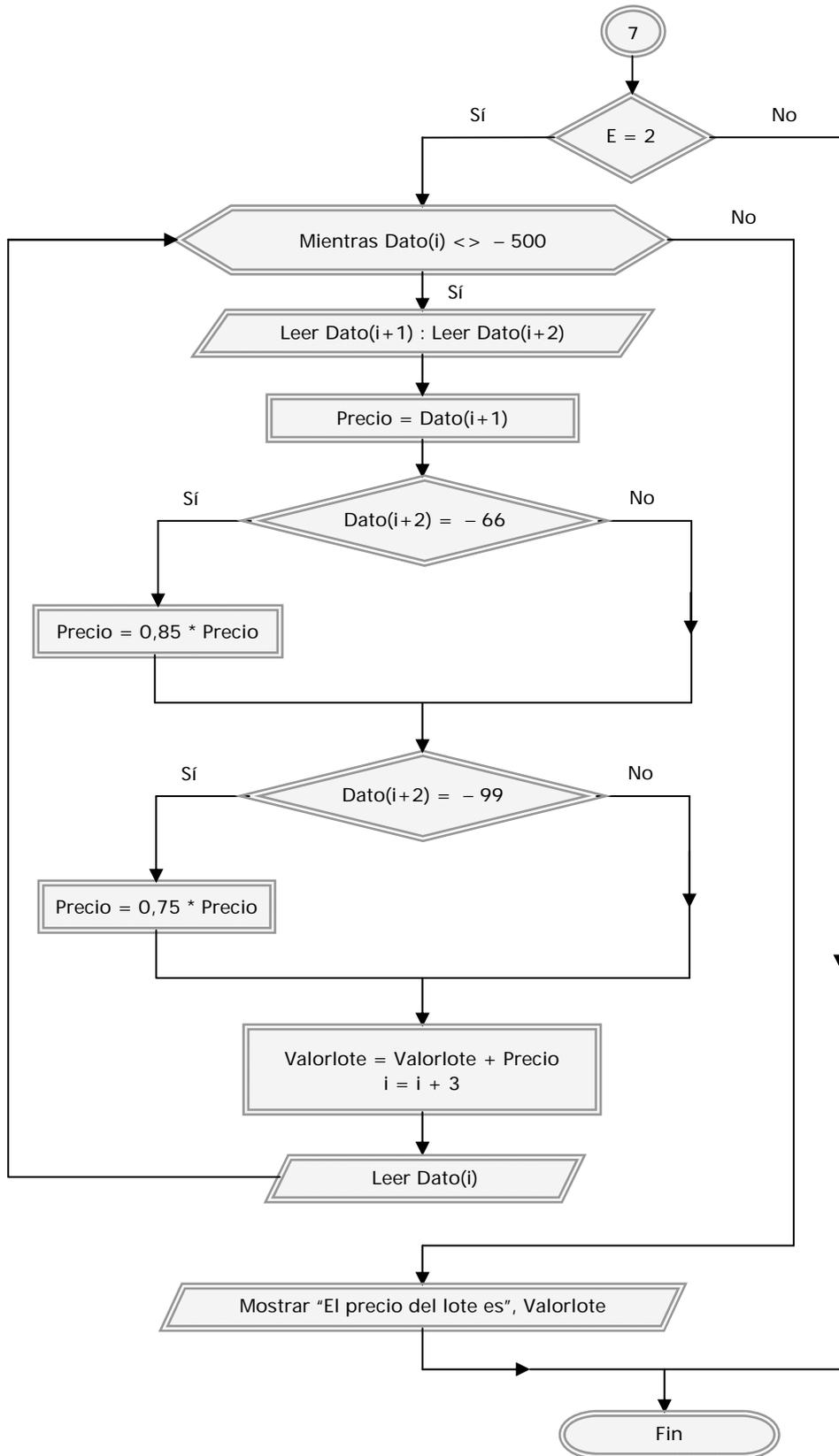
```

1. Inicio [Pseudocódigo http://www.aprenderaprogramar.com]
2. Mostrar "Elija una opción. 1 = Precio de un libro. 2 = Precio del lote"
3. Pedir E
4. i = 1 : Leer Dato(i)
5. Si E = 1 Entonces
    5.1 Mostrar "¿Cuál es el ISBN?" : Pedir ISBN
    5.2 Mientras Dato(i) <> - 500 Hacer
        Si Dato(i) = ISBN Entonces
            Leer Dato(i + 1) : Leer Dato(i + 2)
            Precio = Dato(i + 1)
            Si Dato(i + 2) = - 66 Entonces
                Precio = 0,85 * Precio
            FinSi
            Si Dato(i + 2) = - 99 Entonces
                Precio = 0,75 * Precio
            FinSi
            Mostrar "El precio del libro con ISBN", ISBN, "es de", Precio
            Encontrado = Verdadero
        FinSi
        i = i + 3 : Leer Dato(i)
    Repetir
    5.3 Si Encontrado = Falso Entonces
        Mostrar "No se ha encontrado el ISBN deseado"
    FinSi
    FinSi
6. Si E = 2 Entonces
    6.1 Mientras Dato(i) <> - 500 Hacer
        Leer Dato(i + 1)
        Leer Dato(i + 2)
        Precio = Dato(i + 1)
        Si Dato(i + 2) = - 66 Entonces
            Precio = 0,85 * Precio
        FinSi
        Si Dato(i + 2) = - 99 Entonces
            Precio = 0,75 * Precio
        FinSi
        Valorlote = Valorlote + Precio
        i = i + 3 : Leer Dato(i)
    Repetir
    6.2 Mostrar "El precio del lote es", Valorlote
    FinSi
7. Fin
  
```

**Diagrama de flujo:**







Para un caso supuesto con los siguientes datos:

Dato	Contenido	Dato	Contenido
1	55	6	- 66
2	10	7	88
3	- 99	8	12
4	35	9	- 33
5	7	10	- 500

El programa daría lugar a estas soluciones:

Opción 1 (Precio de un libro), según ISBN introducido.

El precio del libro con ISBN 55 es de 7,5 €

El precio del libro con ISBN 35 es de 5,95 €

El precio del libro con ISBN 88 es de 12 €

Opción 2 (Precio del lote).

El precio del lote es 25,45 €

**Comentarios:** A medida que los programas se hacen más complejos, la posibilidad de variantes va aumentando hasta ser inabarcable. Empecemos por cuestionar asuntos relacionados con el algoritmo.

a) **¿Por qué poner dos bucles similares y no sólo uno?**

Pues depende de los criterios del programador y de la información de que se disponga. A falta de conocimientos específicos, programamos para un caso general. Y en este caso, se ha valorado que el archivo puede tener tanto 10 datos como 10000.

Para el caso de que el usuario quiera mirar el precio de un libro se ha previsto el bucle que no lee todos los datos: extrae únicamente datos de ISBN hasta que encuentra el deseado. Así los datos que se leen son 1, 1 + 3, 1 + 3 + 3, ..., es decir, 1, 4, 7, 10, 13, 16, 19, ... En definitiva si en el fichero hay  $n$  datos estamos extrayendo y leyendo  $((n - 1) / 3) + 1$  datos. En cambio, para calcular el valor del lote hemos de extraer y leer todos los datos porque contienen información que nos es necesaria: el ISBN para saber si se han terminado los datos, el precio para saber el precio base y el señalero para saber el descuento. Ahora supongamos que en extraer / leer un dato se tardan 0,05 segundos ¿Cómo afectaría usar un tipo de bucle u otro?

Datos en fichero	Tiempo de proceso con bucle largo en segundos (leer n datos)	Tiempo de proceso con bucle corto en segundos, leer $((n - 1) / 3) + 1$ datos
10	0,5	0,2
100	5	1,7
1000	50	16,7
10000	500 (8 min 20 s)	166,7 (2 min 47 s)

La proporción entre el número de datos (y tiempo) a extraer en un caso y otro es de:

$$\frac{n}{\frac{n-1}{3}+1} = 3 \cdot \left( \frac{n}{n+2} \right)$$

Para valores de  $n$  grandes,  $(n / (n + 2))$  tiende a 1 resultando que el número de datos a extraer y el tiempo empleado es de aproximadamente tres veces según se elija una vía u otra. Dado que el usuario puede querer calcular sólo el precio de un libro: ¿Para que llevarlo por el camino largo pudiendo ir por el corto?

En este caso no hemos trabajado con datos reales y por otro lado, por fortuna, los ordenadores son muy rápidos y disponemos de más herramientas para resolver problemas. Lo que sí nos parece interesante es quedarnos con el fondo del asunto. En ocasiones, lo que puede ser un ahorro de código se convierte en una pérdida de eficiencia. Y esto no interesa. Un conocido locutor de radio solía decir: “Buscad la belleza”. En nuestro caso, “Busquemos la eficiencia”.

**b) ¿Por qué usar Si ... Entonces en vez de Según ... Caso para el flujo principal del programa?**

El uso de *Según ... Caso* sería una alternativa válida, aunque recordemos que siempre es sustituible. Siendo así, nos interesa dominar plenamente el uso de *Si ... Entonces* pudiendo considerar secundario el *Según ... Caso*.

**c) ¿No es conveniente introducir una opción de “salir” en el menú que permita no ejecutar nada?**

Siempre será conveniente dar una opción a salir de un menú. Lo estudiaremos en el siguiente apartado relativo a “Entradas controladas por el usuario”.

Por último haremos una breve reflexión relativa al contenido inicial de las variables. Venimos suponiendo que cuando el programa se inicia todas las variables valen cero o están vacías. Esto puede ser cierto o no. Es decir, podría darse el caso de trabajar con un sistema informático que guarde el contenido de las variables una vez termina el programa y las conserve cuando se inicia otra vez, o que no reconozca contenido alguno ni a la variable misma si no se le ha asignado contenido. Al respecto: hay que conocer los sistemas y lenguajes con los que se trabaja.

Supongamos que cerramos este programa con un bucle de modo que una vez se llega al final se vuelve al principio y vuelve a pedir la elección de una opción (el programa no termina por sí solo). ¿Qué ocurrirá? Que el programa parece que funciona pero realmente no lo hace bien por un problema con el contenido inicial de variables. Así podrían ocurrir cosas como que si tras calcular el precio de un lote obtenemos 750 €, si repetimos el cálculo nos devuelva 1500 € porque el acumulador partía de su valor anterior y no fue establecido a cero.

**Próxima entrega: CU00172A**

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