



aprenderaprogramar.com

Ejercicios resueltos con acumuladores. Pseudocódigo y diagramas de flujo. Bucle desde-siguiente (for-next). (CU00160A)

Sección: Cursos

Categoría: Curso Bases de la programación Nivel I

Fecha revisión: 2024

Autor: Mario R. Rancel

Resumen: Entrega nº 59 del Curso Bases de la programación Nivel I

24

EJERCICIO

En un fichero tenemos las notas de 35 alumnos de una clase en un array $Nota(1), Nota(2), \dots, Nota(35)$, establecidas entre cero y 10. Se desea desarrollar el pseudocódigo y diagrama de flujo para un programa que determine la nota media.

SOLUCIÓN

Pseudocódigo:

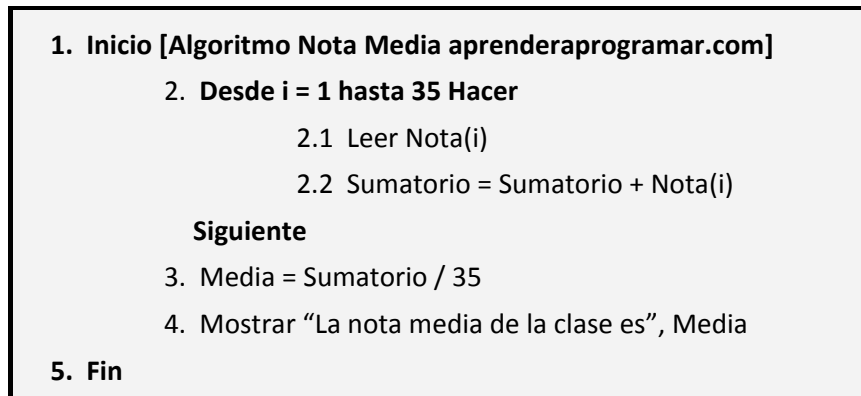
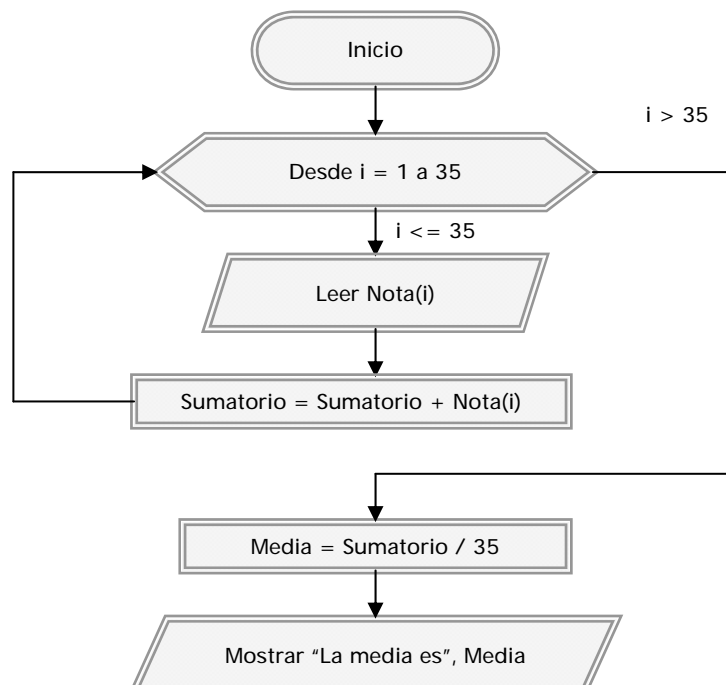


Diagrama de flujo:



Comentarios: El acumulador se alimenta con cada vuelta del bucle, pero no tiene que ver con el control del mismo.

EJERCICIO

Una central distribuye cal hacia diferentes almacenes sucursales. Disponen de un muelle de carga a donde van llegando sacas de cal de entre 3000 y 9000 Kgs, con pesos variables en función de las circunstancias de la producción. La empresa dispone de una flota de camiones con capacidades de carga de entre 18000 y 28000 Kgs.

Se pretende establecer un protocolo consistente en cargar 20 camiones diarios. Cada camión se quiere cargar como máximo a su límite de capacidad debiendo partir si con la siguiente saca en la línea de producción se fuera a exceder su capacidad.

La empresa quiere desarrollar un programa que le pida al operario encargado de carga la capacidad del camión y el peso de las sacas, indicándole si debe cargar la saca o despachar el camión para comenzar a cargar otro.

SOLUCIÓN

Organización previa de variables:

Variable	Descripción - Uso
i, j, k	Uso como contadores
Capacidad	Almacena la capacidad del camión actual en Kgs
PesoSaca	Almacena el peso de la saca actual en Kgs
Cargado	Acumulador con el peso cargado actualmente en el camión

Pseudocódigo:

1. Inicio [Algoritmo Control de Carga de Camiones aprenderaprogramar.com]

2. $i = 1 : j = 0$ [Contadores]

3. Desde $i = 1$ hasta 20 Hacer [Para 20 camiones]

3.1 Mostrar "Introduzca capacidad del camión número", i , "en Kgs"

3.2 Pedir Capacidad

3.3 Mientras (Cargado + PesoSaca) <= Capacidad Hacer

3.3.1 **Si** PesoSaca > 0 **Entonces**

Mostrar "Cargue la saca número", j

Cargado = Cargado + PesoSaca

FinSi

3.3.2 $j = j + 1$

3.3.3 Mostrar "Introduzca el peso de la saca número", j

3.3.4 Pedir PesoSaca

Repetir

3.4 Si $i < 20$ **Entonces**

Mostrar "No cargue todavía. Despache el camión número", i ,

"y ponga a cargar el camión número", $i + 1$

FinSi

3.5 Cargado = 0

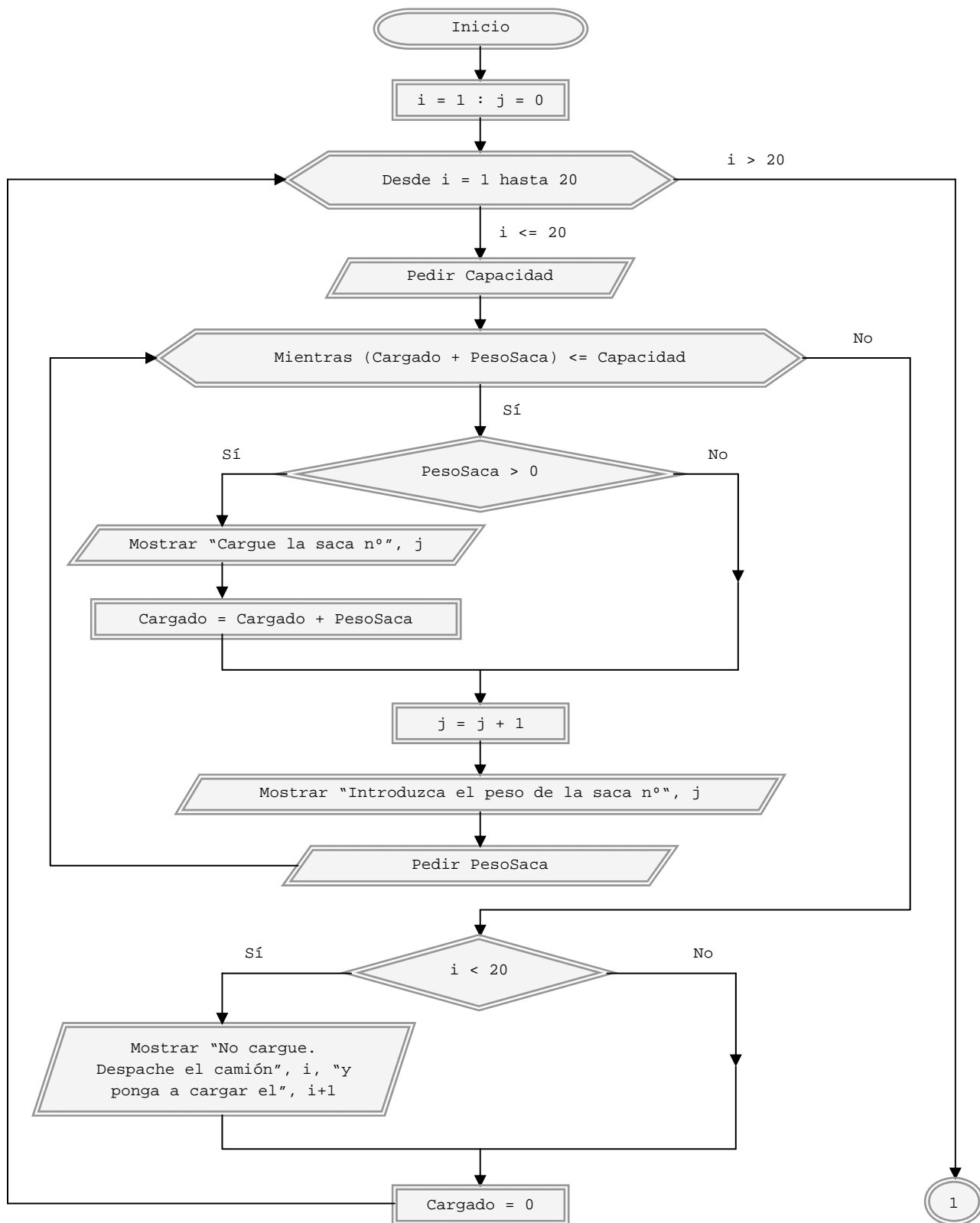
Siguiente

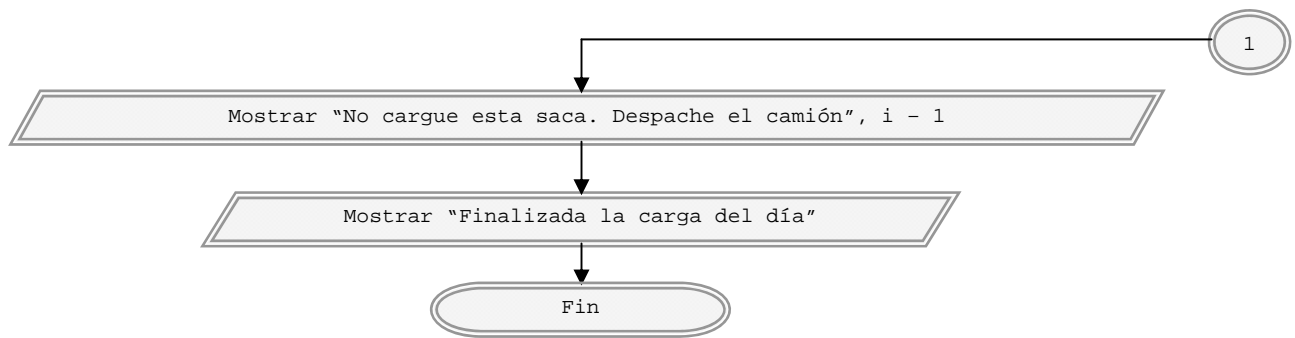
4. Mostrar "No cargue esta saca. Despache el camión", $i - 1$

5. Mostrar "Finalizada la carga del día"

6. Fin

Diagrama de flujo aprenderaprogramar.com:





Caso numérico: Supongamos que los datos reales de un día de trabajo son los siguientes.

Camión nº	Capacidad Kgs	Saca nº	Capacidad Kgs	Saca nº	Capacidad Kgs
1	22000	1	3000	21	4000
2	18000	2	5000	22	5000
3	18000	3	9000	23	6000
4	20000	4	3000	24	3000
5	28000	5	3000	25	3000
6	20000	6	8000	26	9000
7	22000	7	9000	27	4000
8	18000	8	7000	28	8000
9	20000	9	7000	29	3000
10	20000	10	9000	30	5000
11	20000	11	4000	31	5000
12	28000	12	5000	32	3000
13	22000	13	9000	33	9000
14	20000	14	9000	34	5000
15	22000	15	3000	35	6000
16	18000	16	4000	36	6000
17	20000	17	3000	37	7000
18	20000	18	5000	38	3000
19	22000	19	4000	39	7000
20	28000	20	3000	40	5000

Entramos al algoritmo. Para el camión 1 nos pide la capacidad y tecleamos 22000. Entra en el bucle mientras con $Cargado + Peso = 0$. $PesoSaca$ no es mayor que cero por lo que hace $j = j + 1 = 1$ y nos pide el peso de la saca 1. Tecleamos 3000. Vuelve al comienzo del bucle y $Cargado + PesoSaca = 3000$ inferior a la capacidad del camión. Ahora $PesoSaca > 0$ y nos muestra "Cargue la saca número 1" y hace $Cargado = Cargado + PesoSaca = 0 + 3000 = 3000$. $j = j + 1 = 2$. Nos pide el peso de la saca 2 y tecleamos 5000.

Vuelve al origen del bucle y ahora $Cargado + PesoSaca = 3000 + 5000 = 8000$, todavía inferior a la capacidad del camión (22000 Kgs). $PesoSaca$ es mayor que cero y nos muestra "Cargue la saca número 2" y hace $Cargado = Cargado + PesoSaca = 3000 + 5000 = 8000$. Hace $j = j + 1 = 3$, y nos pide el peso de la saca número 3. Tecleamos 9000. Ahora $Cargado + Peso$ es $8000 + 9000 = 17000 < 22000$. $PesoSaca > 0$ y nos muestra "Cargue la saca número 3" y hace $Cargado = 8000 + 9000 = 17000$. Hace $j = j + 1 = 4$ y nos pide el peso de la saca 4. Tecleamos 3000. Vuelve al inicio del bucle, $17000 < 22000$, $PesoSaca > 0$ y muestra "Cargue la saca número 4". Hace $Cargado = 17000 + 3000 = 20000$, $j = j + 1 = 5$ y pide el peso de la saca 5. Tecleamos 3000. Retornamos al origen de bucle y $Cargado + PesoSaca = 20000 + 3000 = 23000$. Ahora no se cumple $Cargado + PesoSaca \leq Capacidad$ ya que excedemos la capacidad en 1000 Kgs por lo que no se entra en el bucle. El flujo se deriva para mostrar "No cargue todavía. Despache el camión número 1 y ponga a cargar el camión número 2". Se hace el valor cargado igual a cero y se regresa al bucle de camiones tomando el valor $i = 2$. El proceso se repite dando lugar a estos resultados.

Camión nº	Número de las sacas que carga	Carga que lleva en Kgs
2	5 y 6	11000
3	7 y 8	16000
4	9, 10 y 11	20000
5	12, 13, 14 y 15	26000
6	16, 17, 18, 19 y 20	19000
7	21, 22, 23, 24 y 25	21000
8	26 y 27	13000
9	28, 29 y 30	16000
10	31, 32 y 33	17000
11	34, 35 y 36	17000
Etc.	Etc.	Etc.

El proceso continuará hasta alcanzar el llenado de los 20 camiones, momento en el que se saldría del bucle exterior para mostrar "Finalizada la carga del día".

Comentarios: No es un problema de optimización sino de control de valores acumulados. El fondo del asunto está en lograr que los camiones carguen sin superar nunca su carga máxima, incluso aunque se queden lejos del óptimo (que sería ir llenos).

El problema es demostrativo del uso de contadores, acumuladores y bucles anidados. Además podemos buscarle el por qué al uso de un tipo de bucle u otro. Si analizamos la instrucción empleada y el número de repeticiones observamos:

- La instrucción *Desde ... Siguiete* se ha usado para un bucle en el que el número de repeticiones está predeterminado (en este caso, 20).
- La instrucción *Mientras ... Hacer* se ha usado para un bucle en el que el número de repeticiones es desconocido, al depender de variables que no podemos controlar como el peso de las sacas.

Podemos decir que es “habitual” el uso de la instrucción *Desde ... Siguiete* para bucles con número de repeticiones prefijado y *Mientras ... Hacer* para bucles con número de repeticiones variable, pero no siempre es así. No existe ninguna regla o norma que obligue al uso de ciertas instrucciones. Cada programador conjuga los recursos y la experiencia para construir los algoritmos más eficaces posibles.

Próxima entrega: CU00161A

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