Ciclos Combinados y Corte de Control

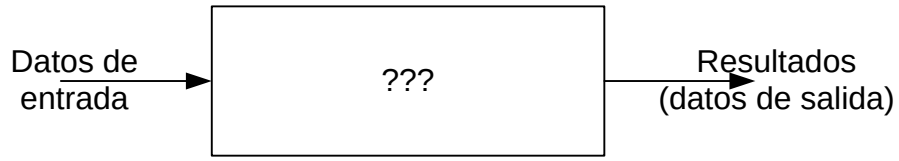
Como se mencionó durante el desarrollo de la materia, las estructuras de programación son la **secuencia**, la **decisión** (simple y múltiple), y los **ciclos** (exacto e inexacto). También se señaló que con estas estructuras puede resolverse cualquier problema, y que pueden combinarse sin ninguna restricción.

En los TPs vistos hemos trabajado con todas las estructuras, pero, con excepción de algunos ejercicios de la práctica 4, no hemos resuelto problemas que nos exijan la combinación de ciclos. Este tipo de ejercicios son los que se presentan en el TP 5.

Ahora, ¿cuál es la razón por la que debemos combinar ciclos?

La decisión sobre el tipo de estructura a utilizar para la resolución de un problema no puede definirse a priori; no es algo que podamos establecer arbitrariamente: es la naturaleza del problema, sus características, la que va a determinar si es necesario o no utilizar decisiones, ciclos, o cualquier combinación entre ellos (la secuencia siempre estará presente).

La clave entonces, está en analizar cuidadosamente el enunciado, y esto se consigue aplicando la metodología de análisis explicada, que plantea que sea cual sea la complejidad del problema a resolver podemos representarlo con el siguiente esquema:



Análisis de los datos de entrada: entender cuál es la organización de los datos de entrada es central para determinar la estructura general del programa. Debemos tener siempre en claro que los problemas se resuelven a partir de los datos de entrada: no podemos alterarlos, ni modificar el orden en el que se presentan, ni agregar datos que no existen en el enunciado; sí pueden presentarse casos en que no sea obligatorio ingresar todos los datos (no es necesario ingresar datos que no se utilicen para la resolución del problema). Además, en todos los casos los datos de entrada deben ingresarse sólo una vez, independientemente de que el ejercicio pida varios datos de salida.

Por lo general en los ejercicios se habla de conjuntos o listas de números a analizar, lotes de registros, o lotes divididos en sublotes. Para resolver el último caso es probable que sea necesario utilizar más de un ciclo, pero esto sólo podrá afirmarse luego de analizar los datos de salida que el programa exige.

Análisis de los datos de salida: los datos de salida son la información que el programa proporciona, esto es, procesa los datos de entrada para generar la información requerida.

El análisis de la/s salida/s que el programa debe generar, sumado al anterior (las entradas), determinará la estructura general del programa. Saber cuántas son, y en que momento entregarlas son las incógnitas a resolver. Veamos ejemplos:

1. a) Una empresa tiene un lote de registros con la producción de las 50 máquinas que posee por cada día del mes de marzo. Cada registro tiene:

- Nº de máquina (1 a 50)
- Dia (1 a 31)
- Cantidad de piezas producidas totales (entero).
- Cantidad de piezas defectuosas (entero).

El lote está agrupado –no ordenado – por número de máquina. En los sublotes correspondientes a cada máquina los registros se encuentran ordenados por día.
Se pide desarrollar un programa que calcule e informe:

a) El/los número/s de máquina/s y el/los día/s en que la cantidad de piezas defectuosas sea mayor a 10.
b) La cantidad total de piezas producidas por cada máquina
c) El promedio general de piezas producidas por día entre todas las máquinas durante todo el mes.
d) La máquina que menos piezas produjo el día 15 de marzo (entre las máquinas que produjeron piezas).

Datos de entrada:
50 sublotes que contienen los datos relevados durante el mes de marzo para cada máquina. Cada sublote contiene 31 registros pertenecientes a una misma máquina; cada registro tiene la información de la producción de un día en particular (nº de máquina, día, cantidad de piezas producidas y cantidad de piezas defectuosas).

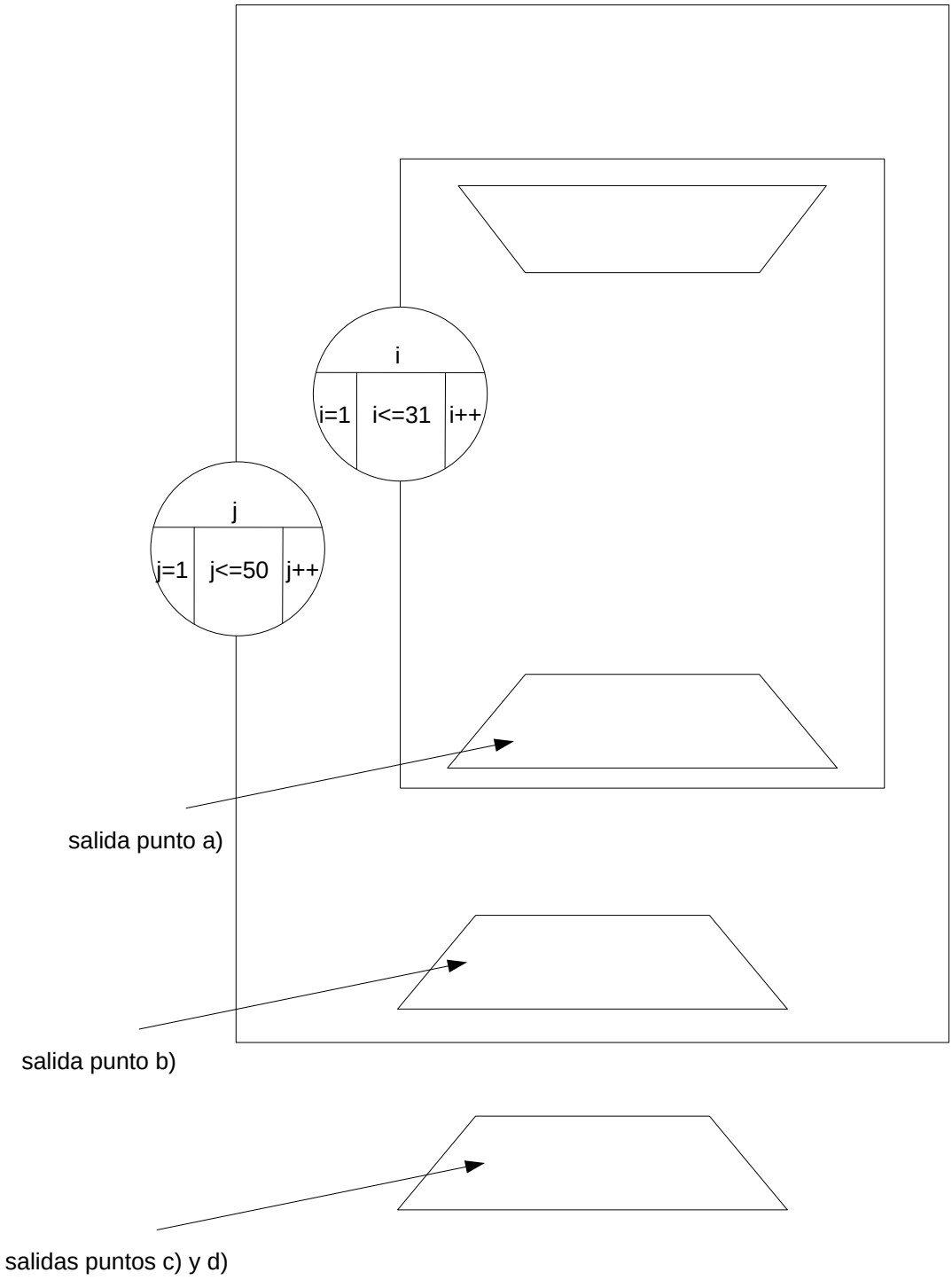
El enunciado dice que el lote de datos a procesar (compuesto por 50 sublotes) está agrupado por número de máquina; esto quiere decir que todos los registros pertenecientes a una misma máquina están todos juntos. Se aclara que no están ordenados, lo que indica que no necesariamente van a aparecer los sublotes ordenados de menor a mayor por número de máquina. Lo que sí está ordenado son los sublotes: dentro de ellos la información está ordenada por día.

Datos de salida:

a) No es posible decir cuántas resultados hay que informar. Hay que analizar cada registro, preguntarse si la cantidad defectuosa es mayor a 10, y si es así informar el número de máquina y el día.
b) Se deben informar 50 valores, uno por cada máquina. Cada valor debe informarse luego de procesar todos los registros de una máquina.
c) Se debe informar 1 solo valor, luego de procesar todos los registros de todas las máquinas.
d) Se debe informar 1 solo valor, luego de procesar todos los registros de todas las máquinas. Para cada máquina hay que averiguar cuántas piezas se produjo el día 15, y compararlas con las producidas por las otras máquinas.

Estructura general del programa:
La información de entrada está organizada en 50 sublotes de 31 registros. Para resolver el punto b) y el punto d) es necesario procesar cada sublote, por lo que la estructura general contará con 2 ciclos exactos: el externo de 50 (todas las máquinas), y el interno de 31 (para una máquina en particular).

El punto a) se debe informar dentro del ciclo interno; el punto b) fuera del ciclo interno pero dentro del externo, y los puntos c) y d) antes de finalizar el programa, fuera de ambos ciclos.
El ingreso de datos se deberá realizar dentro del ciclo interno. Como los sublotes están ordenados por día, no será necesario ingresar el día, ya que coincidirá siempre con el valor de la variable que maneja el ciclo.



1. b) Igual al anterior, pero el lote no contiene registros para aquellos días en los que la máquina no produjo piezas. Se supone que todas las máquinas produjeron piezas al

menos en un día del mes. El fin de los registros de cada máquina se indica con un día igual a 0.

Datos de entrada:
50 sublotes que contienen los datos relevados durante el mes de marzo para cada máquina. Cada sublote contiene entre 1 a 31 registros pertenecientes a una misma máquina: no se sabe cuántos son porque no existen registros para los días en los que la máquina no trabajo; cada registro tiene la información de la producción de un día en particular (nº de máquina, día, cantidad de piezas producidas y cantidad de piezas defectuosas).

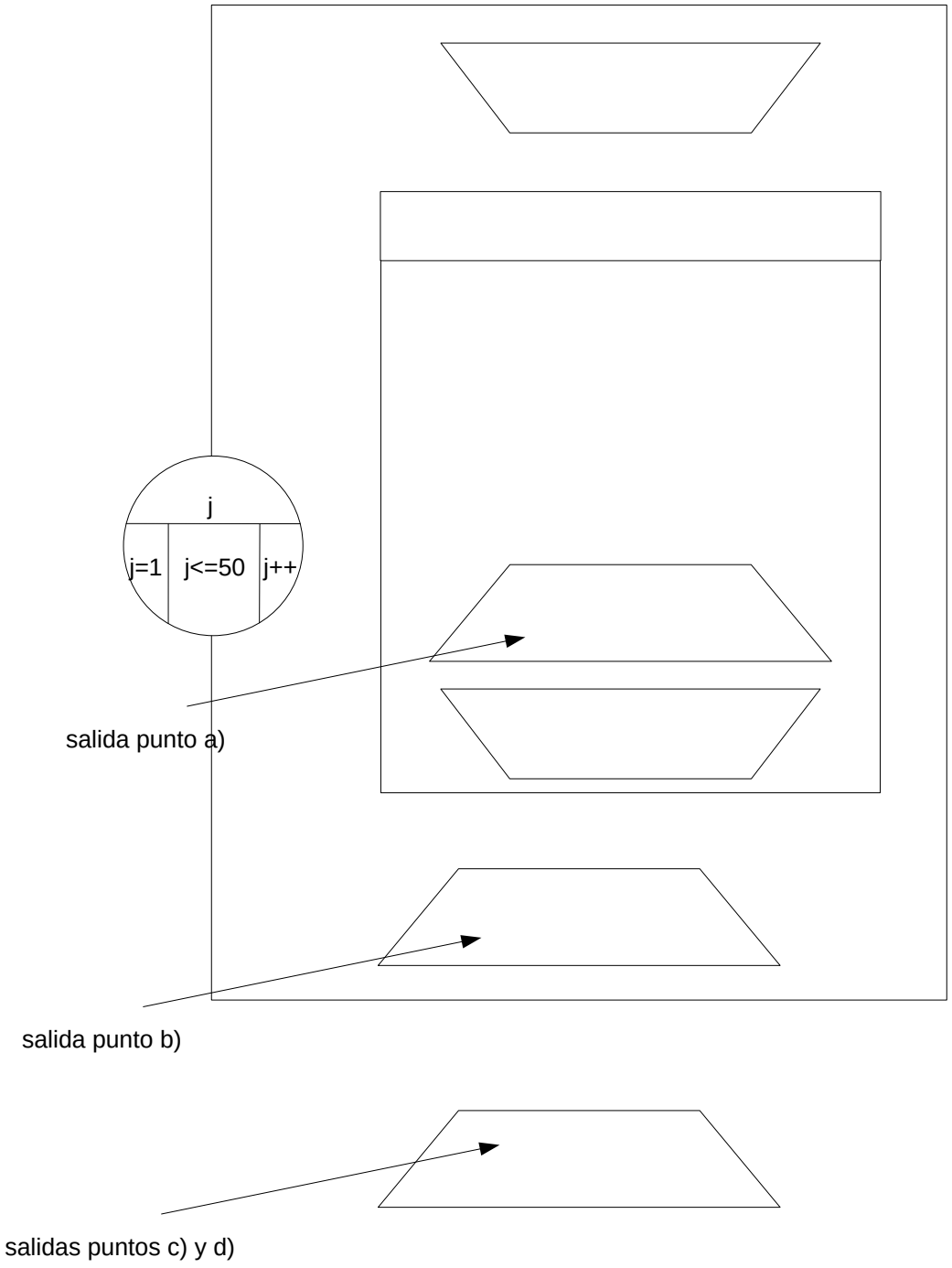
El enunciado dice que el lote de datos a procesar (compuesto por 50 sublotes) está agrupado por número de máquina; esto quiere decir que todos los registros pertenecientes a una misma máquina están todos juntos. Se aclara que no están ordenados, lo que indica que no necesariamente van a aparecer los sublotes ordenados de menor a mayor por número de máquina. Lo que sí está ordenado son los sublotes: dentro de ellos la información está ordenada por día, pero no necesariamente hay un registro por día.

Datos de salida: igual al anterior. La única diferencia es que en el punto d) hay que tener en cuenta solamente las máquinas que produjeron piezas ese día.

Estructura general del programa:
La información de entrada está organizada en 50 sublotes; no se sabe cuántos registros tiene cada sublote. Para resolver el punto b) y el punto d) es necesario procesar cada sublote, por lo que la estructura general contará con 2 ciclos: el externo un ciclo exacto de 50 (todas las máquinas), y el interno un ciclo inexacto que corta cuando se ingresa un día igual a 0.

El punto a) se debe informar dentro del ciclo inexacto interno; el punto b) fuera del ciclo interno pero dentro del externo, y los puntos c) y d) antes de finalizar el programa, fuera de ambos ciclos.

El ingreso de datos se deberá realizar como se muestra en el diagrama. A pesar de que los sublotes están ordenados por día, será necesario ingresar el día, ya que puede que no existan registros para algún día, y porque se lo utiliza para que se corte el ciclo inexacto interno.


j
j=1
j<=50
j++
salida punto a)
salida punto b)
salidas puntos c) y d)

1. c) Igual al 1.a) pero puede que en el lote no existan registros para alguna/s de las máquinas. Por cada máquina hay un registro por día. El fin del lote se informa con un número de máquina negativa.

Datos de entrada:
Un número de sublotes indeterminado (entre 1 y 50) que contienen los datos relevados durante el mes de marzo para cada máquina. Cada sublote contiene 31 registros pertenecientes a una misma máquina; cada registro tiene la información de la producción de un día en particular (nº de máquina, día, cantidad de piezas producidas y cantidad de piezas defectuosas). El fin del lote se indica con un número de máquina negativa

El enunciado dice que el lote de datos a procesar (compuesto por un número indeterminado de sublotes) está agrupado por número de máquina; esto quiere decir que todos los registros pertenecientes a una misma máquina están todos juntos. Se aclara que no están ordenados, lo que indica que no necesariamente van a aparecer los sublotes ordenados de menor a mayor por número de máquina. Lo que sí está ordenado son los sublotes: dentro de ellos la información está ordenada por día, y hay un registro por día.

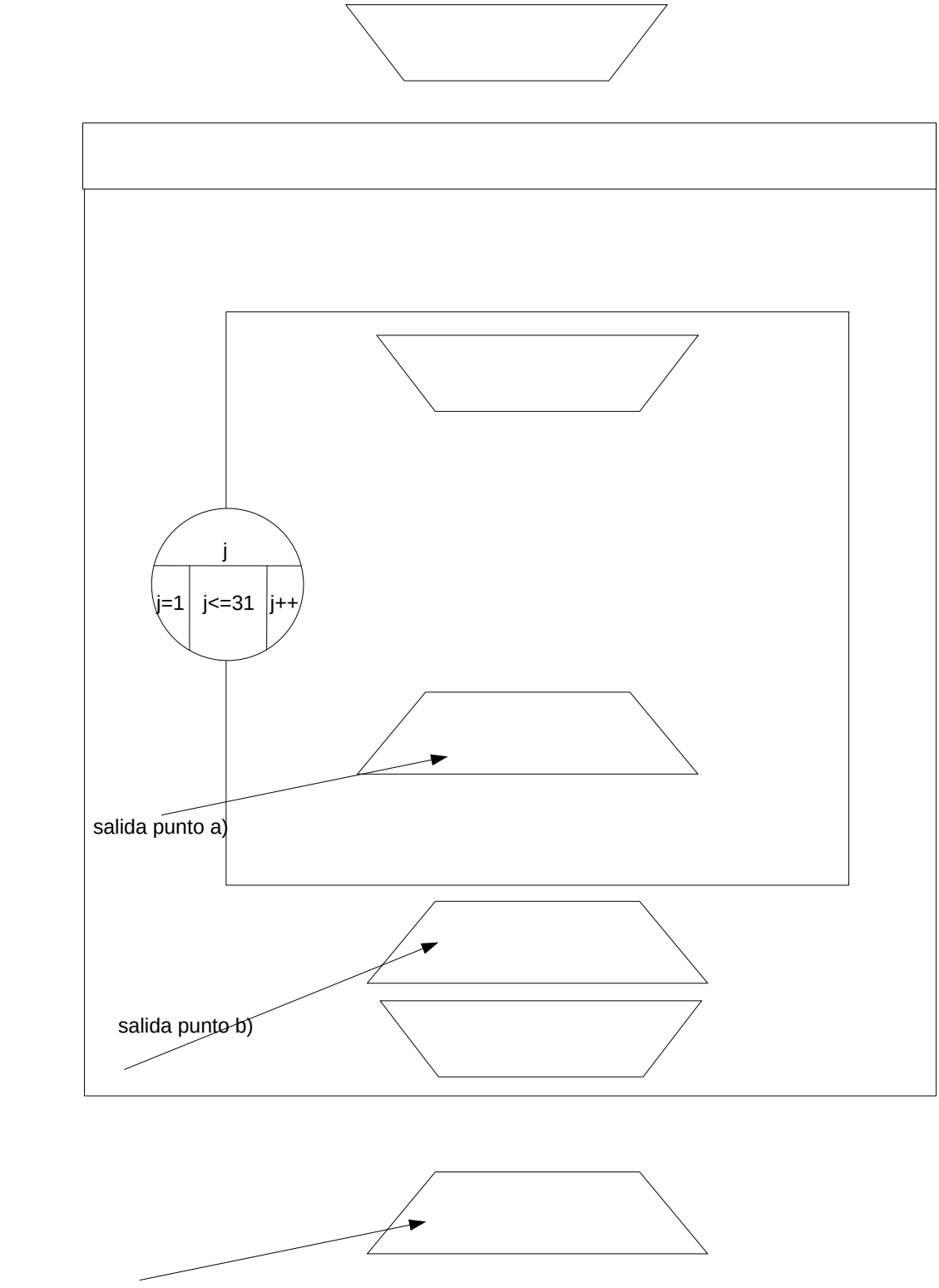
Datos de salida:

a) No es posible decir cuántas resultados hay que informar. Hay que analizar cada registro, preguntarse si la cantidad defectuosa es mayor a 10, y si es así informar el número de máquina y el día.
b) No se sabe cuántos valores hay que informar porque no se saben cuántas máquinas hay; se deberá informar un valor por cada máquina. Cada valor debe informarse luego de procesar todos los registros de una máquina.
c) Se debe informar 1 solo valor, luego de procesar todos los registros de todas las máquinas.
d) Se debe informar 1 solo valor, luego de procesar todos los registros de todas las máquinas. Para cada máquina hay que calcular cuántas piezas se produjo el día 15, y compararlas con las producidas por las otras máquinas.

Estructura general del programa:
La información de entrada está organizada en un número indeterminado de sublotes; dentro de cada sublote hay 31 registros. Para resolver el punto b) y el punto d) es necesario procesar cada sublote, por lo que la estructura general contará con 2 ciclos: el externo un ciclo inexacto que corte cuando aparezca un número de máquina negativo, y el interno un ciclo exacto de 31.

El punto a) se debe informar dentro del ciclo exacto interno; el punto b) fuera del ciclo interno pero dentro del externo, y los puntos c) y d) antes de finalizar el programa, fuera de ambos ciclos.

El ingreso de datos se deberá realizar como se muestra en el diagrama. Como los sublotes están ordenados por día, no será necesario ingresar el día, ya que coincidirá con la variable que maneja el ciclo exacto interno.



1. d) Igual al 1.a), pero el lote no contiene registros para aquellas máquinas que no produjeron piezas en todo el mes y para aquellas que figuran en el lote sólo se registran los días en los que la producción fue mayor a 0. El fin del lote se informa con un número de máquina negativa.

Datos de entrada:
Un número de sublotes indeterminado (entre 1 y 50) que contienen los datos relevados durante el mes de marzo para cada máquina. Cada sublote contiene un número indeterminado de registros (entre 1 y 31) pertenecientes a una misma máquina; cada registro tiene la información de la producción de un día en particular (nº de máquina, día, cantidad de piezas producidas y cantidad de piezas defectuosas). El fin del lote se indica con un número de máquina negativa.

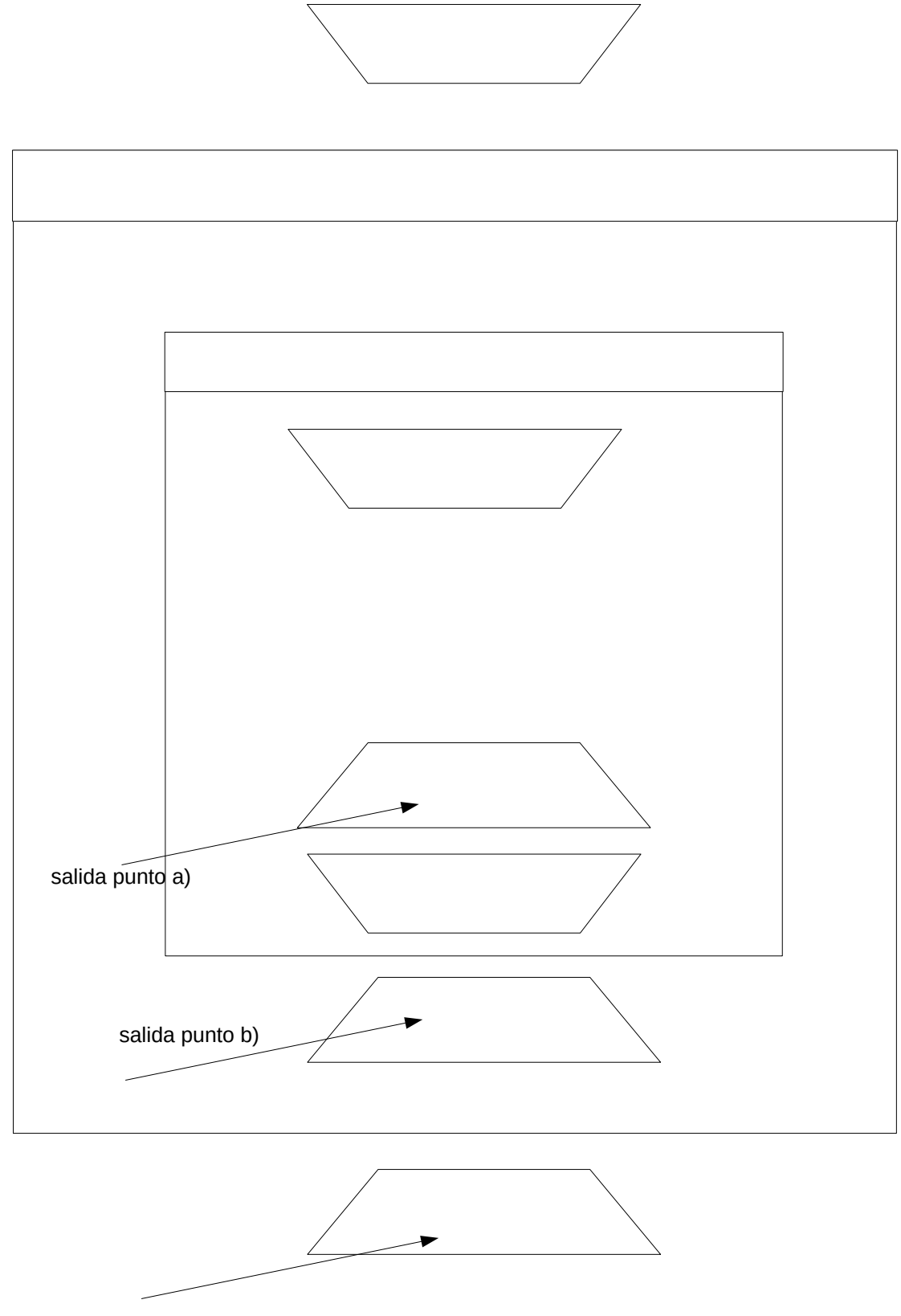
El enunciado dice que el lote de datos a procesar (compuesto por un número indeterminado de sublotes) está agrupado por número de máquina; esto quiere decir que todos los registros pertenecientes a una misma máquina están todos juntos. Se aclara que no están ordenados, lo que indica que no necesariamente van a aparecer los sublotes ordenados de menor a mayor por número de máquina. Lo que sí está ordenado son los sublotes: dentro de ellos la información está ordenada por día, pero no necesariamente hay un registro por día.

Datos de salida:
a) No es posible decir cuántas resultados hay que informar. Hay que analizar cada registro, preguntarse si la cantidad defectuosa es mayor a 10, y si es así informar el número de máquina y el día.
b) No se sabe cuántos valores hay que informar porque no se saben cuántas máquinas hay; se deberá informar un valor por cada máquina. Cada valor debe informarse luego de procesar todos los registros de una máquina.
c) Se debe informar 1 solo valor, luego de procesar todos los registros de todas las máquinas.
d) Se debe informar 1 solo valor, luego de procesar todos los registros de todas las máquinas. Para cada máquina hay que calcular cuántas piezas se produjo el día 15, y compararlas con las producidas por las otras máquinas. Como no se sabe si las máquinas trabajaron el día 15, se debe tomar en cuenta solamente aquellas máquinas que produjeron piezas ese día. Se supone que al menos una de las máquinas trabajó el día 15.

Estructura general del programa:
La información de entrada está organizada en un número indeterminado de sublotes; dentro de cada sublote hay un número indeterminado de registros. Para resolver el punto b) y el punto d) es necesario procesar cada sublote, por lo que la estructura general contará con 2 ciclos inexactos: el externo que cortará cuando aparezca un número de máquina negativo, y el interno que cortará cuando se produzca un cambio de máquina.

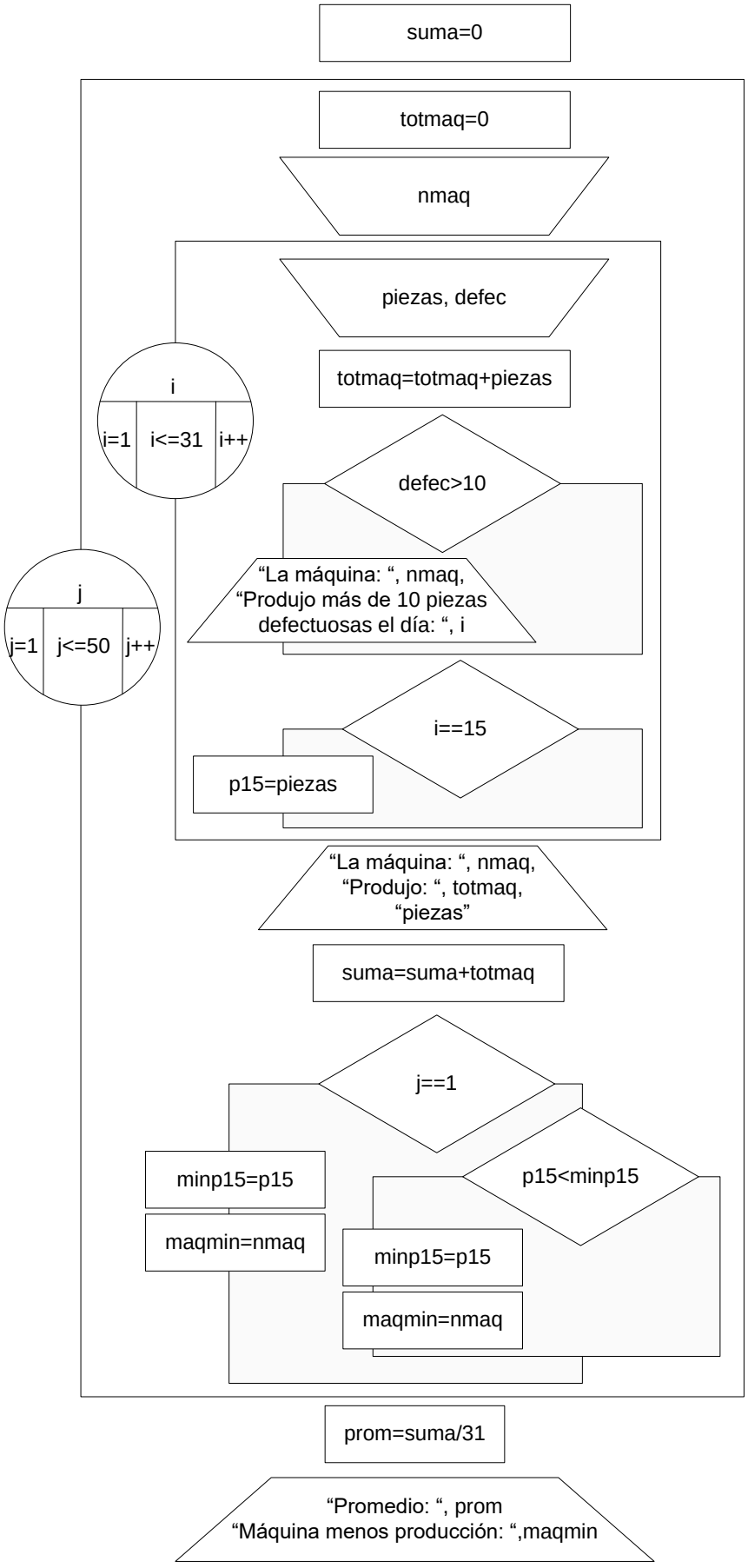
El punto a) se debe informar dentro del ciclo inexacto interno; el punto b) fuera del ciclo interno pero dentro del externo, y los puntos c) y d) antes de finalizar el programa, fuera de ambos ciclos.

El ingreso de datos se deberá realizar como se muestra en el diagrama. A pesar de que los sublotes están ordenados por día, deberán ingresarse el día, ya que no necesariamente habrá registros para todos los días.


salida punto a)
salida punto b)
salidas puntos c) y d)

Resolución

1a)


suma=0
totmaq=0
nmaq
piezas, defec
totmaq=totmaq+piezas
i
i=1
i<=31
i++
defec>10
"La máquina: ", nmaq,
"Produjo más de 10 piezas
defectuosas el día: ", i
j
j=1
j<=50
j++
i==15
p15=piezas
"La máquina: ", nmaq,
"Produjo: ", totmaq,
"piezas"
suma=suma+totmaq
j==1
minp15=p15
maqmin=nmaq
p15<minp15
minp15=p15
maqmin=nmaq
prom=suma/31
"Promedio: ", prom
"Máquina menos producción: ",maqmin

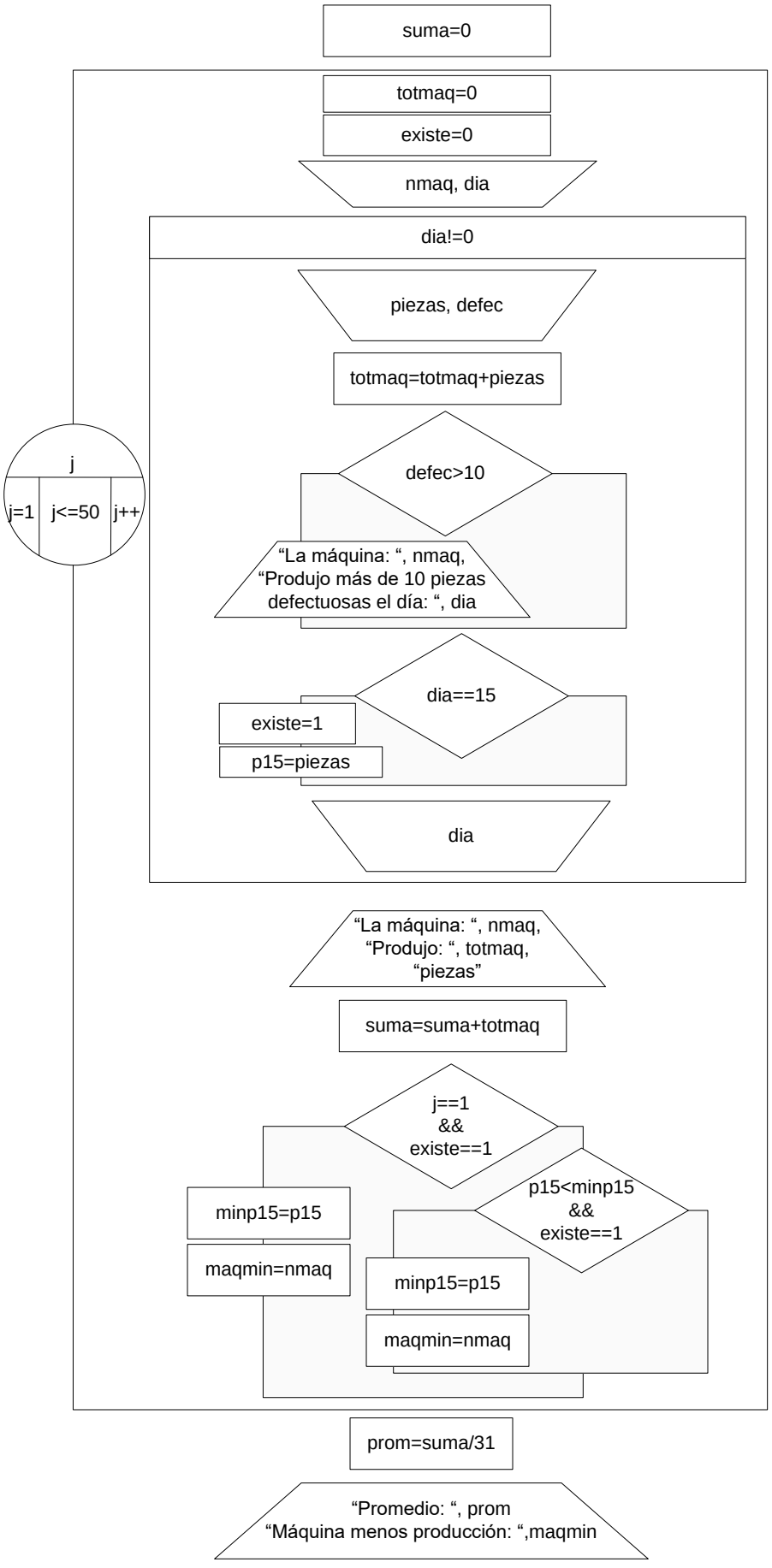
Codificación 1a)

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main()
{
int nmaq, piezas, defec;//para los datos de entrada
int i, j;//para los ciclos for
int suma, totmaq, p15;
int minp15, maqmin;
float prom;
suma=0;
for(j=1;j<=50;j++){
  totmaq=0;
  cout<<"INGRESE EL NUMERO DE MAQUINA: ";
  cin>>nmaq;
  for(i=1;i<=31;i++){
    cout<<"INGRESE LA CANTIDAD DE PIEZAS FABRICADAS: ";
    cin>>piezas;
    cout<<  "INGRESE LA CANTIDAD DE PIEZAS DEFECTUOSAS: ";
    cin>>defec;
    totmaq+=piezas;
    if(defec>10){
      cout<<"LA MAQUINA: "<<nmaq<<endl;
      cout<<"PRODUJO MAS DE 10 PIEZAS DEFECTUOSAS EL DIA: "<<i<<endl;
      }
    if(i==15){
      p15=piezas;
      }
    }
  cout<<"LA MAQUINA: "<<nmaq<<endl;
  cout<<"PRODUJO "<<totmaq<<" PIEZAS"<<endl;
  suma+=totmaq;
  if(j==1){
    minp15=p15;
    maqmin=nmaq;
    }
  else{
    if(p15<minp15){
      minp15=p15;
      maqmin=nmaq;
      }
    }
  }
prom=(float)suma/31;
cout<<"PROMEDIO: "<<prom;
```

```
cout<<"MAQUINA CON MENOR PRODUCCION EL DIA 15: "<<maqmin<<endl;
system("pause");
return 0;
}
```

1b)


suma=0
totmaq=0
existe=0
nmaq, dia
dia!=0
piezas, defec
totmaq=totmaq+piezas
j
j=1
j<=50
j++
defec>10
“La máquina: “, nmaq,
“Produjo más de 10 piezas
defectuosas el día: “, dia
dia==15
existe=1
p15=piezas
dia
“La máquina: “, nmaq,
“Produjo: “, totmaq,
“piezas”
suma=suma+totmaq
j==1
&&
existe==1
minp15=p15
maqmin=nmaq
p15<minp15
&&
existe==1
minp15=p15
maqmin=nmaq
prom=suma/31
“Promedio: “, prom
“Máquina menos producción: “,maqmin

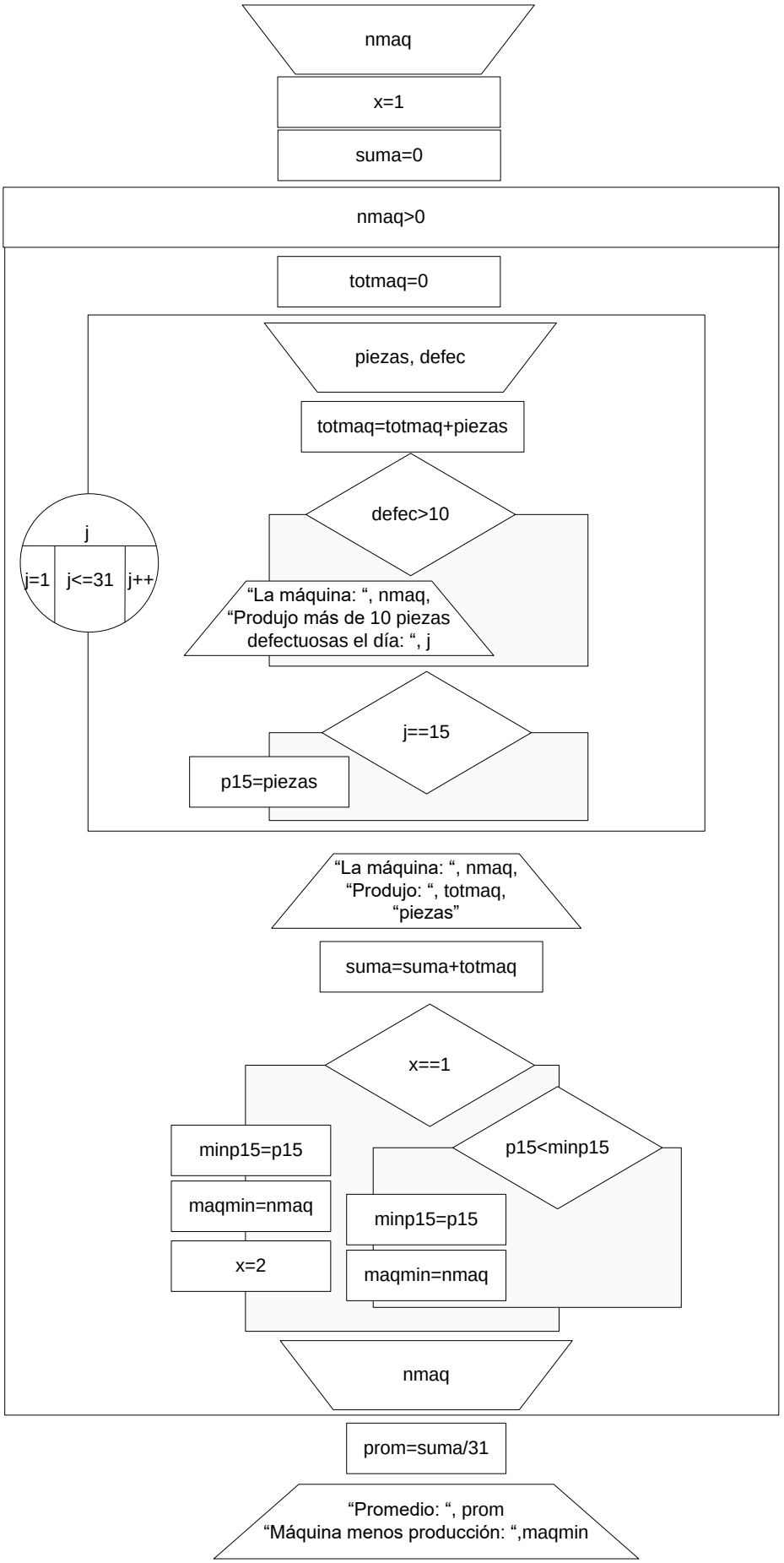
Codificación 1b)

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main()
{
int nmaq, piezas, defec, dia;//para los datos de entrada
int suma=0, totmaq, p15, j, existe, minp15, maqmin;
float prom;
for(j=1;j<=50;j++){
  totmaq=0;
  existe=0;
  cout<<"INGRESE EL NUMERO DE MAQUINA: ";
  cin>>nmaq;
  cout<<"INGRESE EL DIA: ";
  cin>>dia;
  while(dia!=0){
    cout<<"INGRESE LA CANTIDAD DE PIEZAS FABRICADAS: ";
    cin>>piezas;
    cout<<  "INGRESE LA CANTIDAD DE PIEZAS DEFECTUOSAS: ";
    cin>>defec;
    totmaq+=piezas;
    if(defec>10){
      cout<<"LA MAQUINA: "<<nmaq<<endl;
      cout<<"PRODUJO MAS DE 10 PIEZAS DEFECTUOSAS EL DIA: "<<dia<<endl;
      }
    if(dia==15){
      existe=1;
      p15=piezas;
      }
    cout<<"INGRESE EL DIA: ";
    cin>>dia;
    }
  cout<<"LA MAQUINA: "<<nmaq<<endl;
  cout<<"PRODUJO "<<totmaq<<" PIEZAS"<<endl;
  suma+=totmaq;
  if(j==1 && existe==1){
    minp15=p15;
    maqmin=nmaq;
    }
  else{
    if(p15<minp15 && existe==1){
      minp15=p15;
      maqmin=nmaq;
      }
    }
```

```
  }
prom=(float)suma/31;
cout<<"PROMEDIO: "<<prom<<endl;
cout<<"MAQUINA CON MENOR PRODUCCION EL DIA 15: "<<maqmin<<endl;
system("pause");
return 0;
}
```

1c)


nmaq
x=1
suma=0
nmaq>0
totmaq=0
piezas, defec
totmaq=totmaq+piezas
defec>10
j
j=1
j<=31
j++
"La máquina: ", nmaq,
"Produjo más de 10 piezas
defectuosas el día: ", j
j==15
p15=piezas
"La máquina: ", nmaq,
"Produjo: ", totmaq,
"piezas"
suma=suma+totmaq
x==1
minp15=p15
maqmin=nmaq
x=2
p15<minp15
minp15=p15
maqmin=nmaq
nmaq
prom=suma/31
"Promedio: ", prom
"Máquina menos producción: ",maqmin

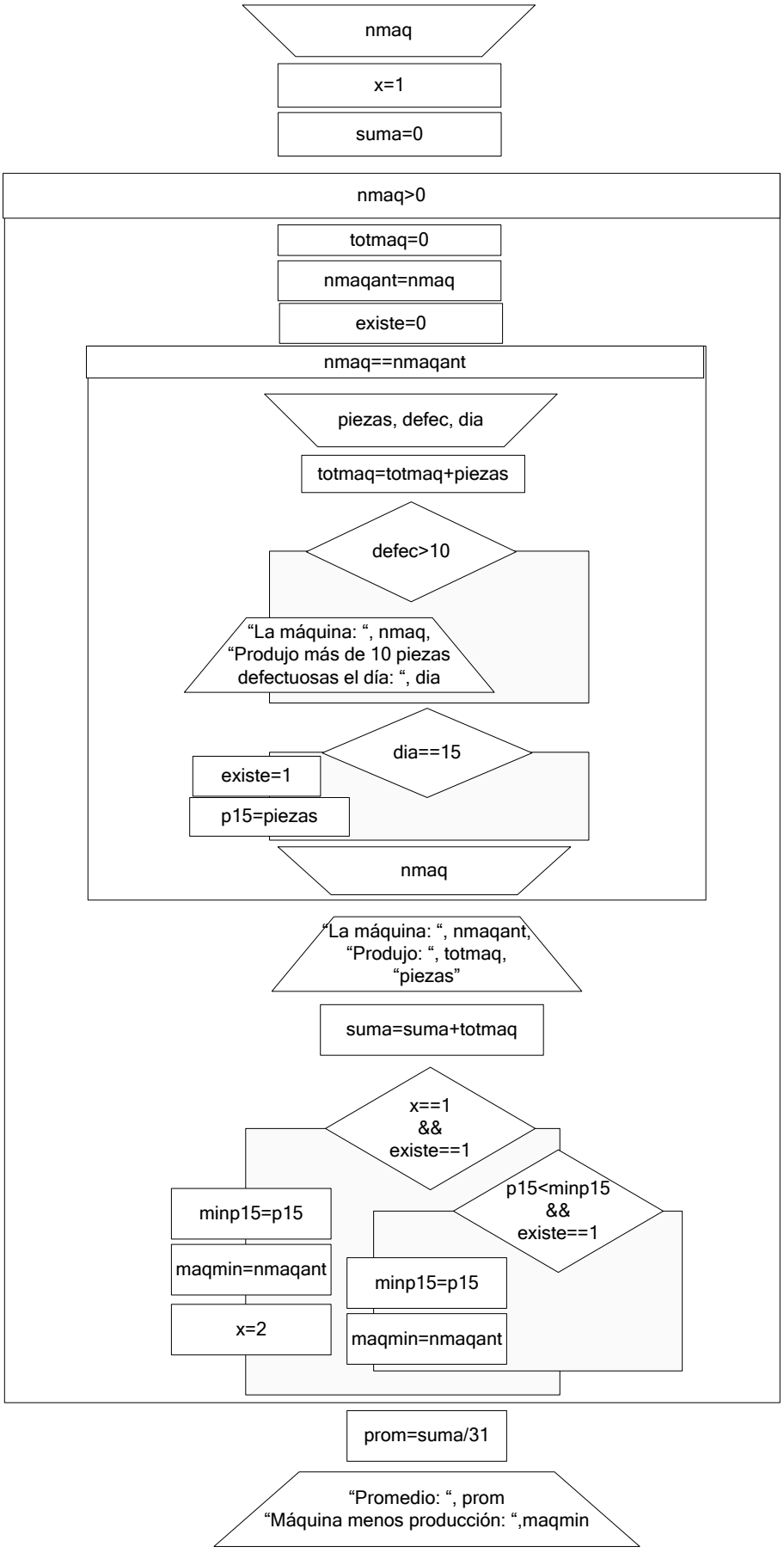
Codificación 1c)

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main()
{
int nmaq, piezas, defec;//para los datos de entrada
int suma, totmaq, p15, j, x;
int minp15, maqmin;
float prom;
cout<<"INGRESE EL NUMERO DE MAQUINA: ";
cin>>nmaq;
suma=0;
x=1;
while(nmaq>0){
  totmaq=0;
  for(j=1;j<=31;j++){
    cout<<"INGRESE LA CANTIDAD DE PIEZAS FABRICADAS: ";
    cin>>piezas;
    cout<<  "INGRESE LA CANTIDAD DE PIEZAS DEFECTUOSAS: ";
    cin>>defec;
    totmaq+=piezas;
    if(defec>10){
      cout<<"LA MAQUINA: "<<nmaq<<endl;
      cout<<"PRODUJO MAS DE 10 PIEZAS DEFECTUOSAS EL DIA: "<<j<<endl;
      }
    if(j==15){
      p15=piezas;
      }
    }
  cout<<"LA MAQUINA: "<<nmaq<<endl;
  cout<<"PRODUJO "<<totmaq<<" PIEZAS"<<endl;
  suma+=totmaq;
  if(x==1){
    minp15=p15;
    maqmin=nmaq;
    x=2;
    }
  else{
    if(p15<minp15){
      minp15=p15;
      maqmin=nmaq;
      }
    }
  }
prom=(float)suma/31;
```

```
cout<<"PROMEDIO: "<<prom<<endl;
cout<<"MAQUINA CON MENOR PRODUCCION EL DIA 15: "<<maqmin<<endl;
system("pause");
return 0;
}
```

1d)


nmaq
x=1
suma=0
nmaq>0
totmaq=0
nmaqant=nmaq
existe=0
nmaq==nmaqant
piezas, defec, dia
totmaq=totmaq+piezas
defec>10
"La máquina: ", nmaq,
"Produjo más de 10 piezas
defectuosas el día: ", dia
dia==15
existe=1
p15=piezas
nmaq
"La máquina: ", nmaqant,
"Produjo: ", totmaq,
"piezas"
suma=suma+totmaq
x==1
&&
existe==1
minp15=p15
maqmin=nmaqant
x=2
p15<minp15
&&
existe==1
minp15=p15
maqmin=nmaqant
prom=suma/31
"Promedio: ", prom
"Máquina menos producción: ",maqmin

Codificación 1d)

```
#include <iostream>
#include <cstdlib>
using namespace std;

int main()
{
int nmaq, piezas, defec, dia;//para los datos de entrada
int suma=0, totmaq, p15, existe, x=1;
int minp15, maqmin, nmaqant;
float prom;
cout<<"INGRESE EL NUMERO DE MAQUINA: ";
cin>>nmaq;
while(nmaq>0){
  totmaq=0;
  nmaqant=nmaq;
  existe=0;
  while(nmaq==nmaqant){
    cout<<"INGRESE EL DIA: ";
    cin>>dia;
    cout<<"INGRESE LA CANTIDAD DE PIEZAS FABRICADAS: ";
    cin>>piezas;
    cout<<  "INGRESE LA CANTIDAD DE PIEZAS DEFECTUOSAS: ";
    cin>>defec;
    totmaq+=piezas;
    if(defec>10){
      cout<<"LA MAQUINA: "<<nmaq<<endl;
      cout<<"PRODUJO MAS DE 10 PIEZAS DEFECTUOSAS EL DIA: "<<dia<<endl;
      }
    if(dia==15){
      existe=1;
      p15=piezas;
      }
    cout<<"INGRESE EL NUMERO DE MAQUINA: ";
    cin>>nmaq;
    }
  cout<<"LA MAQUINA: "<<nmaqant<<endl;
  cout<<"PRODUJO "<<totmaq<<" PIEZAS"<<endl;
  suma+=totmaq;
  if(x==1 && existe==1){
    minp15=p15;
    maqmin=nmaqant;
    x=2;
    }
  else    if(p15<minp15 && existe==1){
      minp15=p15;
      maqmin=nmaqant;
      }
```

```
  }
prom=(float)suma/31;
cout<<"PROMEDIO: "<<prom<<endl;
cout<<"MAQUINA CON MENOR PRODUCCION EL DIA 15: "<<maqmin<<endl;
system("pause");
return 0;
}
```

Corte de control

El último de los ejercicios es un ejemplo de la técnica denominada corte de control.

TRABAJO PRACTICO Nº 5. EJERCICIOS RESUELTOS

3)
Datos de entrada: conjunto indeterminado de 3 valores que deben analizarse de manera conjunta (registros de alumnos). Cada registro incluye un código de universidad, un número de legajo, y un puntaje. Todos los valores que se ingresan son números.

Los registros se encuentran agrupados por código de universidad, es decir todos los registros pertenecientes a una universidad estarán juntos (uno a continuación del otro). No se sabe cuántas universidades son las que se analizan.

Para indicar el fin del ingreso de datos se ingresa un código de universidad igual a cero.

Datos de salida:
Por cada universidad se debe informar el código de universidad, la cantidad de alumnos y el promedio de notas.

Resolución de los puntos:

a) Por cada conjunto de registros de alumnos ingresados pertenecientes a una misma universidad se debe informar el promedio, por lo que se necesita contar los alumnos y sumar las notas de esos alumnos. Para eso se utilizan las variables cantalumnos (contador de alumnos) y sumanotas (acumulador para los puntajes). Cada vez que se empieza a evaluar una universidad estas dos variables deben ponerse en 0.

   Cuando se detecta el cambio de universidad (cambia el código de universidad), se calcula e informa el promedio.

Estructura general del programa:
Como los datos de entrada son un conjunto indeterminado de registros, y sobre cada uno de ellos deben aplicarse las mismas instrucciones, deberá utilizarse un ciclo inexacto.

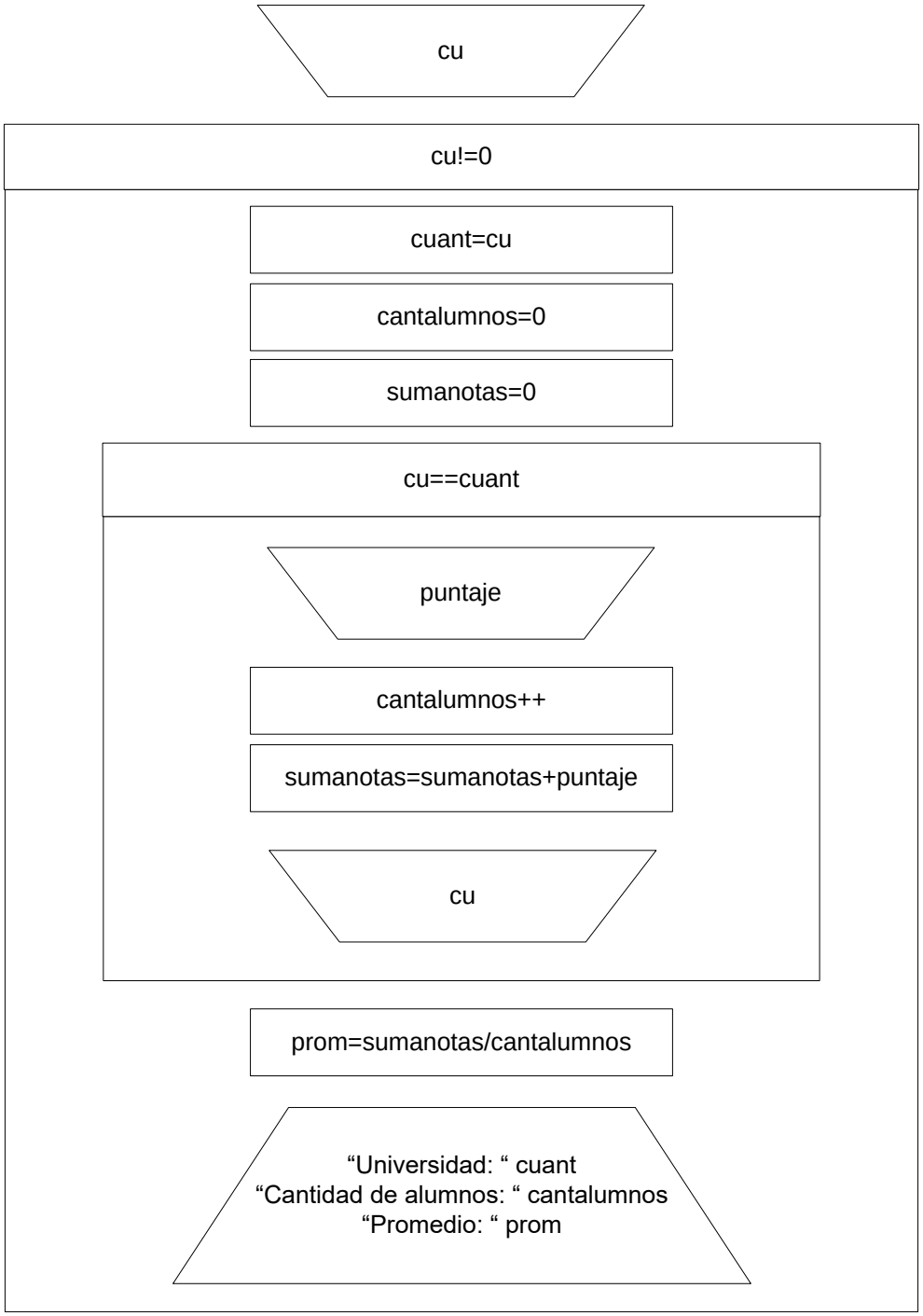
Como el fin del ingreso de los datos se indica con un código de universidad igual a cero, la condición de funcionamiento del ciclo será la negación de ese valor (cu!=0).

Lo anterior permite que el programa termine cuando corresponde, pero no alcanza para resolver el problema planteado, ya que la información a procesar está separada en subconjuntos o lotes (registros pertenecientes a una misma universidad).

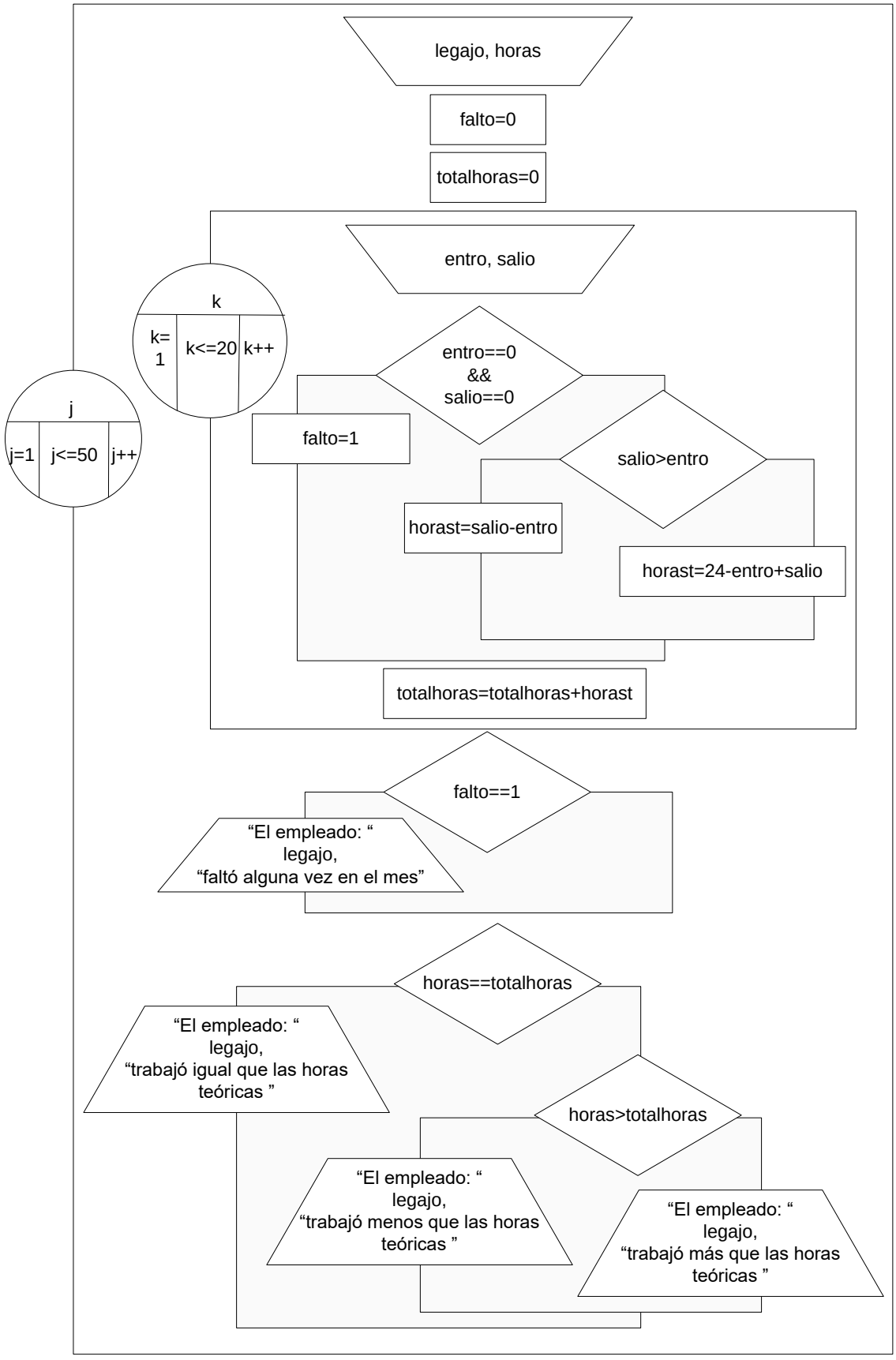
Para los subconjuntos no existe tampoco la información que indique cuántos registros tienen, por lo que deberá utilizarse otro ciclo inexacto. Este ciclo interno deberá cortar cuando detecte el cambio de universidad. Esto se consigue de la siguiente manera:
- Se ingresa el primer código de universidad al inicio del programa.
- Se analiza si es distinto de 0, y si es esto es verdadero se ingresa al while externo.

- Dentro del while externo se asigna a una variable (cuant) el valor del código de universidad, y se ponen en cero las variables para el cálculo del promedio.
- Luego se analiza la condición del segundo ciclo (para el primer ingreso siempre va a ser verdadero), se cuenta el alumno y se acumulan los puntajes, y se pide el ingreso del código de universidad del próximo registro. Como esto se hace al final del ciclo interno, se vuelve a analizar la condición de éste (cu==cuant) .
- En este punto la ejecución del programa continuará dentro del ciclo interno, mientras que no cambie el código de universidad.
- Cuando se detecta el cambio de universidad se corta el while interno, y se calcula e informa el promedio. Notesé que se informa cuant, ya que cu contiene el código de la próxima universidad.
- Luego se analiza la condición del while externo. Si es distinto de cero, se vuelve a asignar a cuant el valor de cu, se ponen en 0 las variables, y se comienza a analizar otra universidad.
- El programa se termina cuando se ingresa un cu con valor 0.

Nota: si bien el enunciado dice que los registros contienen el legajo de cada alumno, este no se ingresa ya que no es necesario para resolver el ejercicio.


cu
cu!=0
cuant=cu
cantalumnos=0
sumanotas=0
cu==cuant
puntaje
cantalumnos++
sumanotas=sumanotas+puntaje
cu
prom=sumanotas/cantalumnos
“Universidad: “ cuant
“Cantidad de alumnos: “ cantalumnos
“Promedio: “ prom

6b)


legajo, horas
falto=0
totalhoras=0
entro, salio
k
k=
1
k<=20
k++
j
j=1
j<=50
j++
entro==0
&&
salio==0
falto=1
salio>entro
horast=salio-entro
horast=24-entro+salio
totalhoras=totalhoras+horast
falto==1
“El empleado: “
legajo,
“faltó alguna vez en el mes”
horas==totalhoras
“El empleado: “
legajo,
“trabajó igual que las horas
teóricas ”
horas>totalhoras
“El empleado: “
legajo,
“trabajó menos que las horas
teóricas ”
“El empleado: “
legajo,
“trabajó más que las horas
teóricas ”