

Guía 3 – Estructuras iterativas.

Alumna: Ruiz, Abril Valentina.

Ejercicio 1) Lotería.

Un apostador necesita procesar los 20 premios de la lotería, para luego conocer el promedio de todos los impares, el mayor, menor y la cantidad de los números pares que han salido sorteados.

Desarrolle una aplicación que permita ingresar los 20 números y presente los resultados.

Análisis: Datos: premio

Incógnita: promedio de premios impares, cantidad de premios pares, ¿Cuál es el mayor y el menor?

Relaciones: Si premio > mayor Entonces
 Si premio < menor Entonces
 Si premio MOD 2 <> 0 Entonces
 Sino Si premio <> 0 Entonces
 Si cantIm > 0 Entonces
 promIm <- sumIm / cantIm

Estrategia: 1) Solicitar al usuario los 20 premios de la lotería.

2) Para i <- 1 Hasta 20 Hacer

A) Informar el número de premio a ingresar

B) Leer el premio

C) Si premio > mayor Entonces hacer

-Asignarle a mayor el premio

D) Si premio < menor Entonces hacer

-Asignarle a menor el premio

E) Si premio MOD 2 != 0 Entonces hacer

-Acumular el valor del premio

-Incrementar el contador de premios impares

-Sino, Incrementar el contador de premios pares

G) Si cantIm > 0 Entonces hacer

-Calcular el promedio de números impares

-Informar el promedio de números impares

H) Informar al usuario el premio mayor, el premio menor y la cantidad de números pares.

Ambiente:

Variable	Tipo de Dato	Descripción
premio	Entero	Premios de la lotería
sumIm	Entero	Acumulador de premios impares
cantIm	Entero	Contador de premios impares
cantPar	Entero	Contador de premios pares
mayor	Entero	Premio mayor
menor	Entero	Premio menor
promIm	Real	Promedio de premios impares

Seudocódigo:

Proceso guia3_1

Definir premio, sumIm, cantIm, cantPar, mayor, menor, i Como Entero;

Definir promIm Como Real;

mayor <- 0;

menor <- 10000;

sumIm <- 0;

```

cantlm <- 0;
cantPar <- 0;
Escribir "Ingrese los 20 premios de la lotería";
Para i <- 1 Hasta 20 Hacer
  Escribir "Premio ", i;
  Leer premio;

```

```

Si premio > mayor Entonces
  mayor <- premio;
FinSi

```

```

Si premio < menor Entonces
  menor <- premio;
FinSi

```

```

Si premio MOD 2 != 0 Entonces
  sumalm <- sumalm + premio;
  cantlm <- cantlm + 1;
Sino
  cantPar <- cantPar + 1;
FinSi
FinSi
FinPara

```

```

Si cantlm > 0 Entonces
  promlm <- sumalm / cantlm;
  Escribir "El promedio de premios impares es ", promlm;
Sino
  Escribir "No hubo premios impares";
FinSi

```

```

Escribir "El premio mayor es ", mayor;
Escribir "El premio menor es ", menor;
Escribir "La cantidad de premios pares es ", cantPar;
FinProceso

```

Seguimiento:

N°	premio	mayor	menor	sumalm	cantlm	cantPar	promlm	Salida
1	-	0	10000	0	0	0	-	Ingrese los 20 premios de la lotería
2	-	0	10000	0	0	0	-	Premio 1
3	3539	3539	3539	3539	1	0	-	-
4	3539	3539	3539	3539	1	0	-	Premio2
5	1463	3539	1463	5002	2	0	-	-
6	1463	3539	1463	5002	2	0	-	Premio 3
7	1471	3539	1463	6473	3	0	-	-
8	1471	3539	1463	6473	3	0	-	Premio 4

9	3936	3936	1463	6473	3	1	-	-
10	3936	3936	1463	6473	3	1	-	Premio 5
11	3154	3936	1463	6473	3	2	-	-
12	3154	3936	1463	6473	3	2	-	Premio 6
13	8740	8740	1463	6473	3	3	-	-
14	8740	8740	1463	6473	3	3	-	Premio 7
15	3730	8740	1463	6473	3	4	-	-
16	3730	8740	1463	6473	3	4	-	Premio 8
17	2342	8740	1463	6473	3	5	-	-
18	2342	8740	1463	6473	3	5	-	Premio 9
19	5528	8740	1463	6473	3	6	-	-
20	5528	8740	1463	6473	3	6	-	Premio 10
21	9731	9731	1463	16204	4	6	-	-
22	9731	9731	1463	16204	4	6	-	Premio 11
23	3497	9731	1463	19701	5	6	-	-
24	3497	9731	1463	19701	5	6	-	Premio 12
25	6720	9731	1463	19701	5	7	-	-
26	6720	9731	1463	19701	5	7	-	Premio13
27	4044	9731	1463	19701	5	8	-	-
28	4044	9731	1463	19701	5	8	-	Premio 14
29	9985	9985	1463	29686	6	8	-	-
30	9985	9985	1463	29686	6	8	-	Premio 15
31	9266	9985	1463	29686	6	9	-	-
32	9266	9985	1463	29686	6	9	-	Premio 16
33	6642	9985	1463	29686	6	10	-	-
34	6642	9985	1463	29686	6	10	-	Premio 17
35	0579	9985	0579	30265	7	10	-	-
36	0579	9985	0579	30265	7	10	-	Premio 18
37	3446	9985	0579	30265	7	11	-	-
38	3446	9985	0579	30265	7	11	-	Premio 19
39	2744	9985	0579	30265	7	12	-	-
40	2744	9985	0579	30265	7	12	-	Premio 20
41	7476	9985	0579	30265	7	13	-	-
42	7476	9985	0579	30265	7	13	4323.5	El promedio de impares es 4323.5
43	7476	9985	0579	30265	7	13	4323.5	El premio mayor es 9985
44	7476	9985	0579	30265	7	13	4323.5	El premio menor es 0579
45	7476	9985	0579	30265	7	13	4323.5	La cantidad de números pares es 13

Ejercicio 2) Disc Jockey

Un Disc Jockey requiere de un proceso que

- le permita ingresar una lista de canciones con Nombre y duración en MM:SS, informando al final el tiempo total en HH:MM: SS y La canción con mayor y menor duración
- Modifique el ejercicio anterior para que corte en forma automática la carga en el caso que supere los 74 Minutos

Análisis: Datos: dCan, nCan

Incógnitas: Tiempo total en hh:mm:ss, ¿Cuál es la canción con mayor y menor duración?

Relaciones: $dCan <- (mm * 60) + ss$;
Si $dCan > may$ Entonces
Si $dCan < men$ Entonces
 $totS <- totS + dCan$;

$hh <- totS / 3600$;

$mm <- totS / 60$;

$ss <- totS \bmod 60$;

Estrategia: 1) Mientras el total de segundos sea menor a 74 minutos, iterar:

- Solicitar que se ingrese el nombre de una canción
- Solicitar la duración de la canción en MM: SS
- Convertir la duración a segundos
- Si la duración es mayor, entonces asignarle el valor a “mayor”
- Si la duración es menor, entonces asignarle el valor a “menor”
- Acumular la duración.

2) Convertir el total de la duración a HH:MM: SS

3) Informar el tiempo total de reproducción

4) Informar la canción con mayor duración

5) Informar la canción con menor duración.

Ambiente:

Variable	Tipo de dato	Descripción
totS	Entero	Acumulación de todas las canciones en ss
may	Entero	Tiempo mayor de canción
men	Entero	Tiempo menor de canción
dCan	Entero	Duración de la canción
nCan	Cadena	Nombre de la canción
nomMay	Cadena	Nombre de la canción con mayor duración
nomMen	Cadena	Nombre de la canción con menor duración
hh	Entero	Horas
mm	Entero	Minutos
ss	Entero	Segundos

Seudocódigo:

Proceso ej2

Definir totS, may, men, dCan, hh, mm, ss Como Entero;

Definir nCan, nomMay, nomMen como cadena;

totS <- 0;

may <- 0;

men <- 9999;

Mientras totS <= 4440 Hacer

Escribir "ingrese el nombre de la canción";

Leer nCan;

Escribir "ingrese la duración de la canción en MM: SS";

Leer MM, SS;

dCan <- (mm * 60) + ss;

Si dCan > may Entonces

may <- dCan;

nomMay <- nCan;

FinSi

Si dCan < men Entonces

men <- dCan;

nomMen <- nCan;

FinSi

totS <- totS+dCan;

FinMientras

hh <- totS / 3600;

mm <- totS / 60;

ss <- totS mod 60;

Escribir "Tiempo total: ",hh,"horas, ", mm," minutos y ",ss," segundos";

Escribir "Canción con mayor duración ",nomMay;

Escribir "Canción con menor duración ",nomMen;

FinProceso

Seguimiento:

N	nCan	hh	mm	ss	dCan	may	men	nom May	nom Men	totS	Salida
1	-	-	-	-	-	0	9999	-	-	0	Ingrese el nombre y la duración
2	Ab	-	3	23	-	0	9999	-	-	0	-
3	Ab	-	3	23	203	203	203	Ab	Ab	203	Ingrese ""
4	Bc	-	4	46	286	286	203	Bc	Ab	489	Ingrese ""
5	Cd	-	2	56	176	286	176	Bc	Cd	665	Ingrese " "
6	De	-	6	57	417	417	176	De	Cd	1082	Ingrese" "
7	Ef	-	5	59	359	417	176	De	Cd	1441	Ingrese" "
8	Fg	-	8	40	520	520	176	Fg	Cd	1961	Ingrese ""
9	Gh	-	3	30	210	520	176	Fg	Cd	2171	Ingrese ""
10	Hi	-	4	50	290	520	176	Fg	Cd	2461	Ingrese ""
11	ij	-	5	60	360	520	176	Fg	Cd	2821	Ingrese ""
12	JK	-	4	50	290	520	176	Fg	Cd	3111	Ingrese ""
13	Kl	-	4	40	280	520	176	Fg	Cd	3391	Ingrese ""
14	Lm	-	5	55	355	520	176	Fg	Cd	3746	Ingrese ""

15	Mn	-	6	48	408	520	176	Fg	Cd	4154	Ingrese ""
16	No	-	3	45	225	520	176	Fg	Cd	4379	
17	No	1	59	5	225	520	176	Fg	Cd	4379	Tiempo total: 1h, 59 m, 5s
18	No	1	59	5	225	520	176	Fg	Cd	4379	Canción mayor: Fg
19	No	1	59	5	225	520	176	Fg	Cd	4379	Canción menor: Cd

Ejercicio 3) Tornillos

Una fábrica de tornillos realiza el control de calidad de su producción evaluando 10 productos de cada lote.

Al iniciar cada lote se ingresa el número de código, la medida esperada y la medición de los 10 elementos tomados al azar. Al finalizar la carga debe informar el mayor error absoluto y el porcentaje de productos con fallas. Al terminar de procesar todos los lotes (ingresando el número de código 0) debe informar:

- Cantidad de lotes procesados
- % total de fallas
- Lote con menor cantidad de fallas
- Lote con mayor cantidad de fallas.

Análisis: Datos: código, medEsp, medición

Incógnita: Cantidad de lotes procesados, porcentaje total de fallas, lote con menor cantidad de fallas y lote con mayor cantidad de fallas.

Relaciones: Si medición \geq medEsp Entonces

Si error $>$ mayorError Entonces

Si error $>$ 0 Entonces

Si totProd $>$ 0 Entonces

Si totProd $<$ menorFallas Entonces

Si totProd $>$ mayorFallas Entonces

porcProd \leftarrow (totProd * 100) / 10;

porcentajeFallas \leftarrow (lotesFallas*100) / totLotes;

Estrategia: 1) Solicitar el código del lote

2) Mientras el código sea distinto de 0, iterar:

- Acumular la cantidad de lotes
- Solicitar la medida esperada
- Para el lote hasta 10 iterar:
 - Solicitar que se ingresen la medición de un elemento
- Calcular el error absoluto
- Verificar cual es el error mayor
- Si hay errores, acumular los productos con fallas

- Si el lote actual tiene productos con fallas entonces, acumular los lotes con fallas.
 - Si el total de productos con fallas es menor que el lote con menor cantidad de fallas, entonces actualizar el valor de la variable “menor cantidad de fallas”
 - Si el total de productos con fallas es mayor que el lote con mayor cantidad de fallas, entonces actualizar el valor de la variable “mayor cantidad de fallas”
 - Calcular el porcentaje de productos con fallas e informar el resultado por pantalla
 - Informar el mayor error absoluto del lote
 - Solicitar el código del siguiente lote
- 3) Calcular el porcentaje de lotes con fallas
- 4) Informar por pantalla: La cantidad de lotes procesados, el porcentaje total de fallas, y los lotes con mayor y menor cantidad de fallas.

Ambiente:

Variable	Tipo de dato	Descripción
cod	Entero	Código del lote
medEsp	Entero	Medida esperada
medición	Entero	Medición de los productos
mayorError	Entero	Mayor error absoluto
totProd	Entero	Total de productos con fallas
totLotes	Entero	Total de lotes procesados
lotesFallas	Entero	Total de lotes con fallas
mayorFallas	Entero	Lote con mayor cantidad de fallas
menorFallas	Entero	Lote con menor cantidad de fallas
error	Entero	Error absoluto
porcFallas	Real	Porcentaje de lotes con fallas
porcProd	Real	Porcentaje de productos con fallas

Seudocódigo:

Proceso ej3

Definir cod, medEsp, medición, mayorError, totProd, totLotes, lotesFallas, menorFallas, mayorFallas, error Como Entero;

Definir porcFallas, porcProd Como Real;

totLotes <- 0;

lotesFallas <- 0;

menorFallas <- 0;

mayorFallas <- 0;

Escribir "ingrese el código del lote (0 para finalizar)";

Leer cod;

Mientras cod <> 0 Hacer

totLotes <- totLotes + 1;

```
totProd<- 0;
mayorError<- 0;
Escribir "ingrese la medida esperada";
leer medEsp;
Para i<-1 Hasta 10 Hacer
  Escribir "Ingrese la medición del elemento";
  Leer medición;
  Si medición >= medEsp Entonces
    error <- medición - medEsp;
  SiNo
    error<- medEsp - medición;
  FinSi
  Si error > mayorError Entonces
    mayorError <- error;
  FinSi
  Si error > 0 Entonces
    totProd <- totProd + 1;
  FinSi
FinPara
Si totProd > 0 Entonces
  lotesFallas <- lotesFallas + 1;
FinSi
Si totProd < menorFallas Entonces
  menorFallas<- totProd;
FinSi
Si totProd > mayorFallas Entonces
  mayorFallas <- totProd;
FinSi
porcProd <- (totProd * 100) / 10;
Escribir "el mayor error absoluto en el lote es: ",mayorError;
Escribir "el porcentaje de productos con fallas es: ",porcProd;
Escribir "ingrese el código del siguiente lote (0 para finalizar)";
Leer cod;
FinMientras
porcFallas <- (lotesFallas*100) /totLotes;
Escribir "Cantidad de lotes procesados ",totLotes;
Escribir "Porcentaje total de fallas ",porcFallas;
Escribir "Lote con mayor cantidad de fallas ",mayorFallas;
Escribir "Lote con menor cantidad de fallas ",menorFallas;
FinProceso
```

Seguimiento:

N	cod	med Esp	med ión	may orError	tot Prod	totL otes	lotes Falla	meno rFalla	may or Fall as	erro r	porc Falla s	porc Prod	Salida
1	-	-	-	-	-	0	0	0	0	-	-	-	Ingrese el código del lote
2	100	-	-	0	0	1	0	0	0	-	-	-	Ingrese la medición esperada
3	100	15	-	0	0	1	0	0	0	-	-	-	Ingrese la medida del elemento
4	100	15	8	7	1	1	0	0	0	7	-	-	Ingrese la medida del elemento
5	100	15	15	7	1	1	0	0	0	0	-	-	Ingrese la medida del elemento
6	100	15	10	7	2	1	0	0	0	5	-	-	Ingrese la medida del elemento
7	100	15	11	7	3	1	0	0	0	4	-	-	Ingrese la medida del elemento
8	100	15	7	8	4	1	0	0	0	8	-	-	Ingrese la medida del elemento
9	100	15	15	8	4	1	0	0	0	0	-	-	Ingrese la medida del elemento
10	100	15	15	8	4	1	0	0	0	0	-	-	Ingrese la medida del elemento
11	100	15	15	8	4	1	0	0	0	0	-	-	Ingrese la medida del elemento
12	100	15	15	8	4	1	0	0	0	0	-	-	Ingrese la medida del elemento
13	100	15	15	8	4	1	1	4	4	0	-	40	Mayor error absoluto: 8
14	100	15	15	8	4	1	1	4	4	0	-	40	Porcentaje de productos con fallas: 40% Ingrese el código del sig. lote:
15	220	-	-	0	0	2	1	4	4	0	-	40	Ingrese la medición esperada:
16	220	20	-	0	0	2	1	-	-	0	-	-	Ingrese la medida del elemento
17	220	20	15	5	1	2	2	-	-	5	-	-	Ingrese la medida del elemento
18	220	20	9	11	2	2	2	-	-	11	-	-	Ingrese la medida del elemento
19	220	20	20	11	2	2	2	-	-	0	-	-	Ingrese la medida del elemento

20	220	20	20	11	2	2	2	-	-	0	-	-	Ingrese la medida del elemento
21	220	20	20	11	2	2	2	-	-	0	-	-	Ingrese la medida del elemento
22	220	20	20	11	2	2	2	-	-	0	-	-	Ingrese la medida del elemento
23	220	20	20	11	2	2	2	-	-	0	-	-	Ingrese la medida del elemento
24	220	20	20	11	2	2	2	-	-	0	-	-	Ingrese la medida del elemento
25	220	20	20	11	2	2	2	-	-	0	-	-	Ingrese la medida del elemento
26	220	20	20	11	2	2	2	2	4	0	-	20	El mayor error absoluto es: 11
27	220	20	20	11	2	2	2	2	4	0	-	20	Porcentaje de productos con fallas: 20% Ingrese el código del siguiente lote
28	0	-	-	-	-	-	-	-	-	-	100	-	
29	0	-	-	-	-	-	-	-	-	-	100	-	Cantidad de lotes procesados: 2 Porcentaje total de fallas: 100% Lote con mayor cantidad de fallas: 4 Lote con menor cantidad de fallas: 2

Ejercicio 4)

Una forma de determinar si un número es primo consiste en verificar si es divisible por sí mismo y por uno.

Implemente una aplicación que determine si un valor entero ingresado por el usuario es o no un número primo.

Análisis: Datos: num

Incógnitas: ¿Es un número primo?

Relaciones: Si $\text{num} \bmod \text{divisor} = 0$ Entonces
Si $\text{cont} = 2$ Entonces

Estrategia: 1) Solicitar y leer un numero entero

2) Inicializar un contador

3) Iterar de 1 hasta 10 lo siguiente:

-Si el número es divisible por divisor Entonces, aumentar en 1 el contador

4) Si el contador es igual a dos Entonces:

-Informar que es un numero primo.

-Sino, informar que no es un numero primo.

Ambiente:

Variable	Tipo de dato	Descripción
num	Entero	Número ingresado por el usuario
divisor	Entero	Números divisores del número ingresado
cont	Entero	Contador de divisores

Seudocódigo:

```

Proceso ej4
Definir num, divisor, cont Como Entero;
Escribir "Ingrese un número entero";
Leer num;
cont <- 0;
Para divisor = 1 hasta num Hacer
Si num mod divisor = 0 Entonces
cont <- cont + 1;
Fin Si
Fin Para
Si cont = 2 Entonces
Escribir "El número ", num, " es primo";
Sino
Escribir "El número ", num, " no es primo";
Fin Si
FinProceso

```

Seguimiento:

N°	num	divisor	cont	Salida
1	-	-	-	Ingrese un numero entero
2	11	-	0	-
3	11	1	1	-
4	11	2	2	-
5	11	2	2	El número es primo

Ejercicio 5)

Considerando el ejercicio anterior realice un proceso que muestre en pantalla todos los números primos que pertenezcan a un rango ingresado por el usuario. Considere cualquier orden de ingreso de los valores del rango.

Análisis: Datos: in, f

Incógnitas: Números primos según el rango ingresado

Relaciones: Si in > f entonces

Si num mod div = 0 Entonces

Si cont = 2 Entonces

Estrategia: 1) Solicitar y leer el primer número del rango

2) Solicitar y leer el segundo número del rango

3) Si el segundo número es mayor que el primero intercambiar los valores

4) Iterar a través de todos los números dentro del rango

- Inicializar el contador de divisores

- Si el número es divisible por divisor Entonces, aumentar en 1 el contador

- Si el contador es igual a dos Entonces, informar por pantalla los números primos.

Ambiente:

Variable	Tipo de dato	Descripción
in	Entero	Inicio del rango
f	Entero	Final del rango
num	Entero	Número que se dividirá
div	Entero	Divisor del numero
cont	Entero	Contador de divisores
ord	Entero	Orden del rango

Seudocódigo:

```

Proceso ej5
Definir in, f, num, div, cont Como Entero;
Escribir "Ingrese el primer número del rango";
Leer in;
Escribir "Ingrese el segundo número del rango";
Leer f;
Si in > f Entonces
    Definir ord Como Entero;
    ord <- in;
    in <- f;
    f <- ord;
FinSi
Para num = in hasta f Hacer
    cont <- 0;
    Para div = 1 hasta num Hacer
        Si num mod div = 0 Entonces
            cont <- cont + 1;
        FinSi
    FinPara
    Si cont = 2 Entonces
        Escribir num, " es primo";
    FinSi
FinPara
FinProceso
  
```

Seguimiento:

N	In	f	ord	num	div	mod	cont	Salida
1	-	-	-	-	-	-	-	Ingrese el primer y el ultimo número del rango
2	1	9	-	-	-	-	-	-
3	1	9	-	1	-	-	-	-
4	1	9	-	1	1	V	1	1 es primo

5	1	9	-	2	1	V	2	-
6	1	9	-	2	2	V	3	-
7	1	9	-	3	1	V	2	-
8	1	9	-	3	2	F	2	3 es primo
9	1	9	-	4	1	V	3	-
10	1	9	-	4	2	V	4	-
11	1	9	-	5	1	V	2	-
12	1	9	-	5	2	F	2	-
13	1	9	-	6	1	V	3	-
14	1	9	-	6	2	V	4	-
15	1	9	-	6	3	V	5	-
16	1	9	-	7	1	V	2	-
17	1	9	-	7	2	F	-	7 es primo
18	1	9	-	8	1	V	3	-
19	1	9	-	8	2	V	4	-
20	1	9	-	9	1	V	2	-
21	1	9	-	9	2	F	-	9 es primo

Ejercicio 6)

Se ingresa una lista de notas correspondientes a una evaluación de programación numeradas entre 0 y 10. Al finalizar se debe mostrar en pantalla:

- Cantidad de notas
- Promedio
- Cantidad de aprobados y no aprobados
- Porcentaje de alumnos con:
 - o Muy Bueno (8 o más)
 - o Bueno (6 o 7)
 - o Regular (4 o 5)
 - o Insuficiente (3 o menos)

Análisis: Datos: nota

Incógnitas: cantidad de Notas, promedio, cantidad de aprobados y no aprobados, porcentaje de alumnos con: muy bueno, bueno, regular e insuficiente.

Relaciones: `cantN <- cantN + 1;`

`sumaN <- sumaN + nota;`

Si `nota >= 4` Entonces

`aprob <- aprob + 1;`

Si `nota >= 8` Entonces

`muyB <- muyB + 1;`

Si `nota >= 6` Entonces

`bueno <- bueno + 1;`

`regular <- regular + 1;`

Si nota < 4 Entonces

```
noAprob <- noAprob + 1;
```

Si nota <= 3 Entonces

```
insuf <- insuf + 1;
```

Si cantN > 0 Entonces

```
prom <- sumaN / cantN;
```

```
porcMB <- (muyB / cantN) * 100;
```

```
porcB <- (bueno / cantN) * 100;
```

```
porcR <- (regular / cantN) * 100;
```

```
porcIn <- (insuf / cantN) * 100;
```

Estrategia: 1) Solicitar las notas de los exámenes

2) Mientras las notas sean mayores a 0, iterar:

- Aumentar en 1 la cantidad de notas ingresadas
- Acumular la suma de las notas
- Si la nota es >= a 4, aumentar en 1 el contador de aprobados
- Si la nota es >= a 8, aumentar en 1 el contador de muy buenos
- Si la nota es >= a 6, aumentar en 1 el contador de buenos. Sino aumentar en 1 el contador de regulares.
- Si la nota es < a 4, aumentar en 1 el contador de no aprobados
- Si la nota es <= a 3, aumentar en 1 el contador de insuficientes

3) Si la cantidad de notas es mayor a 0, iterar:

- Calcular el promedio de notas
- Calcular el porcentaje de muy buenos
- Calcular el porcentaje de buenos
- Calcular el porcentaje de reguares
- Calcular el porcentaje de insuficientes
- Informar por pantalla todos los resultados y la cantidad de notas.

Ambiente:

Variable	Tipo de dato	Descripción
cantN	Entero	Cantidad de notas

suman	Entero	Suma de las notas
aprob	Entero	Cantidad de alumnos aprobados
noAprob	Entero	Cantidad de alumnos no aprobados
muyB	Entero	Cantidad de Muy bueno (8, 9 o 10)
Bueno	Entero	Cantidad de Bueno (6 o 7)
regular	Entero	Cantidad de regular (4 o 5)
insuf	Entero	Cantidad de insuficiente (3, 2 o 1)
Nota	Real	Nota ingresada por el usuario
porcMB	Real	Porcentaje de muy bueno
porcB	Real	Porcentaje de bueno
porcR	Real	Porcentaje de regular
porcIn	Real	Porcentaje de insuficiente
Prom	Real	Promedio de notas

Seudocódigo:

Proceso ej6

Definir cantN, sumaN, aprob, noAprob, muyB, bueno, regular, insuf Como Entero;

Definir nota, porcMB, porcB, porcR, porcIn, prom Como Real;

cantN <- 0;

sumaN <- 0;

aprob <- 0;

noAprob <- 0;

muyB <- 0;

bueno <- 0;

regular <- 0;

insuf <- 0;

Escribir "Ingrese las notas (0 a 10). Ingrese un valor negativo para finalizar.";

Leer nota;

Mientras nota >= 0 Hacer

 cantN <- cantN + 1;

 sumaN <- sumaN + nota;

 Si nota >= 4 Entonces

 aprob <- aprob + 1;

 Si nota >= 8 Entonces

 muyB <- muyB + 1;

Sino

Si nota ≥ 6 Entonces

bueno \leftarrow bueno + 1;

Sino

regular \leftarrow regular + 1;

Fin Si

FinSi

FinSi

Si nota < 4 Entonces

noAprob \leftarrow noAprob + 1;

Si nota ≤ 3 Entonces

insuf \leftarrow insuf + 1;

FinSi

Fin Si

Leer nota;

Fin Mientras

Si cantN > 0 Entonces

prom \leftarrow sumaN / cantN;

porcMB \leftarrow (muyB / cantN) * 100;

porcB \leftarrow (bueno / cantN) * 100;

porcR \leftarrow (regular / cantN) * 100;

porcIn \leftarrow (insuf / cantN) * 100;

Escribir "Cantidad de notas: ", cantN;

Escribir "Promedio: ", prom;

Escribir "Aprobados: ", aprob;

Escribir "No aprobados: ", noAprob;

Escribir "Porcentaje de muy buenos: ", porcMB, "%";

Escribir "Porcentaje de buenos: ", porcB, "%";

Escribir "Porcentaje de regulares: ", porcR, "%";

Escribir "Porcentaje de insuficientes: ", porcIn, "%";

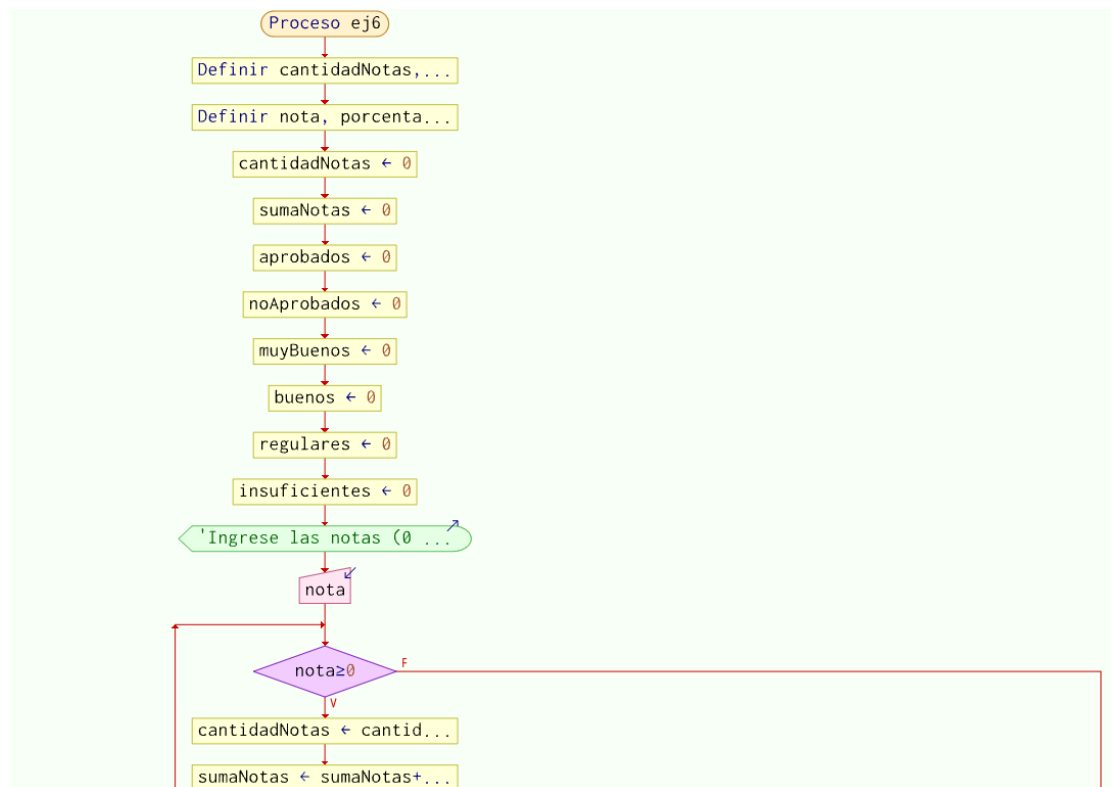
Fin Si

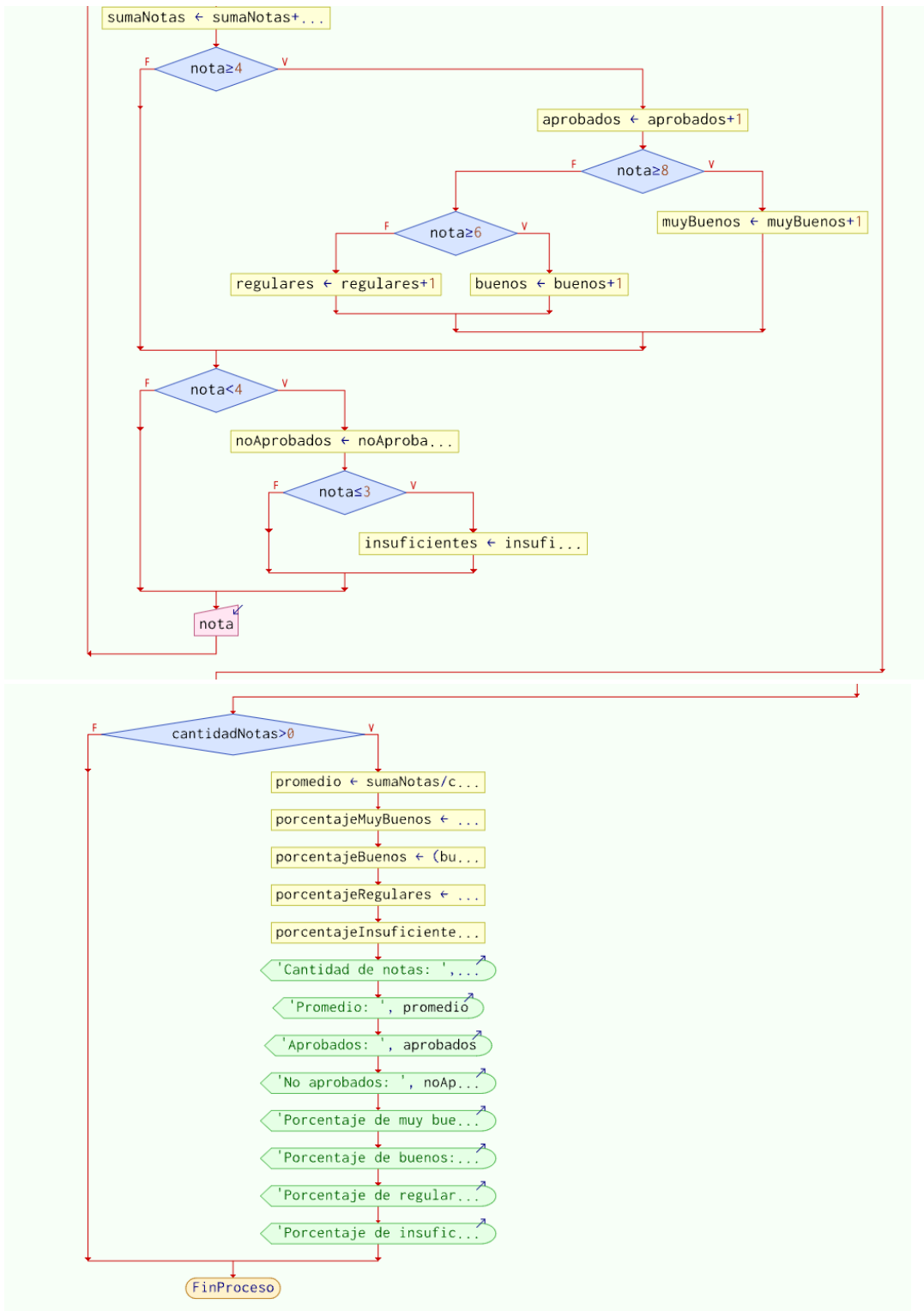
FinProceso

Seguimiento:

2	N	nota	cantN	sumaN	aprob	noAprob	muyB	bueno	regular	insuf	porcMB	porcB	porcR	porcIn	prom	Salida
3	1		0	0	0	0	0	0	0	0						Ingrese la nota de 0 a 10
4	2	10	1	10	1	0	1	0	0	0						
5	3	8	2	18	2	0	1	0	0	0						
6	4	7	3	25	3	0	1	1	0	0						
7	5	6	4	31	4	0	1	1	0	0						
8	6	3	5	34	4	1	1	1	0	1						
9	7	4	6	38	5	1	1	1	1	1						
10	8	3	7	41	5	2	1	1	1	2						
11	9	3	8	44	5	3	1	1	1	3						
12	10	10	9	54	6	3	2	1	1	3						
13	11	-1	9	54	6	3	2	1	1	3 33.3%	22.2%	11%	33.3%			6 La cantidad de notas es 9
14	12		9	54	6	3	2	1	1	3 33.3%	22.2%	11%	33.3%			6 El promedio es 6
15	13		9	54	6	3	2	1	1	3 33.3%	22.2%	11%	33.3%			6 Aprobados: 6. No aprobados: 3
16	14		9	54	6	3	2	1	1	3 33.3%	22.2%	11%	33.3%			6 Porcentaje de muy buenos: 33.3%
17	15		9	54	6	3	2	1	1	3 33.3%	22.2%	11%	33.3%			6 Porcentaje de buenos: 22.2%
18	16		9	54	6	3	2	1	1	3 33.3%	22.2%	11%	33.3%			6 Porcentaje de regulares: 11%
19	17		9	54	6	3	2	1	1	3 33.3%	22.2%	11%	33.3%			6 Porcentaje de insuficientes: 33.3%

Diagrama de Flujo:



**Ejercicio 7)**

Implemente una pequeña aplicación que permita calcular el total a pagar por una compra ingresando la cantidad y el precio unitario de cada producto. Debe informar además la cantidad de productos adquiridos.

Análisis: Datos: prod, cant, precioU

Incógnitas: total, prodAdq

Relaciones: Si cant > 0 Entonces

total <- total + (cant * precioU);

Estrategia: 1) Iterar hasta que cant <= 0:

- Solicitar el nombre del producto y la cantidad a comprar:

- Si cant > 0, entonces:

° Solicitar el precio unitario del producto

° Calcular el total a pagar

° Aumentar el contador de productos adquiridos en 1

2) Informar el total a pagar y la cantidad de productos adquiridos

Ambiente:

Variable	Tipo de dato	Descripción
Total	Real	Total a pagar
cant	Real	Cantidad de un mismo producto
precioU	Real	Precio unitario del producto
prodAdq	Entero	Contador de productos comprados/adquiridos
prod	Cadena	Nombre del producto

Seudocódigo:

Proceso ej7

Definir total, cant, precioU Como real;

Definir prodAdq Como entero;

Definir prod Como cadena;

total <- 0;

prodAdq <- 0;

Escribir "Ingrese la cantidad y el precio unitario de cada producto. Para finalizar ingrese 0";

Repetir

Escribir "Producto";

Leer prod;

Escribir "Cantidad:";

Leer cant;

Si cant > 0 Entonces

Escribir "Precio unitario:";

Leer precioU;

total <- total + (cant * precioU);

prodAdq <- prodAdq + 1;

Fin Si

Hasta Que cant <= 0

Escribir "Total a pagar:", total;

Escribir "Cantidad de productos adquiridos:", prodAdq;

FinProceso

Seguimiento:

N	Total	prodAdq	prod	Cant	precioU	Salida
---	-------	---------	------	------	---------	--------

1	0	0	-	-	-	Ingrese la cantidad y el precio unitario de cada producto. Para finalizar ingrese 0
2	0	0	-	-	-	Producto y cantidad:
3	1000	1	Leche	2	500	Producto y cantidad:
4	1400	2	Cocoa	1	400	Producto y cantidad:
5	2000	3	Azúcar	1	600	Producto y cantidad:
6	2000	3	-	0	-	
7	2000	3	-	-	-	Total a pagar: 2000
8	2000	3	-	-	-	Cantidad de productos adquiridos: 3