

# Programación I

## Guía 3 – Estructuras Iterativas

Martinez, Facundo

### Ejercicio 1)

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

20 Numero enteros

Salidas

Promedio de Impares

Mayor

Menor

Cantidad de números pares

Procesos

Si numero % 2 == 0 Entonces

Cantidad Pares = Cantidad Pares + 1

Sino

Suma Impares = Suma Impares + Numero

Cantidad Impares = Cantidad Impares + 1

Si numero > mayor Entonces

Mayor = numero

Sino

Si numero < menor Entonces

Menor = numero

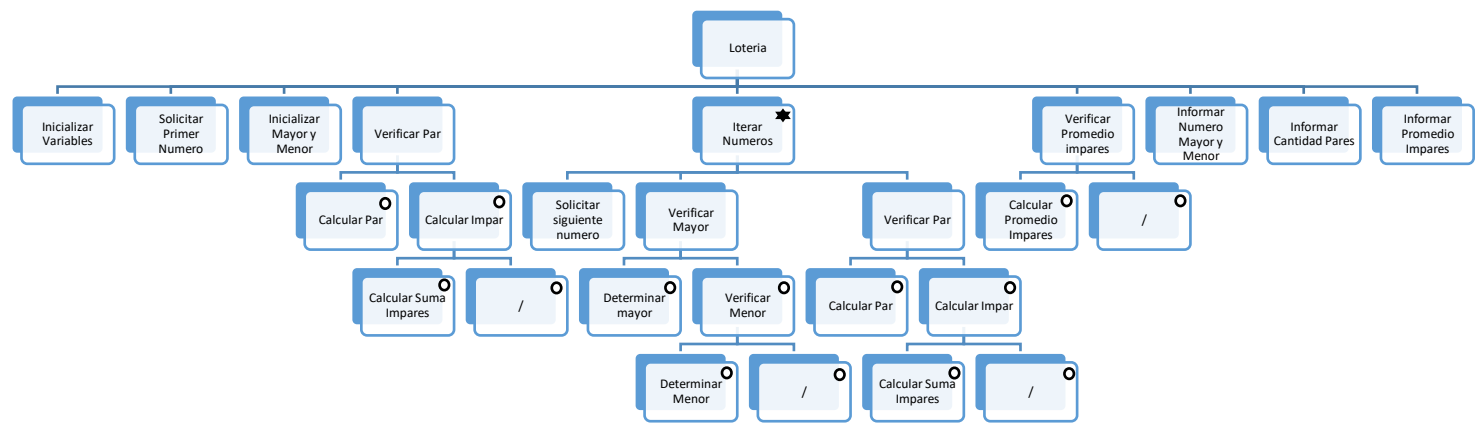
Si Cantidad Impares > 0 Entonces

Promedio Impares = Suma Impares / Cantidad Impares

Sino

Promedio Impares = 0

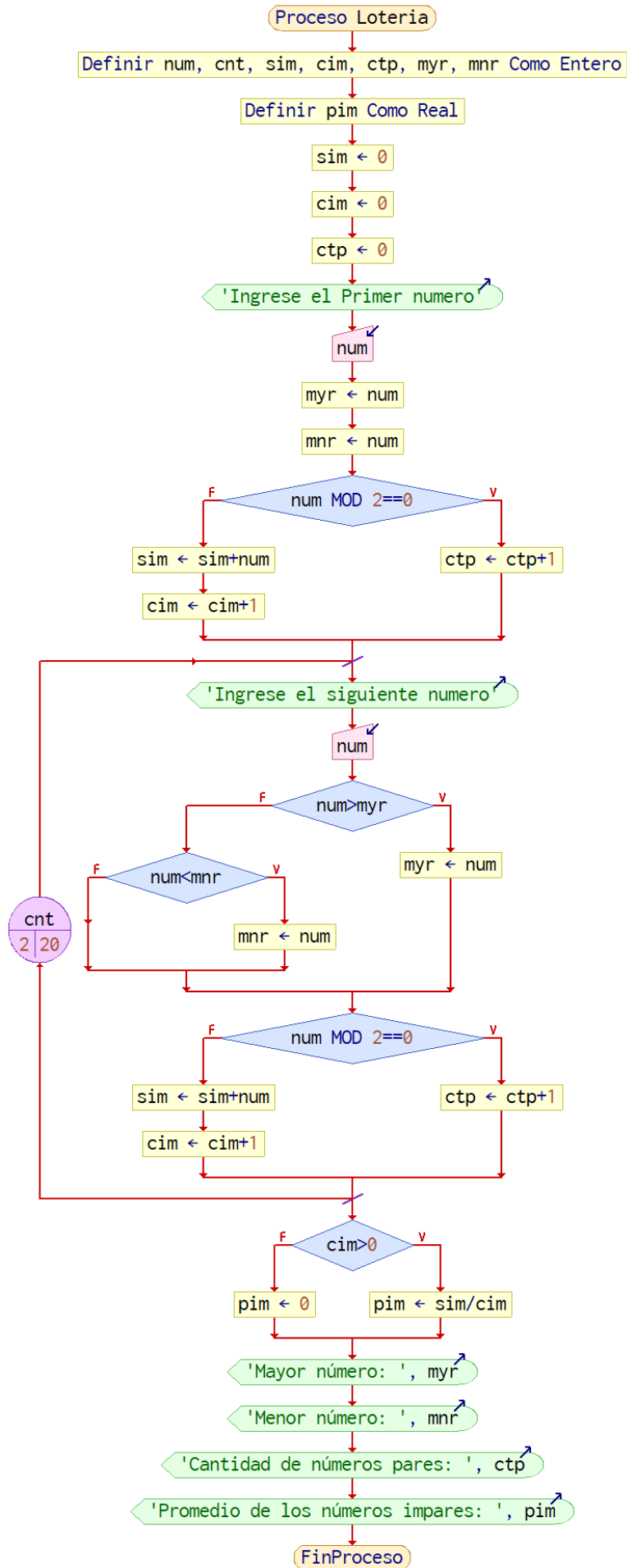
Estrategia



Ambiente

Variable	Tipo	Definición
num	Entero	Números a evaluar
cnt	Entero	Contador
sim	Entero	Suma Impares
cim	Entero	Cantidad Impares
ctp	Entero	Cantidad Pares
myr	Entero	Número Mayor
mnr	Entero	Número Menor
pim	Real	Promedio Impares

## Diagrama de flujo



## Seudocódigo

### Proceso Loteria

Definir num, cnt, sim, cim, ctp, myr, mnr Como Entero;

Definir pim Como Real;

sim <- 0;

cim <- 0;

ctp <- 0;

Escribir 'Ingrese el Primer numero';

Leer num;

myr <- num;

mnr <- num;

Si num MOD 2 == 0 Entonces

    ctp <- ctp + 1;

SiNo

    sim <- sim + num;

    cim <- cim + 1;

FinSi

Para cnt<-2 Hasta 20 Hacer

    Escribir 'Ingrese el siguiente numero';

    Leer num;

    Si num > myr Entonces

        myr <- num;

    SiNo

        Si num < mnr Entonces

            mnr <- num;

        FinSi

    FinSi

    Si num MOD 2 == 0 Entonces

        ctp <- ctp + 1;

    SiNo

        sim <- sim + num;

        cim <- cim+1;

    FinSi

FinPara

Si  $cim > 0$  Entonces

$pim \leftarrow sim / cim;$

SiNo

$pim \leftarrow 0;$

FinSi

Escribir 'Mayor número: ', myr;

Escribir 'Menor número: ', mnr;

Escribir 'Cantidad de números pares: ', ctp;

Escribir 'Promedio de los números impares: ', pim;

FinProceso

Prueba de escritorio

## Ejercicio 2)

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

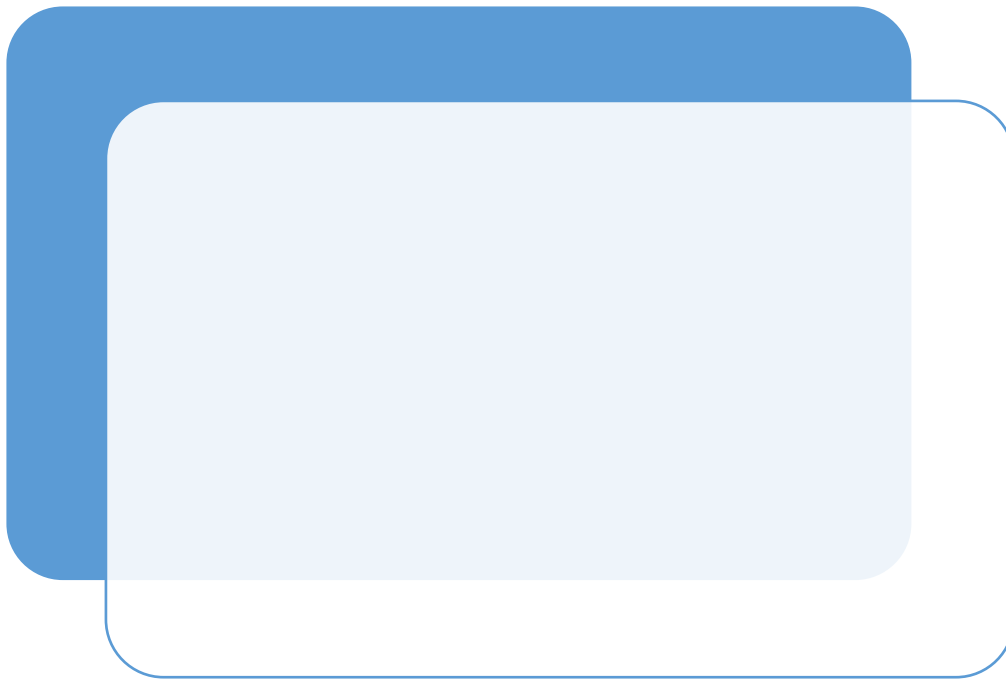
Análisis

Datos

Salidas

Procesos

Estrategia



Ambiente

Diagrama de flujo

Seudocódigo

Prueba de escritorio

### Ejercicio 3)

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.

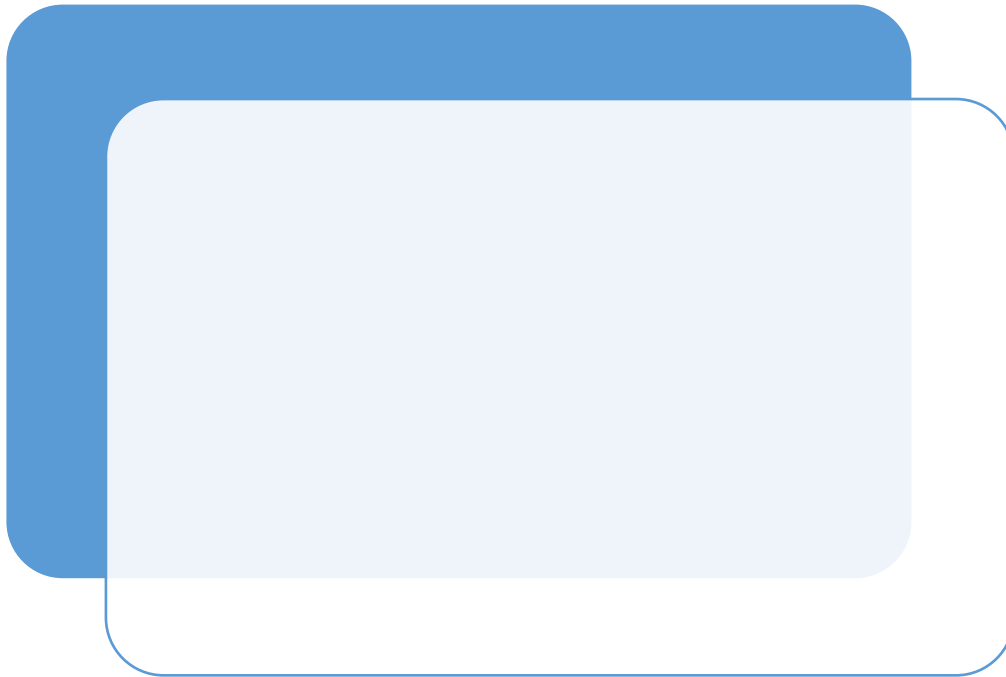
Análisis

Datos

Salidas

Procesos

Estrategia



Ambiente

Diagrama de flujo

Seudocódigo

Prueba de escritorio

#### 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

Numero Entero

Salidas

Determinar si es primo

Procesos

Variable numérica = Si numero entero % contador == 0 Entonces

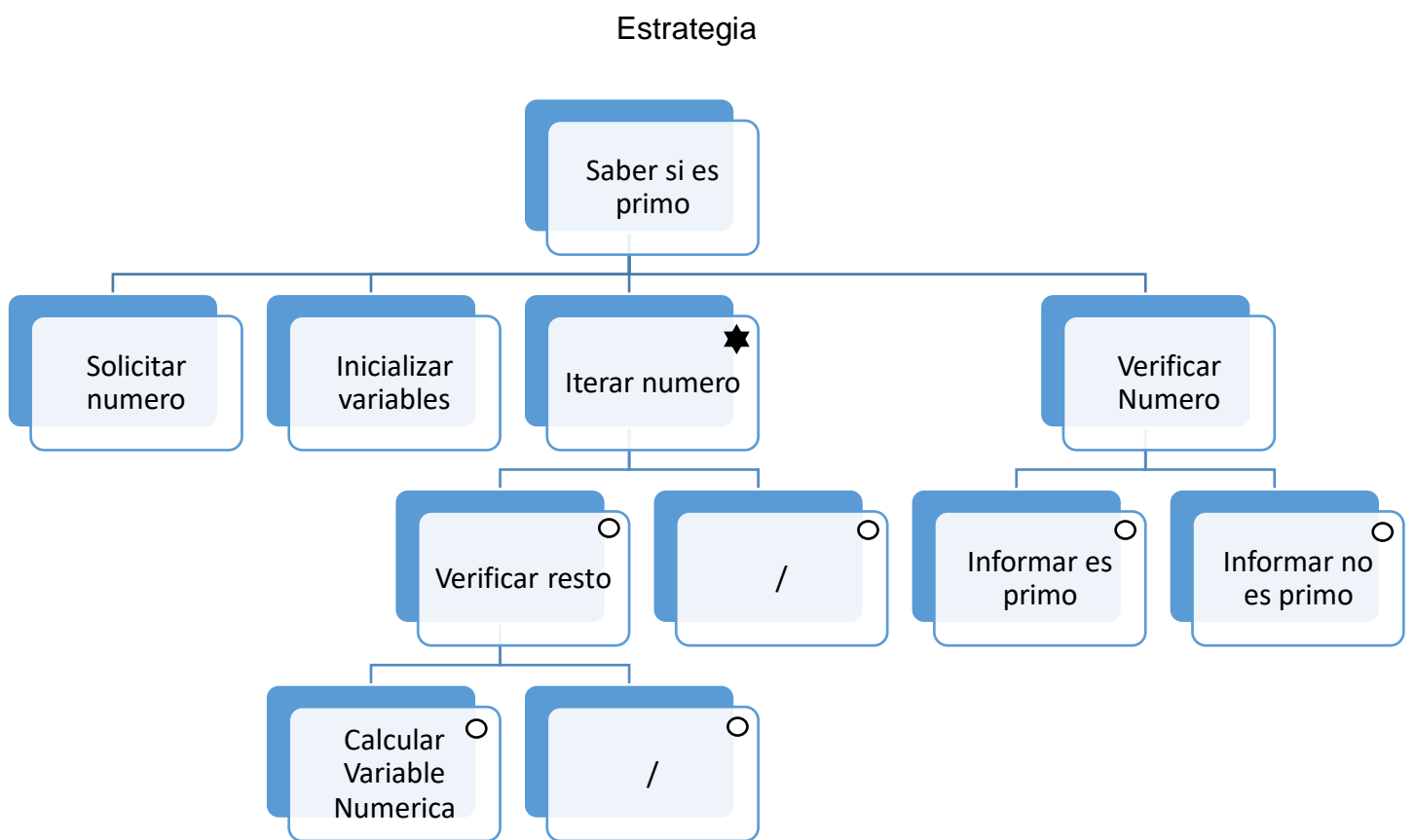
Variable numerica = variable numerica + 1;

Si variable numerica = 2 Entonces

    Escribir 'El número es primo.';

SiNo

    Escribir 'El número no es primo.';

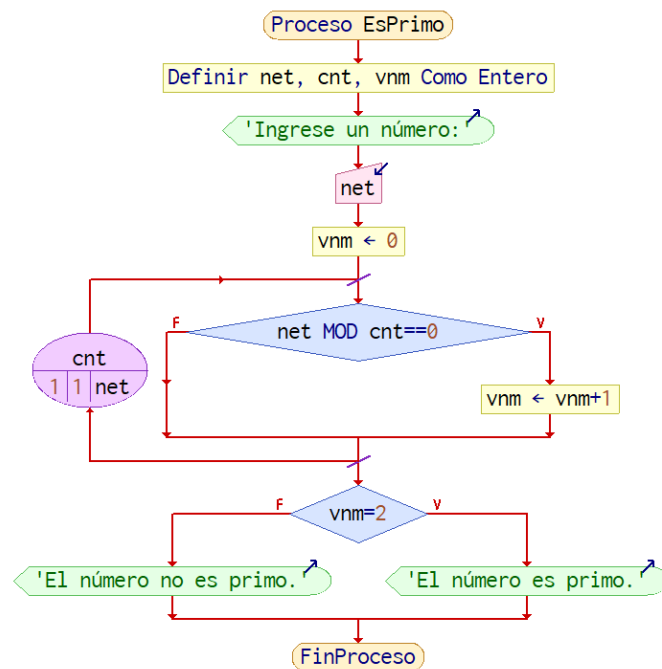


Ambiente



Variable	Tipo	Definición
net	Entero	Numero a verificar
cnt	Entero	Contador
vnv	Entero	Variable numérica

### Diagrama de flujo



### Seudocódigo

#### Proceso EsPrimo

Definir net, cnt, vnm Como Entero;

Escribir 'Ingrese un número:';

Leer net;

vnm <- 0;

Para cnt<-1 Hasta net Con Paso 1 Hacer

    Si net MOD cnt == 0 Entonces

        vnm <- vnm + 1;

    FinSi

FinPara

Si vnm = 2 Entonces

    Escribir 'El número es primo.';

SiNo

    Escribir 'El número no es primo.';

FinSi

Línea	net	vnv	cnt	Salidas
3	-	-	-	//Ingrese un número: '
4	1	-	-	//Lee net
5	1	0	-	//Inicializa vnm
6	1	0	-	//Inicia Para
7	1	0	1	//Verifica net
8	1	1	1	//Calcula vnm
10	1	1	2	//Finaliza Para
11	1	1	2	//Verifica vnm
12	-	-	-	//Linea no Ejecutada
14	1	1	2	//El número no 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:

Numero Inicial

Número Final

Salidas:

Números Primos

Procesos:

Si número inicial > número final Entonces

Auxiliar <- número inicial

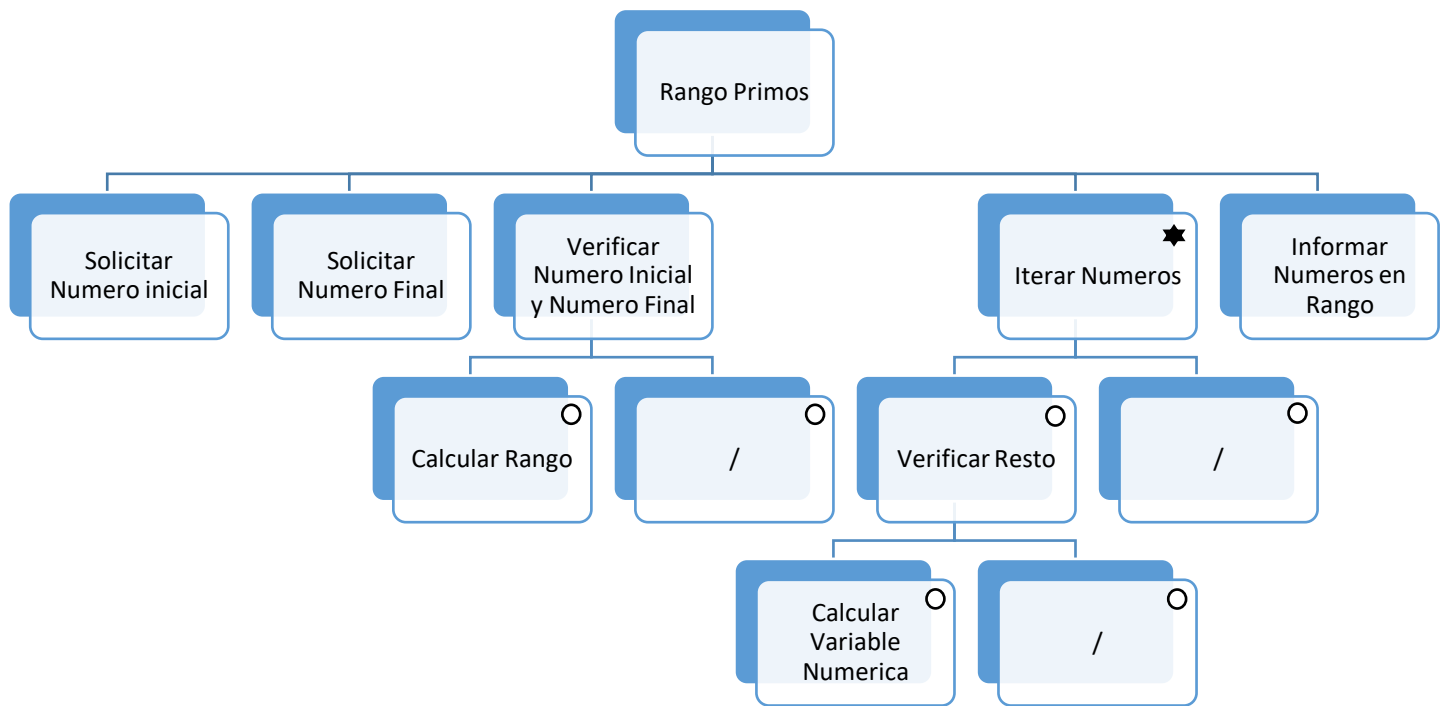
Numero inicial <- número final

Número final <- auxiliar

Si Numero % contador == 0 Entonces

Variable Numerica = Variable numerica + 1

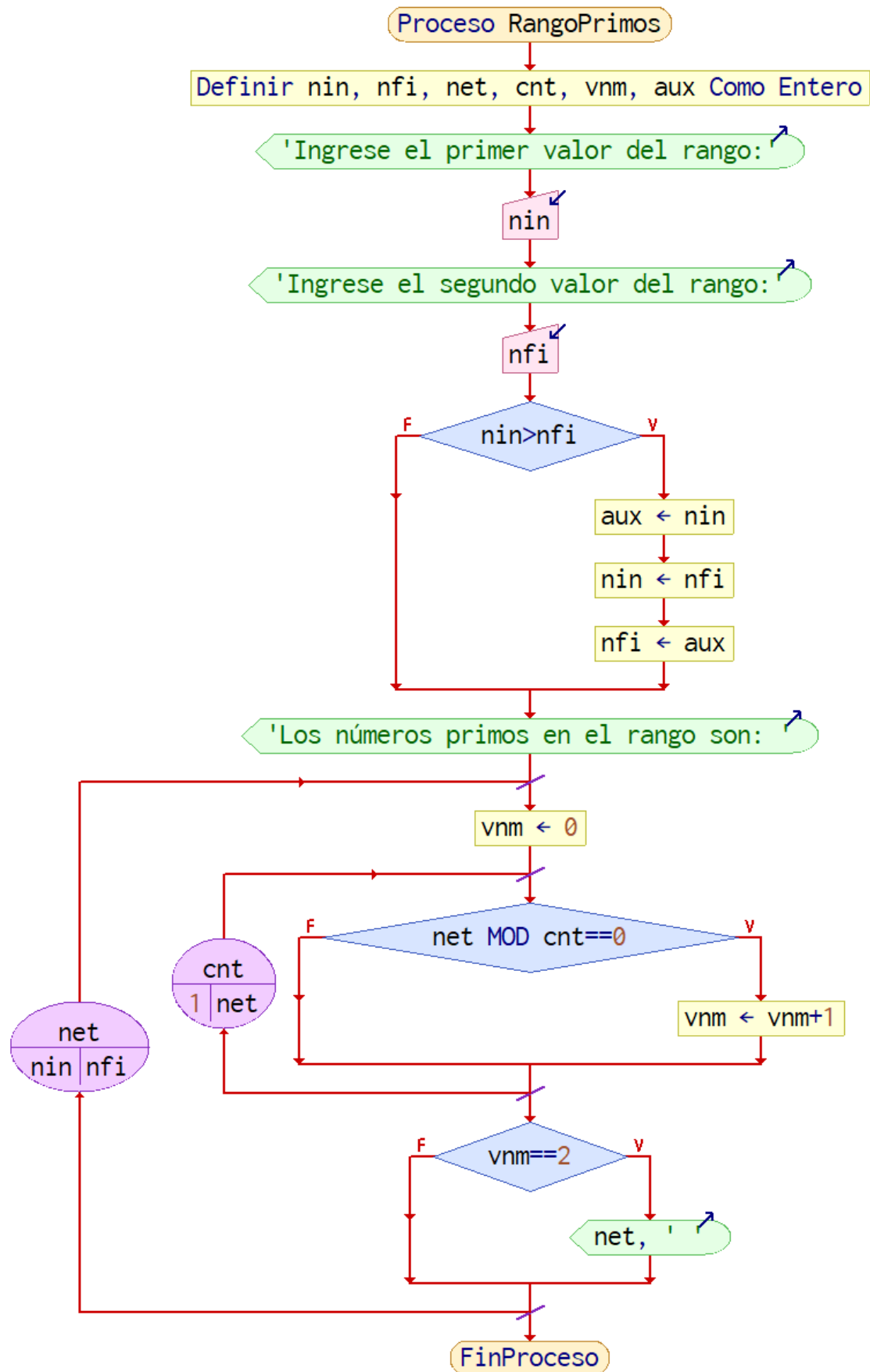
**Estrategia**



### Ambiente

Variable	Tipo	Definición
nin	Entero	Numero Inicial
nfi	Entero	Número Final
net	Entero	Numero Entero
cnt	Entero	Contador
vnm	Entero	Variable Numérica
aux	Entero	Variable Auxiliar

## Diagrama de flujo



## Seudocódigo

### Proceso RangoPrimos

Definir nin, nfi, net, cnt, vnm, aux Como Entero;

Escribir 'Ingrese el primer valor del rango:';

Leer nin;

Escribir 'Ingrese el segundo valor del rango:';

Leer nfi;

Si nin > nfi Entonces

    aux <- nin;

    nin <- nfi;

    nfi <- aux;

FinSi

Escribir 'Los números primos en el rango son: 'Sin Saltar;

Para net<-nin Hasta nfi Hacer

    vnm <- 0;

    Para cnt<-1 Hasta net Hacer

        Si net MOD cnt == 0 Entonces

            vnm <- vnm + 1;

        FinSi

    FinPara

    Si vnm == 2 Entonces

        Escribir net, ' 'Sin Saltar;

    FinSi

FinPara

### FinProceso

### Prueba de escritorio

Línea	nin	nfi	aux	net	vnm	cnt	Salida
3	-	-	-	-	-	-	//Ingrese Valor Inicial'
4	1	-	-	-	-	-	//Lee nin
5	1	-	-	-	-	-	//Ingrese Valor Final'
6	1	1	-	-	-	-	//Lee nfi
7	1	1	-	-	-	-	//Verifica nin
8	-	-	-	-	-	-	//Línea no ejecutada
9	-	-	-	-	-	-	//Línea no ejecutada
10	-	-	-	-	-	-	//Línea no ejecutada
12	1	1	-	-	-	-	//Los números primos en el rango son: '
13	1	1	-	1	-	-	//Inicia primer Para
14	1	1	-	1	0	-	//Inicializa vnm
15	1	1	-	1	0	-	//Inicia segundo Para
16	1	1	-	1	0	-	//Verifica net
17	1	1	-	1	1	-	//Calcula vnm
19	1	1	-	1	1	-	//Fin segundo para
20	1	1	-	1	1	-	//Verifica vnm

21	1	1	-	1	1	-	//net, ''
23	1	1	-	1	1	-	//Fin Primer Para

## 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:

- Muy Bueno ( 8 o más)
- Bueno (6 o 7)
- Regular (4 o 5)
- Insuficiente (3 o menos).

### Análisis

Datos:

- Número total de alumnos
- Nota del alumno

Salidas:

- Cantidad de notas
- Promedio
- Cantidad de aprobados y no aprobados

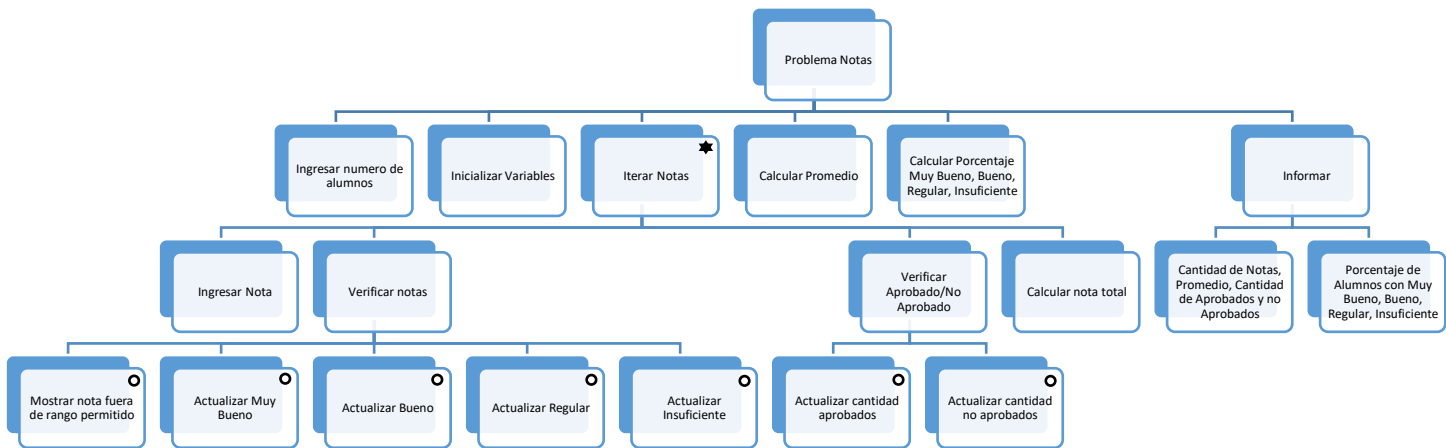
Porcentaje de alumnos con:

- Muy Bueno ( 8 o más)
- Bueno (6 o 7)
- Regular (4 o 5)
- Insuficiente (3 o menos)

Procesos:

- Promedio = suma total notas/suma total alumnos \*100
- Si nota > 5 entonces
  - Aprobado
- Sino
  - No Aprobado
- Porcentaje
  - Si nota >= 8 entonces
    - Muy Bueno
  - Si nota >= 6 y nota <= 7 entonces
    - Bueno
  - Si nota >=4 y nota <=5 entonces
    - Regular
  - Si nota <= 3 entonces
    - Insuficiente

Estrategia



Ambiente

Variable	Tipo	Definición
nta	Entero	Número de alumnos
cdn	Entero	Cantidad de notas
cda	Entero	Cantidad de aprobados
cna	Entero	Cantidad no aprobados
cmb	Entero	Cantidad muy bueno
cbu	Entero	Cantidad Bueno
cre	Entero	Cantidad Regular
cin	Entero	Cantidad Insuficiente
cnt	Entero	Variable Contador
nda	Real	Nota del Alumno
prm	Real	Promedio
pmb	Real	Porcentaje muy bueno
pbu	Real	Porcentaje bueno
pre	Real	Porcentaje regular
pin	Real	Porcentaje insuficiente
snt	Real	Suma de notas total

Diagrama de flujo

Proceso ProblemaNotas

Definir nta, cdn, cda, cna, cmb, cbu, cre, cin, cnt Como Entero

Definir nda, prm, pmb, pbu, pre, pin, snt Como Real

cdn  $\leftarrow$  0

cda  $\leftarrow$  0

cna  $\leftarrow$  0

cmb  $\leftarrow$  0

cbu  $\leftarrow$  0

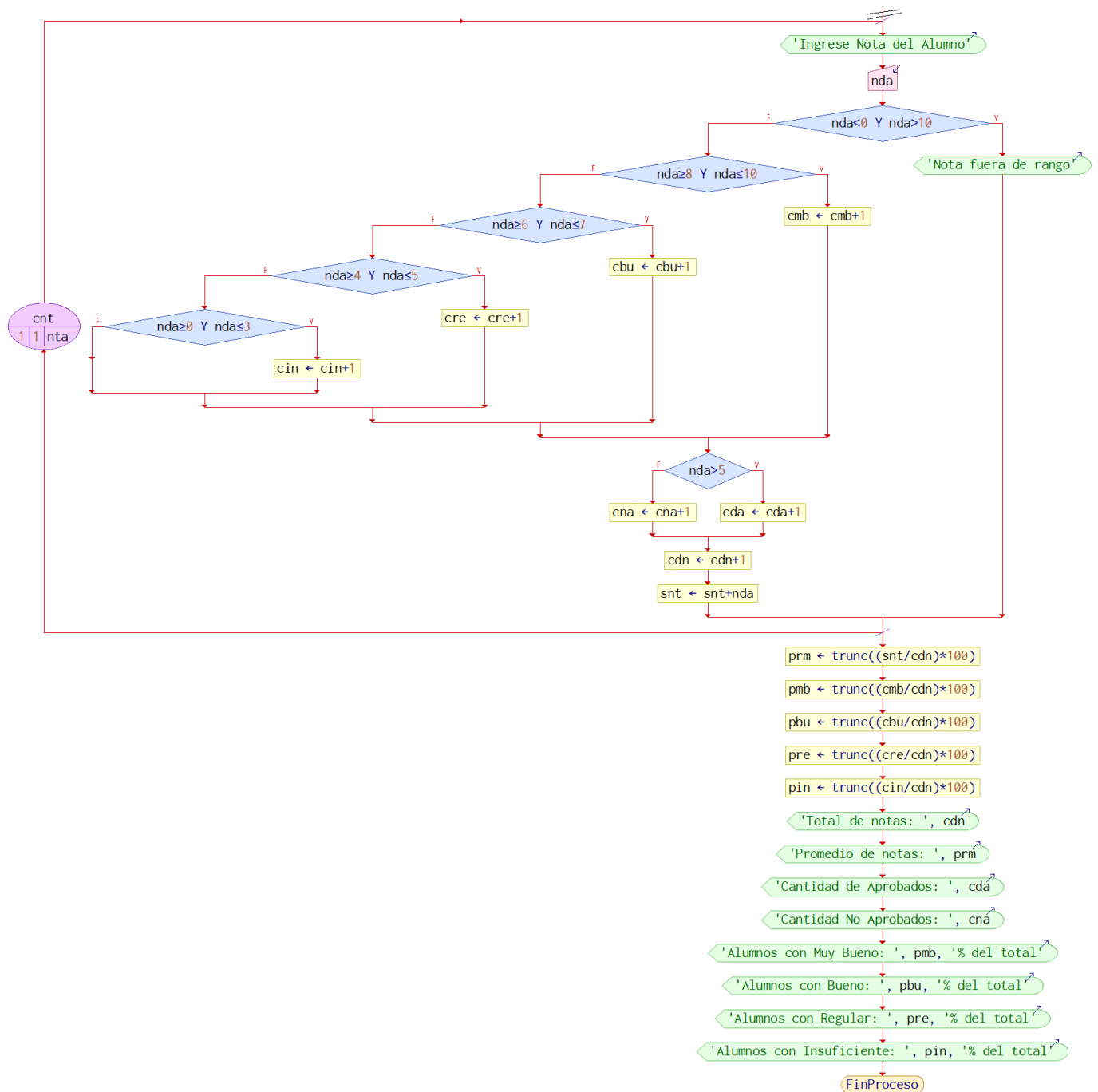
cre  $\leftarrow$  0

cin  $\leftarrow$  0

'Ingrese Numero de Alumnos'

nta





## Seudocódigo

### Proceso ProblemaNotas

Definir nta, cdn, cda, cna, cmb, cbu, cre, cin, cnt Como Entero;

Definir nda, prm, pmb, pbu, pre, pin, snt Como Real;

cdn ← 0;

cda ← 0;

cna ← 0;

cmb ← 0;

cbu ← 0;

cre ← 0;

cin ← 0;

Escribir 'Ingrese Numero de Alumnos';

Leer nta;

Para cnt<-1 Hasta nta Con Paso 1 Hacer

Escribir "Ingrese Nota del Alumno";

Leer nda;

Si nda < 0 y nda > 10 Entonces

Escribir "Nota fuera de rango";

SiNo

Si nda >= 8 y nda <= 10 Entonces

cmb <- cmb + 1;

SiNo

Si nda >= 6 y nda <= 7 Entonces

cbu <- cbu + 1;

SiNo

Si nda >= 4 y nda <= 5 Entonces

cre <- cre + 1;

SiNo

Si nda >= 0 y nda <= 3 Entonces

cin <- cin + 1;

FinSi

FinSi

FinSi

FinSi

Si nda > 5 Entonces

cda <- cda + 1;

SiNo

cna <- cna + 1;

FinSi

cdn <- cdn + 1;

snt <- snt + nda;

FinSi

FinPara

prm <- snt / cdn;

pmb <- trunc((cmb / cdn)\*100);

```

pbu <- trunc((cbu / cdn)*100);
pre <- trunc((cre / cdn)*100);
pin <- trunc((cin / cdn)*100);
Escribir "Total de notas: ", cdn;
Escribir "Promedio de notas: ", prm;
Escribir "Cantidad de Aprobados: ", cda;
Escribir "Cantidad No Aprobados: ", cna;
Escribir "Alumnos con Muy Bueno: ", pmb,"% del total";
Escribir "Alumnos con Bueno: ", pbu,"% del total";
Escribir "Alumnos con Regular: ", pre,"% del total";
Escribir "Alumnos con Insuficiente: ", pin,"% del total";

```

FinProceso

### Prueba de Escritorio

Línea	cdn	cda	cna	cmb	cbu	cre	cin	nta	nda	snt	prm	pmb	pbu	pre	pin	Salida
4	0	-	-	-	-	-	-	-	-	-	-	-	-	-	-	//Inicializa cdn
5	0	0	-	-	-	-	-	-	-	-	-	-	-	-	-	//Inicializa cda
6	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-	//Inicializa cna
7	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	//Inicializa cmb
8	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	//Inicializa cbu
9	0	0	0	0	0	0	-	-	-	-	-	-	-	-	-	//Inicializa cre
10	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	//Inicializa cin
11	0	0	0	0	0	0	0	-	-	-	-	-	-	-	-	//Ingrese Número de Alumnos'
12	0	0	0	0	0	0	0	1	-	-	-	-	-	-	-	//Lee nta
13	0	0	0	0	0	0	0	1	-	-	-	-	-	-	-	//Inicia Para
14	0	0	0	0	0	0	0	1	-	-	-	-	-	-	-	//Ingrese Nota del Alumno'
15	0	0	0	0	0	0	0	1	8	-	-	-	-	-	-	//Lee nda
16	0	0	0	0	0	0	0	1	8	-	-	-	-	-	-	//Verifica nda
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	//Linea no Ejecutada
19	0	0	0	0	0	0	0	1	8	-	-	-	-	-	-	//Verifica nda
20	0	0	0	1	0	0	0	1	8	-	-	-	-	-	-	//Calcula cmb
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	//Linea no Ejecutada
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	//Linea no Ejecutada
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	//Linea no Ejecutada
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	//Linea no Ejecutada
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	//Linea no Ejecutada
29	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	//Linea no Ejecutada
34	0	0	0	1	0	0	0	1	8	-	-	-	-	-	-	//Verifica nda
35	0	1	0	1	0	0	0	1	8	-	-	-	-	-	-	//Calcula cda
37	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	//Linea no Ejecutada
39	1	1	0	1	0	0	0	1	8	-	-	-	-	-	-	//Calcula cdn
40	1	1	0	1	0	0	0	1	8	8	-	-	-	-	-	//Calcula snt
42	1	1	0	1	0	0	0	1	8	8	-	-	-	-	-	//Finaliza Para
43	1	1	0	1	0	0	0	1	8	8	8	-	-	-	-	//Calcula prm
44	1	1	0	1	0	0	0	1	8	8	8	100	-	-	-	//Calcula pmb
45	1	1	0	1	0	0	0	1	8	8	8	100	0	-	-	//Calcula pbu
46	1	1	0	1	0	0	0	1	8	8	8	100	0	0	-	//Calcula pre
47	1	1	0	1	0	0	0	1	8	8	8	100	0	0	0	//Calcula pin
48	1	1	0	1	0	0	0	1	8	8	8	100	0	0	0	//Total de notas: 1'
49	1	1	0	1	0	0	0	1	8	8	8	100	0	0	0	//Promedio de notas: 8'
50	1	1	0	1	0	0	0	1	8	8	8	100	0	0	0	//Cantidad de Aprobados: 1'
51	1	1	0	1	0	0	0	1	8	8	8	100	0	0	0	//Cantidad No Aprobados: 0'
52	1	1	0	1	0	0	0	1	8	8	8	100	0	0	0	//Alumnos con Muy Bueno: 100% del total '
53	1	1	0	1	0	0	0	1	8	8	8	100	0	0	0	//Alumnos con Bueno: 0% del total '
54	1	1	0	1	0	0	0	1	8	8	8	100	0	0	0	//Alumnos con Regular: 0% del total '
55	1	1	0	1	0	0	0	1	8	8	8	100	0	0	0	//Alumnos con Insuficiente: 0% del total '

### 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:

Cantidad por unidad

### Precio por unidad

Salidas:

Total a pagar

Cantidad total de productos

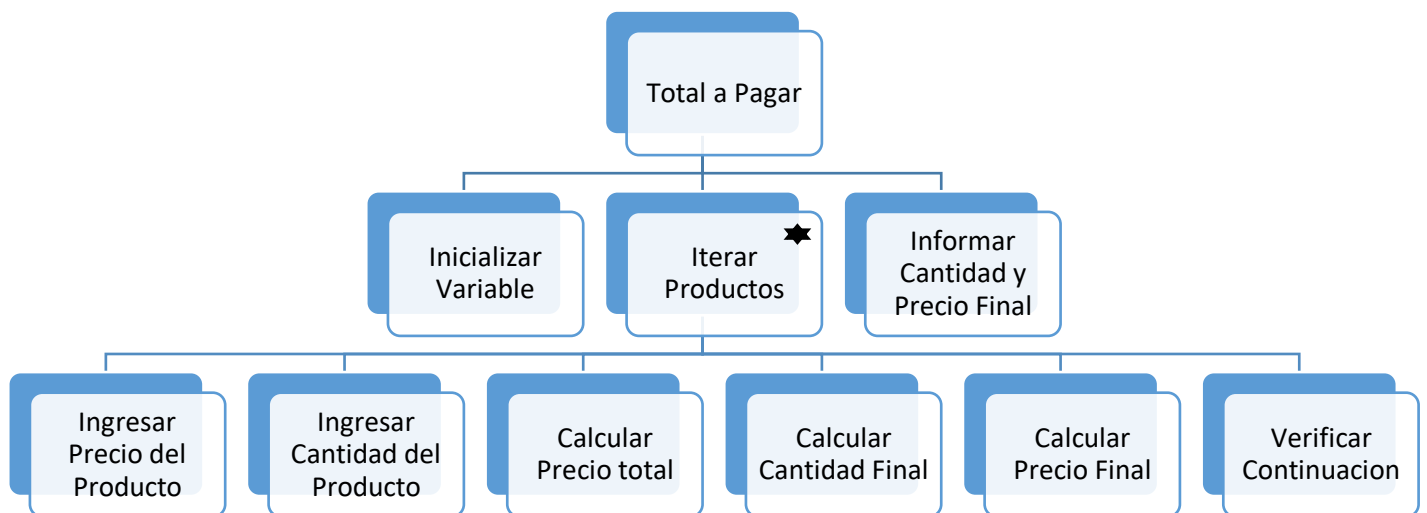
### Procesos:

Precio total = precio por unidad \* cantidad por unidad

Cantidad Final = cantidad final + cantidad por unidad

Precio Final = Precio final + precio total

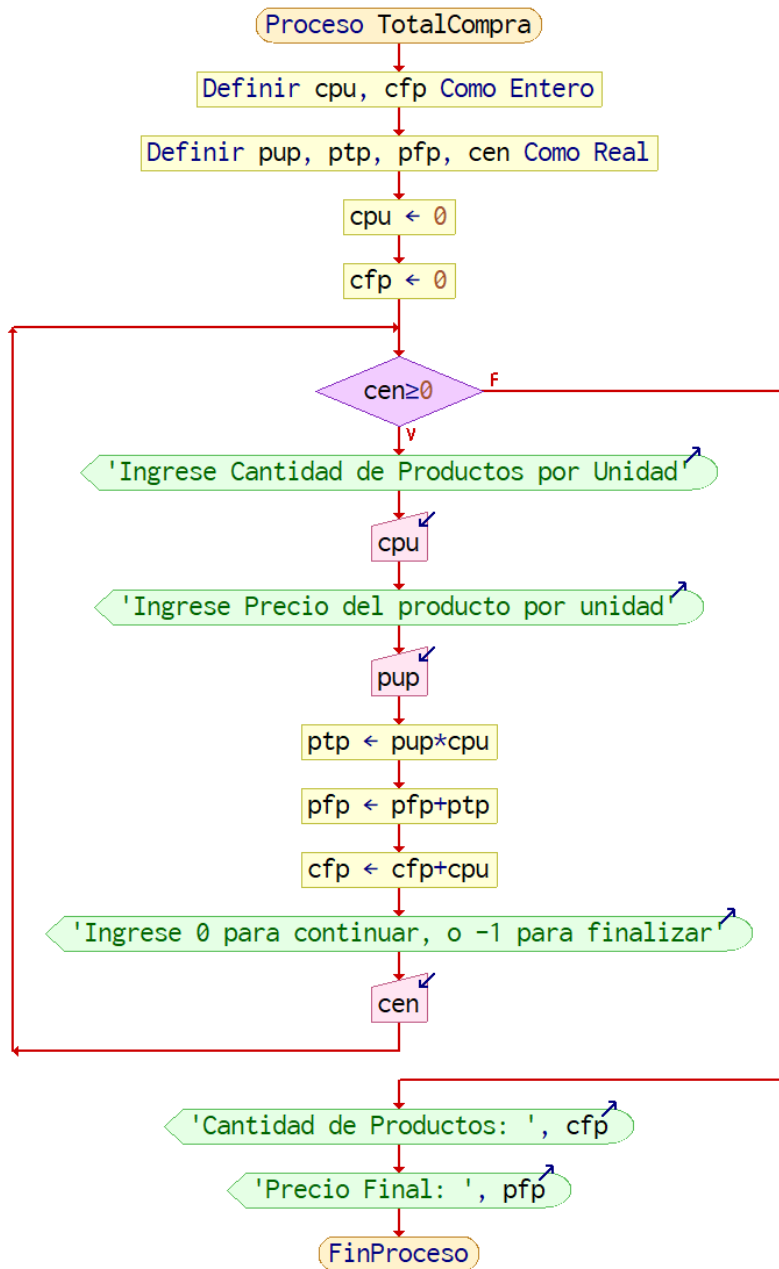
## Estrategia



## Ambiente

Variable	Tipo	Definición
cpu	Entero	Cantidad Producto por Unidad
cfp	Entero	Cantidad Final Producto
pup	Real	Precio Producto por unidad
ptp	Real	Precio total
pfp	Real	Precio Final
cen	Real	Variable Centinela

## Diagrama de flujo



## Seudocódigo

### Proceso TotalCompra

Definir cpu, cfp Como Enteros;

Definir pup, ptp, pfp, cen Como Real;

cpu <- 0;

cfp <- 0;

Mientras cen >= 0 Hacer

    Escribir "Ingrese Cantidad de Productos por Unidad";

    Leer cpu;

    Escribir "Ingrese Precio del producto por unidad";

    Leer pup;

ptp <- pup \* cpu;

pfp <- pfp + ptp;

cfp <- cfp + cpu;

Escribir "Ingrese 0 para continuar, o -1 para finalizar";

Leer cen;

FinMientras

Escribir "Cantidad de Productos: ", cfp;

Escribir "Precio Final: ", pfp;

FinProceso

### Prueba de Escritorio

Linea	cpu	cfp	pup	ptp	pfp	cen	Salidas
4	0	-	-	-	-	-	//Inicializa cpu
5	0	0	-	-	-	-	//Inicializa cfp
6	0	0	-	-	-	-	//Inicia Mientras
7	0	0	-	-	-	-	//Ingrese Cantidad de Productos por Unidad'
8	2	0	-	-	-	-	//Lee cpu
9	2	0	-	-	-	-	//Ingrese Precio del producto por unidad'
10	2	0	2000	-	-	-	//Lee pup
11	2	0	2000	4000	-	-	//Calcula ptp
12	2	0	2000	4000	4000	-	//Calcula pfp
13	2	2	2000	4000	4000	-	//Calcula cfp
14	2	2	2000	4000	4000	-	//Ingrese 0 para continuar, o -1 para finalizar'
15	2	2	2000	4000	4000	-1	//Lee cen
16	2	2	2000	4000	4000	-1	//Finaliza Mientras
17	2	2	2000	4000	4000	-1	//Cantidad de Productos: 2'
18	2	2	2000	4000	4000	-1	//Precio Final: \$4000'