

UNIDAD 4

INTRODUCCIÓN A LA PROGRAMACIÓN

Que es programar

Dar una serie de instrucciones al ordenador para indicarle lo que tiene que hacer

Que es un programa

Secuencia ordenada de instrucciones que un ordenador es capaz de ejecutar



1. INTRODUCCIÓN

¿Cómo creo programas?

1. Es fundamental comprender el problema que queremos resolver al detalle, sin dudas y sin que haya posibles interpretaciones.
2. Analizar los datos que son necesarios para obtener el resultado (datos de entrada) y sus características.

¿Cómo comprender el problema que debo resolver?

Tipos

- * Problemas grandes y complejos: hay que descomponerlo en problemas mas sencillos que podamos manejar.
- * Problemas pequeños y sencillos: analizando los datos que piden y creando diagramas o modelos visuales que nos ayudan a comprobar su correcto funcionamiento.

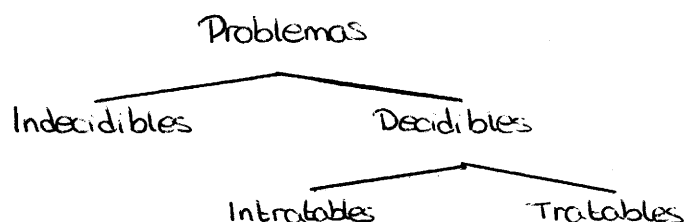
Utilizando modelos de desarrollo de software

Existen muchos y diversos

Ej: 2,10 m.c.m:20 m.c.D:2

Comprender el problema es el paso más importante!

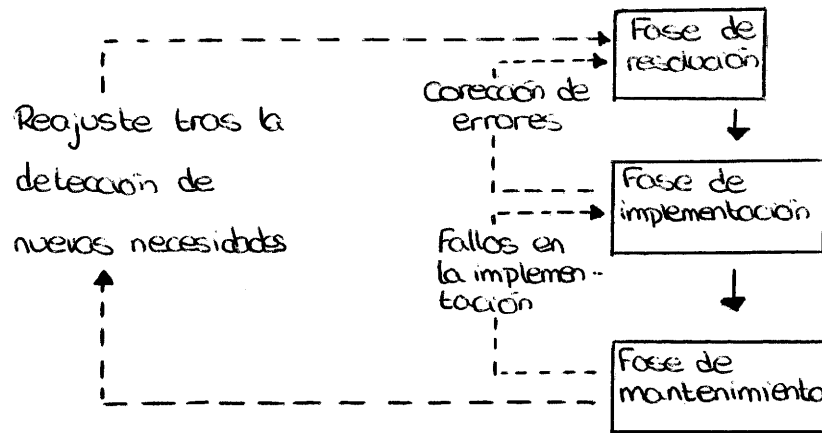
1.1 Clasificación de problemas



Factorial: Se deben multiplicar todos los números naturales de 0 (incluido) en adelante

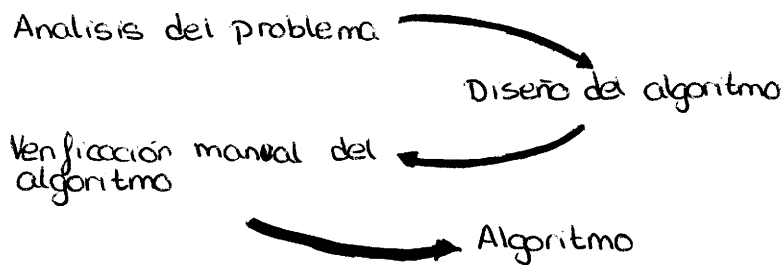
1.2 Fases para resolver un problema

Resolución de un problema



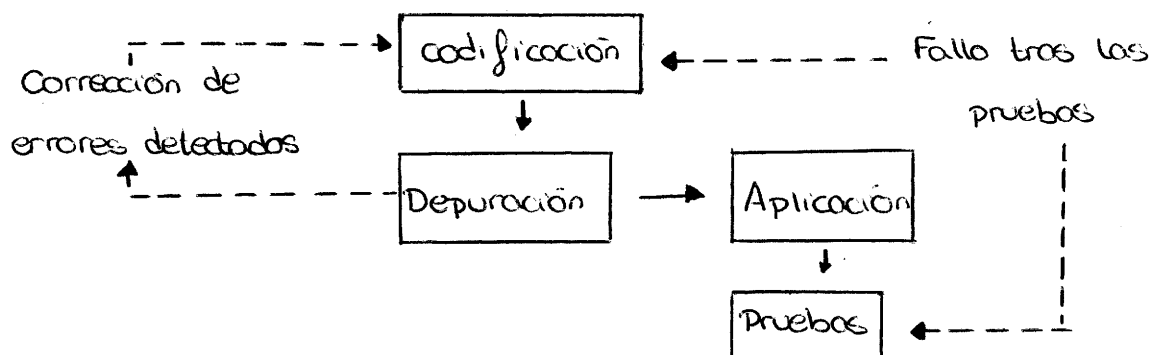
- Fase de resolución

El resultado de esta fase será un algoritmo verificado.



- Análisis del problema: Debemos estudiar el problema hasta comprenderlo. Obtendremos las especificaciones del problema.
- Diseño del problema: Describe la solución del problema (diagrama de flujo). Los problemas complejos debemos dividirlos en problemas más sencillos.
- Verificación manual del algoritmo: Probar el algoritmo con un conjunto de datos que incluya todos los casos posibles.

- Fase de implementación



- **Codificación:** se traduce el algoritmo diseñado en la fase anterior en el lenguaje de programación que usemos.

Tras la codificación, debemos traducir el lenguaje de alto nivel a código máquina: **compilar el programa**

- * **Compiladores**, realiza un análisis de todo el código mostrando un listado completo de errores. El proceso de traducción se realiza una vez solo.
- * **Interpretes**, las instrucciones se analizan una a una (javascript)
- **Depuración:** consiste en corregir los errores encontrados tras la fase de compilación y pruebas

El proceso de depuración incluye:

- * **Compilar el programa**
 1. Analizar el listado de errores
 2. Ejecutar el conjunto de prueba
- * **Editar y corregir el texto del programa**
- * **Volver al paso inicial**

- **Fase de explotación y mantenimiento**

- * **Ajustes:** Tras implantar la aplicación, suele ser necesario realizar unos pequeños ajustes
- * **Mejoras:** Se pueden mejorar diversos aspectos de la aplicación: rapidez, sencillez, etc.
- * **Adaptaciones:** Con el paso del tiempo el programa hay que adaptarlo a nuevos requerimientos, no previstos inicialmente

Ejercicio 1 m.c.m

Entrada: Pedimos al usuario 2 o más números naturales

- Programa:**
1. Descomponemos dichos números en sus factores primos. Una vez descompuestos los números, recogeremos sus factores.
 2. Cogemos los factores comunes y no comunes, entre el nº de números que hayamos descompuesto. En caso de que haya un factor con mayor exponente cogemos este.
 3. Una vez seleccionados los factores los multiplicamos.

Salida: Una vez realizada dicha multiplicación, nos dará como resultado el m.c.m

Ejercicio 2 m.c.d

Entrada: Pedimos al usuario 2 o más números naturales.

- Programa:**
1. Escribiremos todos los divisores del número que nos ha dicho. (los divisores de dicho nº siempre empezaron por 1 y terminará por el, por ejemplo: 18 empezará por 1 seguido de 2, 3, 6, 9 y terminando en 18)
 2. Seleccionamos el divisor común más grande que tengan dichos números

Salida: El mayor divisor común entre esos números será el resultado.

m.c.m: Es el número positivo más pequeño que es múltiplo de dos o más números excluyendo al 0

nº primos: números que solo son divisibles entre ellos mismos y el 1

m.c.d: Es el mayor número que divide exactamente a dos o más números excluyendo al 0

Divisor: número que al dividirlo con otro da de resto 0

2. Definición del algoritmo.

- * Un algoritmo es un método para resolver un problema mediante una serie finita y ordenada de pasos bien definidos.
- * El programador es una persona que resuelve problemas mediante el diseño de algoritmos.
- * Características de un algoritmo: Preciso, sin ambigüedades, bien definido, finito, independiente del lenguaje de programación, no son únicos.

3. DISEÑO DE ALGORITMOS

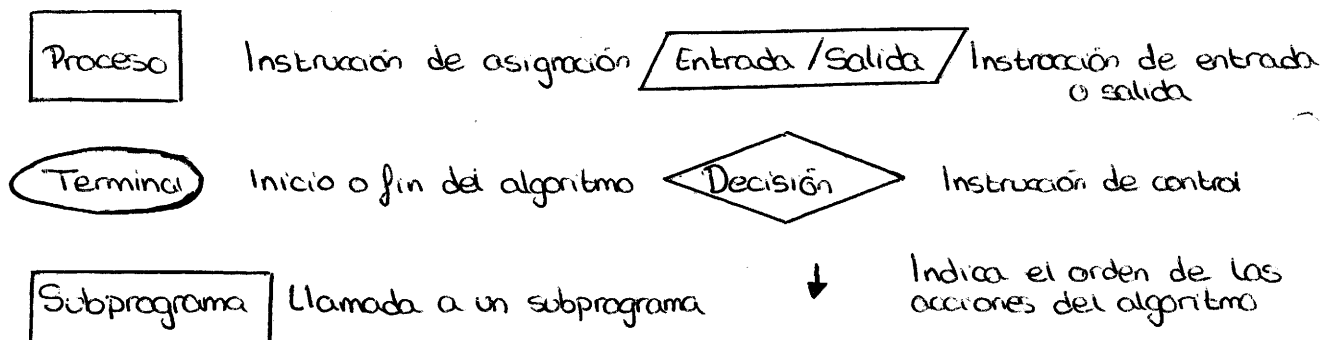
Existen diversas maneras de diseñar un algoritmo.

- * Lenguaje natural, se utiliza una descripción textual para explicar cómo resolver un problema.
- * Diagramas de flujo, representan de forma gráfica la secuencia lógica de pasos que sigue el programa.
- * Pseudocódigo, utiliza una estructura similar a un lenguaje de programación pero no tan estricto, y más cercano al lenguaje natural.

3.1 Diagramas de flujo

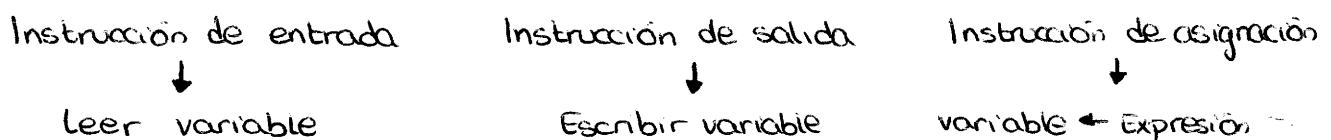
- Características de los diagramas de flujo.
 - * Son visuales
 - * Adecuados para diagramas no muy grandes
 - * Los símbolos están conectados por líneas

- Símbolos de los diagramas de flujo



- Conector de reagrupamiento de una instrucción de control.

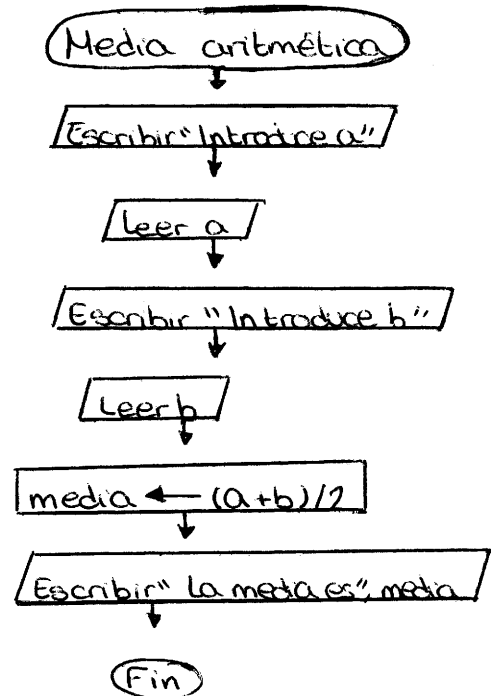
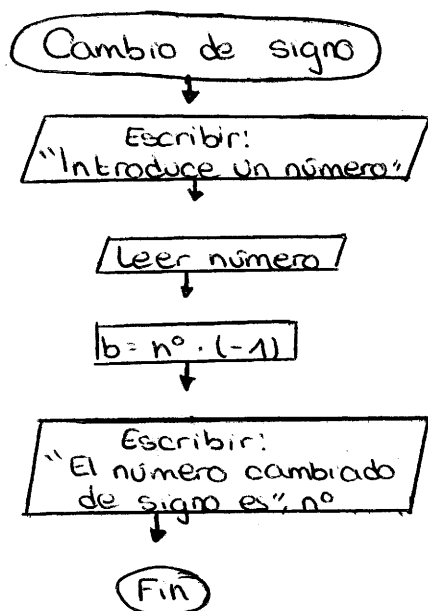
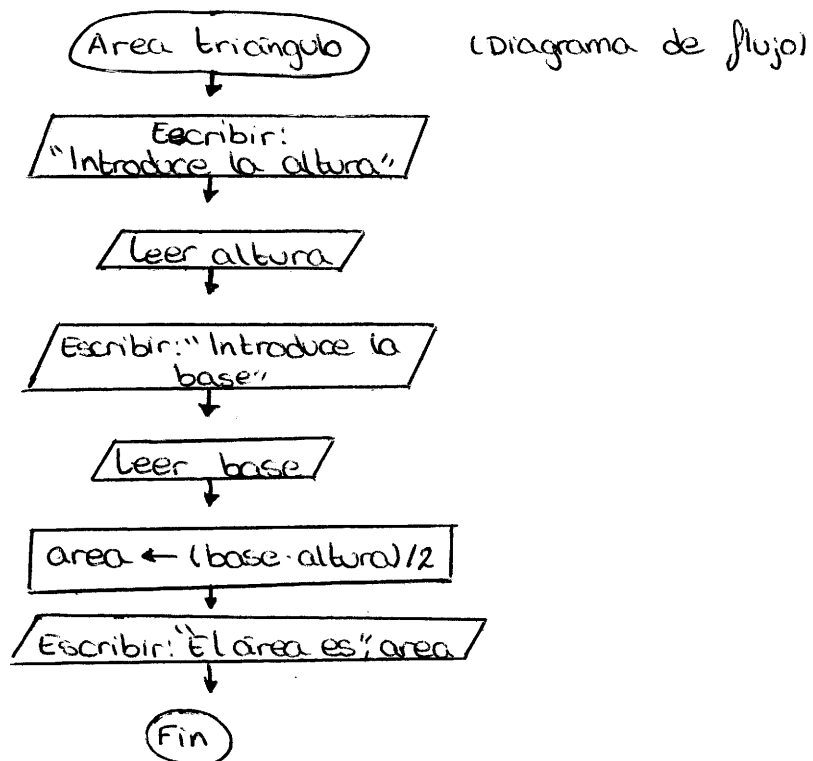
- Instrucciones primitivas

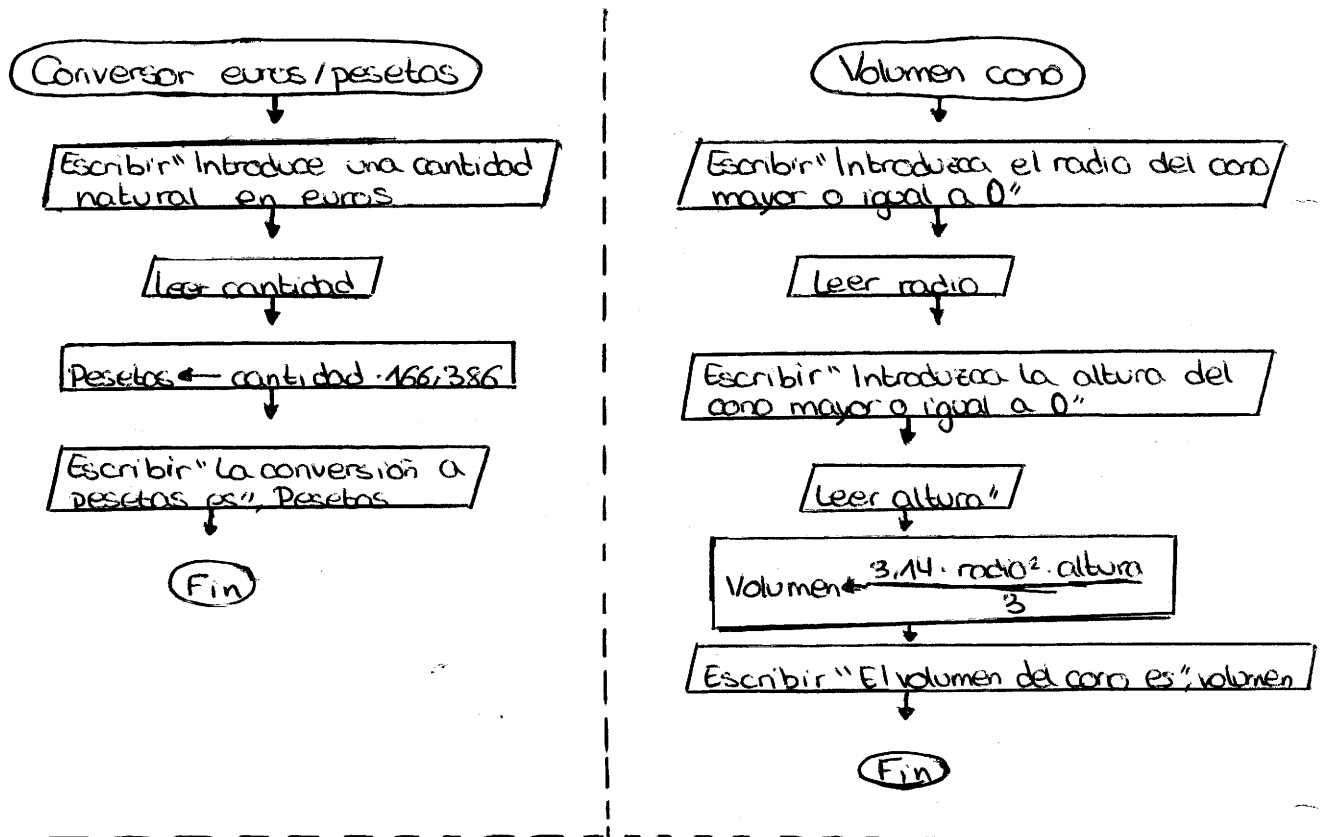


La combinación de estos tres acciones primitivas da lugar a un diagrama secuencial

Ejemplo Calcular el área de un triángulo.

- ¿Cómo se calcula el área de un triángulo? $\frac{b \cdot h}{2}$
- ¿Qué datos necesito? ¿Qué valores pueden tomar esos datos? la base y la altura reales mayores que 0
- ¿Qué resultado se mostrará? El área del triángulo.
- ¿Cuándo acaba el algoritmo? Cuando se muestra el resultado al usuario.





3.2 Variables y tipos de datos

-- Concepto y definición de variable

- Es un dato que se aloja en la memoria del ordenador
- El nombre o identificador que damos a la variable está asociado a la dirección de la memoria.
- Los datos pueden ser de distintos tipos:
 - * Caracter = Para almacenar cadena de caracteres, se encierran entre comillas simples o dobles.
 - * Entero = Para almacenar números sin decimales.
 - * Real = Para almacenar con decimales
 - * Lógico = Para almacenar los valores verdaderos o falso.

-- Declarar variables en diagrama de flujo

- Se declaran al principio del diagrama, antes de leer, escribir u operar con la variable
- Se pueden declarar juntas o separadas por comas para variables del mismo tipo.

Operaciones aritméticas

Declarar, $n1$, $n2$, suma, resta, multiplicación y división como Real

Escribir "Introduce $n1$ "

Leer $n1$

Escribir "Introduce $n2$ "

Leer $n2$

$\text{suma} \leftarrow n1 + n2$

$\text{resta} \leftarrow n1 - n2$

$\text{multiplicación} \leftarrow n1 \cdot n2$

$\text{división} \leftarrow n1 / n2$

Escribir "Los resultados son, suma, resta, multiplicación y, división"

Fin

Cambio de signo

Declarar $número$ y b como Real

Escribir "Introduce un número"

Leer $número$

$b \leftarrow n° \cdot (-1)$

Escribir "El $n°$ cambiado de signo es", b

Fin

Area triángulo

Declarar, altura, base y area como Real

Escribir "Introduce la base"

Leer base

Escribir "Introduce la altura"

Leer altura

$\text{area} \leftarrow (b \cdot h) / 2$

Escribir "El area es", area

Fin

Media aritmética

Declarar, a , b y media como Real

Escribir "Introduce a "

Leer a

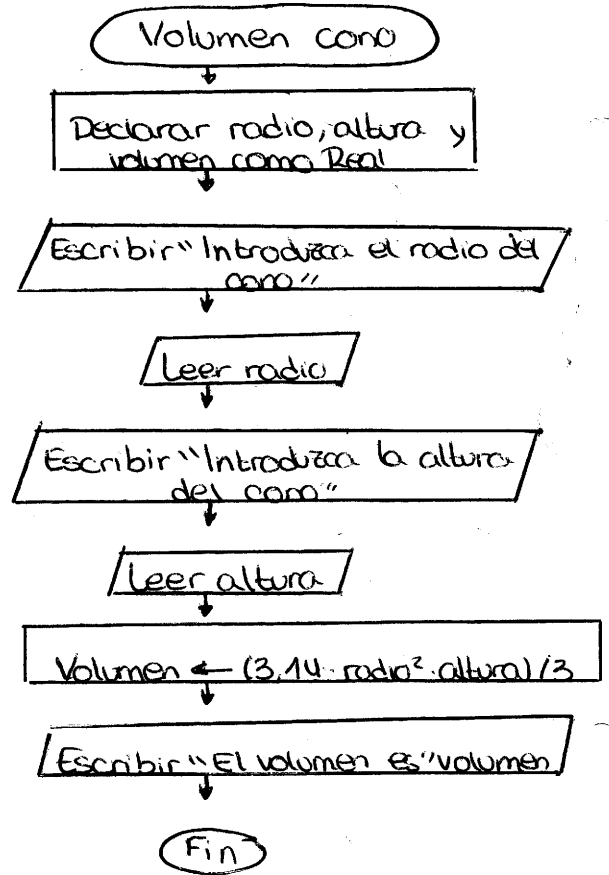
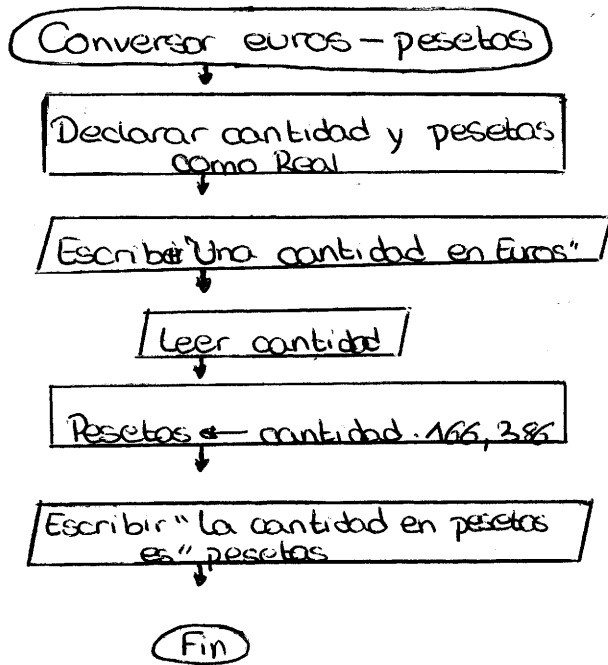
Escribir "Introduce b "

Leer b

$\text{media} \leftarrow (a + b) / 2$

Escribir "La media es", media

Fin



4 OPERADORES Y EXPRESIONES

- Una expresión es una sentencia que se puede evaluar dando como resultado un valor de un tipo de dato determinado
- Los operadores permiten realizar una determinada operación sobre valores, variables u otras expresiones.
- Tipos
 - * Operadores aritméticos
 - * Operadores de comparación o relacionales
 - * Operadores lógicos

4.1 Operadores aritméticos

Operador	Nombre	Ejemplo	Descripción
+	Suma	$a+b+c$ $3+4+edad$	Realiza la suma de los operadores
-	Resta	número-5 $b-c$	Resta los operadores
*	Multiplicación	$a \times b$ 3×4	Multiplica los operadores
/	División	a / b número/10	Divide los operadores (Si es con no enteros, el resultado será entero)
% o mod	Módulo o resto	$a \% b$ $b \%$	Obtiene el resto de una división entera

4.2 Operadores de comparación

Comparan los valores de la expresión e indican si el resultado es verdadero o falso.

Operador	Nombre	Ejemplo	Descripción
<code>==</code>	igual	<code>a == b</code>	a es igual a b
<code>!=</code>	distinto	<code>a != b</code>	a es distinto de b
<code><</code>	menor que	<code>a < b</code>	a es menor que b
<code>></code>	mayor que	<code>a > b</code>	a es mayor que b
<code><=</code>	menor o igual que	<code>a <= b</code>	a es menor o igual que b
<code>>=</code>	mayor o igual que	<code>a >= b</code>	a es mayor o igual que b

4.3 Operadores lógicos

Los operadores lógicos se emplean para determinar la veracidad o falsedad de un conjunto de expresiones de comparación.

Operador	Nombre	Ejemplo	Descripción
<code>&&</code>	y	<code>(7 > 2) && (2 < 4)</code>	Las condiciones son verdaderas
<code> </code>	o	<code>(7 > 2) (2 < 4)</code>	Al menos una de las condiciones es verdad
<code>!</code>	no	<code>!(7 > 2)</code>	La condición es falsa

Los operadores lógicos se pueden utilizar para formar expresiones más complejas.

¡Tabla de la verdad de operadores lógicos!

A	B	<code>A && B</code>	<code>A B</code>	<code>!A !B</code>
V	V	V	V	F F
V	F	F	V	F V
F	V	F	V	V F
F	F	F	F	V V

Ejercicio 1

`a=1 b=2 c=3 d=4`

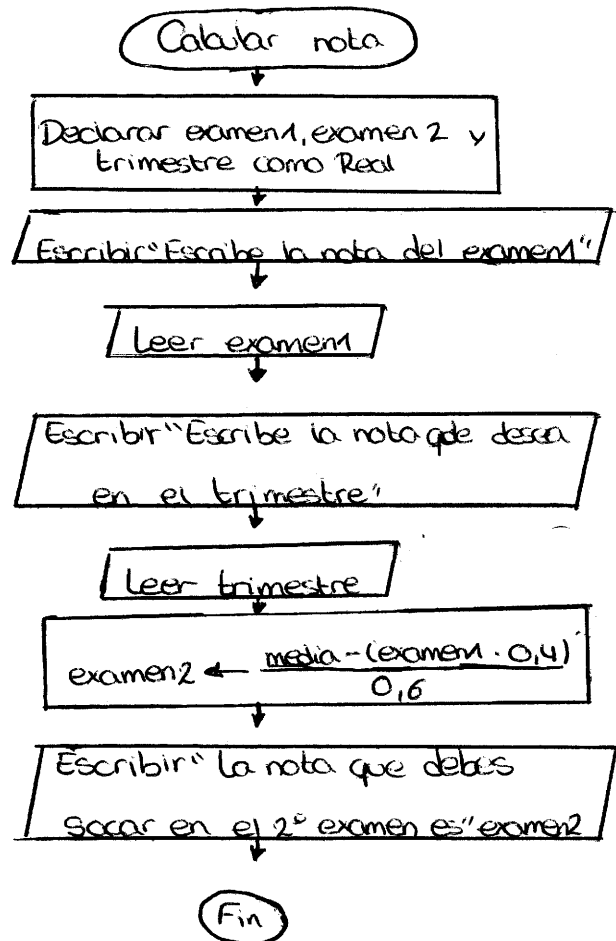
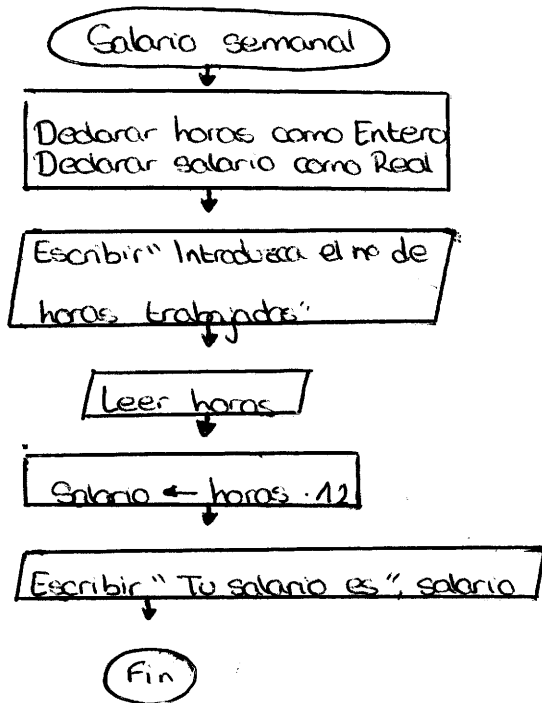
<code>a+b-c+d</code>	<code>1+2-3+4</code>	4
<code>a*b/c</code>	<code>1*2/3</code>	0
<code>1*a*b%c</code>	<code>1*1*2%3</code>	2
<code>a+d%b-c</code>	<code>1+4%2-3</code>	-2
<code>d+c/b+a</code>	<code>4+3/2-1</code>	4

$a < c$	$1 < 3$	Verdadero	$(a+b) \geq c$	$(1+2) \geq 3$	Verdadero
$b \leq c$	$2 \leq 3$	Verdadero	$(a+b) == c$	$(1+2) == 3$	Verdadero
$c \leq a$	$3 \leq 1$	Falso	$a != b$	$1 != 2$	Verdadero
$a > b$	$1 > 2$	Falso	$(a+b) != c$	$(1+2) != 3$	Falso
$b > c$	$2 > 3$	Falso			

$a=5$ $b=7$ $c=17$

$c/b == 2$	$17/7 == 2$	$2 == 2$	Verdadero
$c \% b \leq a \% b$	$17 \% 7 \leq 5 \% 7$	$3 \leq 5$	Verdadero
$b + c / a != c - a$	$7 + 3 != 17 - 5$	$10 != 12$	Verdadero
$(b < c) \& \& (c == 7)$	$(7 < 17) \& \& (17 == 7)$	$V \& \& F$	Falso
$(c + 1 - b == 0) \parallel (b == 5)$	$(17 + 1 - 7 == 0) \parallel (7 == 5)$		Falso

Ejercicio 2



Ejercicio 3 Escribe las expresiones para cada una de las fórmulas siguientes.

$$x = \frac{a + \frac{b}{c}}{a} + d$$

$$(a + b/c) / a + d$$

$$x = \frac{a + \frac{b}{c}}{\frac{a}{b} + c}$$

$$(a + b/c) / (a/b + c)$$

$$x = a \frac{a + b \frac{c}{d \cdot a}}{a + b \cdot \frac{c}{d}}$$

$$a * (a + b * (c / (d * a))) / (a + b * (c / d))$$

- $(C1 + C2 + C3) / 3 > 0 \ \&\& \ (C1 + C2 + C3) / 3 < 100$
- $(n \% 2 == 0) \ \&\& \ 10 \ \&\& \ (n \% 2 != 0) < 5$
- $(2 * edad) > 100 \ \&\& \ nacionalidad == "española"$
- $(k == 0) \ \&\& \ (k == 3) \ \&\& \ (k * 2) > 9$

Ejercicio 4 Equivalencia de expresiones

$!(A == B)$	Equivalentes	$A != B$
$!((A == B) \ \&\& \ (A == C))$	Equivalentes	$(A != B) \ \&\& \ (A != C)$
$!((A == B) \ \&\& \ (C > D))$	Equivalentes	$(A != B) \ \&\& \ (C <= D)$
$!A \ \&\& \ B$	Equivalentes	$B \ \&\& \ !A$
$!A \ \&\& \ B$	Equivalentes	$B \ \&\& \ !A$
$!(A \ \&\& \ B)$	No equivalentes	$A \ \&\& \ B$
$A \ \&\& \ B \ \&\& \ C$	No equivalentes	$A \ \&\& \ (B \ \&\& \ C)$
$(A \ \&\& \ B \ \&\& \ C)$	No equivalentes	$!(A \ \&\& \ B \ \&\& \ C)$

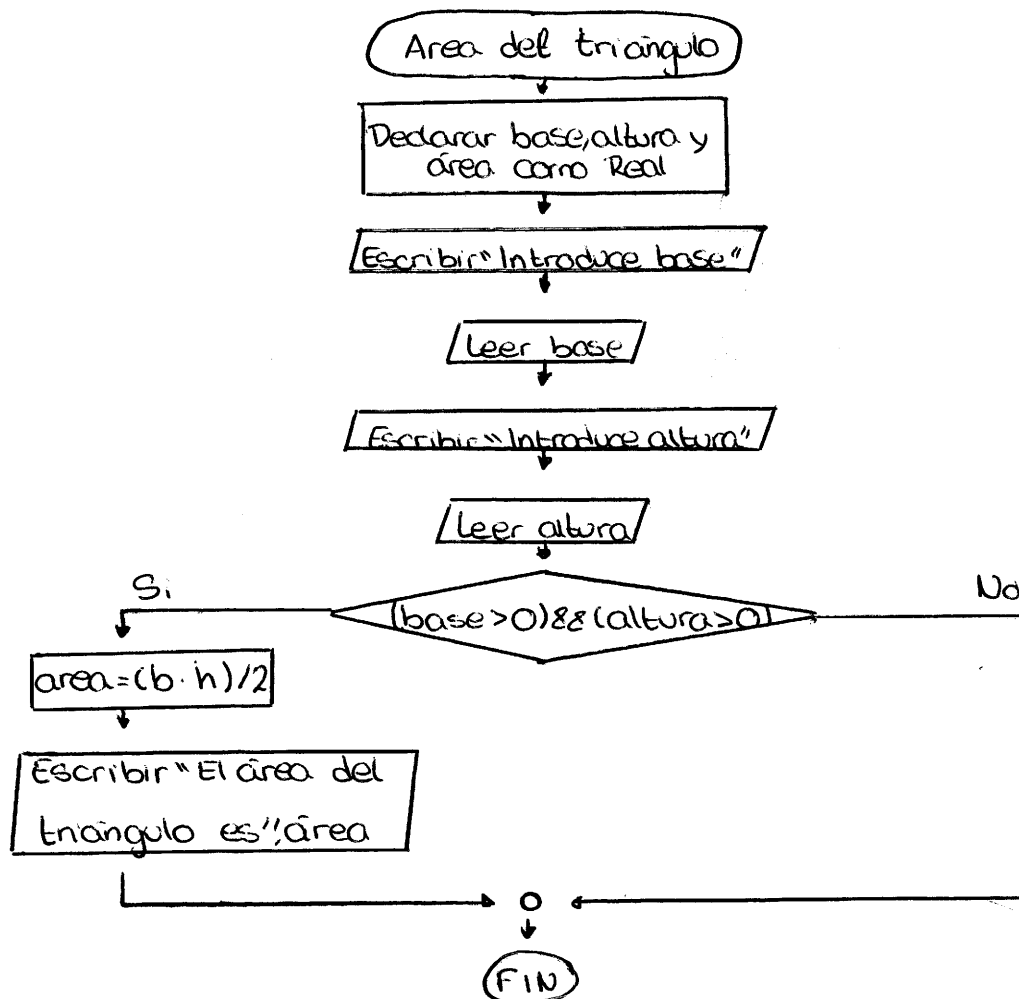
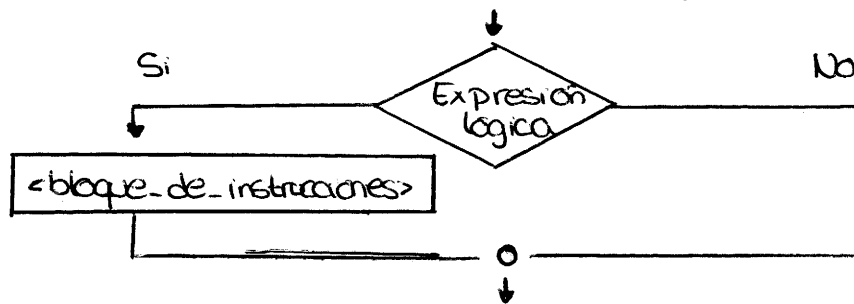
5 ESTRUCTURAS DE CONTROL DE FLUJO

- Se puede demostrar un algoritmo con 3 estructuras solamente
 - * Secuencial: las sentencias se ejecutan una detrás de otra
 - * Condicional: Se ejecutan o no, sentencias en función del valor de verdad de una condición. Operadores condicionales o lógicos.
 - * Cíclica: Se ejecutan repetidamente unas sentencias mientras se cumpla una determinada condición. Operadores condicionales o lógicos.
- Instrucciones condicionales, permiten bifurcar el código del programa en función de determinadas condiciones:
 - * Condicional simple
 - * Condicional doble
 - * Condicional múltiple.

- Instrucciones cíclicas, permiten repetir bloques de código.
 - * Mientras: se ejecuta mientras se cumpla una condición, primero pregunta y después se ejecuta
 - * Hacer-mientras: se ejecuta mientras se cumpla una condición, primero se ejecuta y después pregunta.
 - * Para: Se ejecuta un número determinado de iteraciones

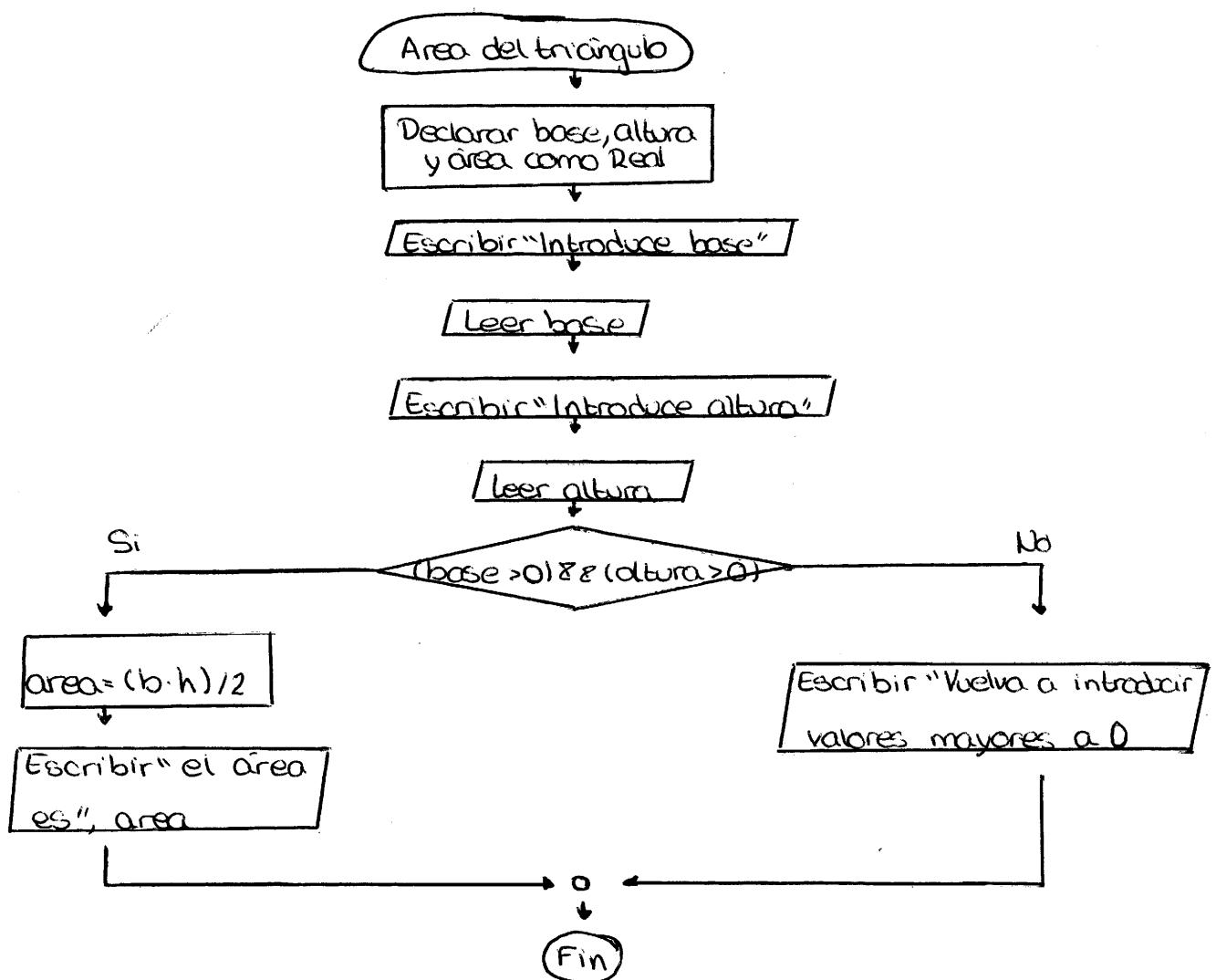
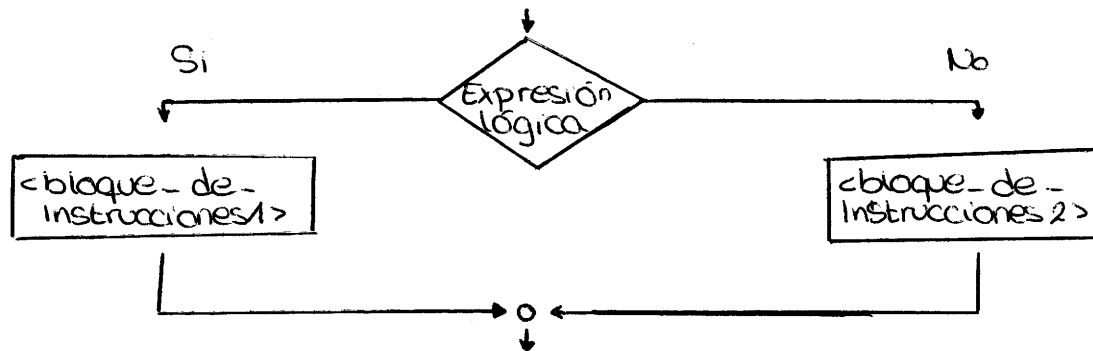
5.1 Condicional simple

Se evalúa si se cumple la condición, si en cierta se ejecuta el bloque de instrucción, si no, se continúa por el flujo normal del programa.



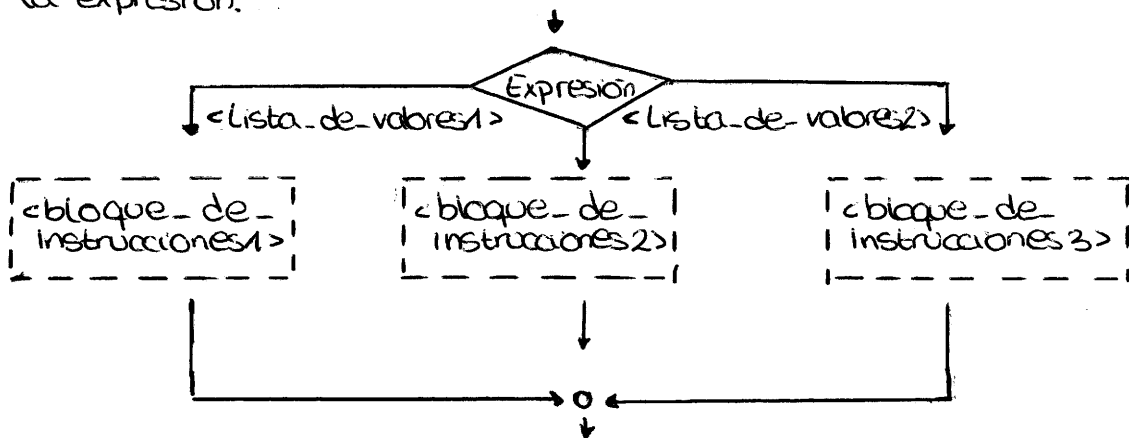
5.2 Condicional doble

Se evalúa si se cumple la condición si es cierta se ejecuta el bloque de instrucciones 1, si no, se ejecuta el bloque de instrucciones 2. Finalmente, se continúa por el flujo normal del programa.



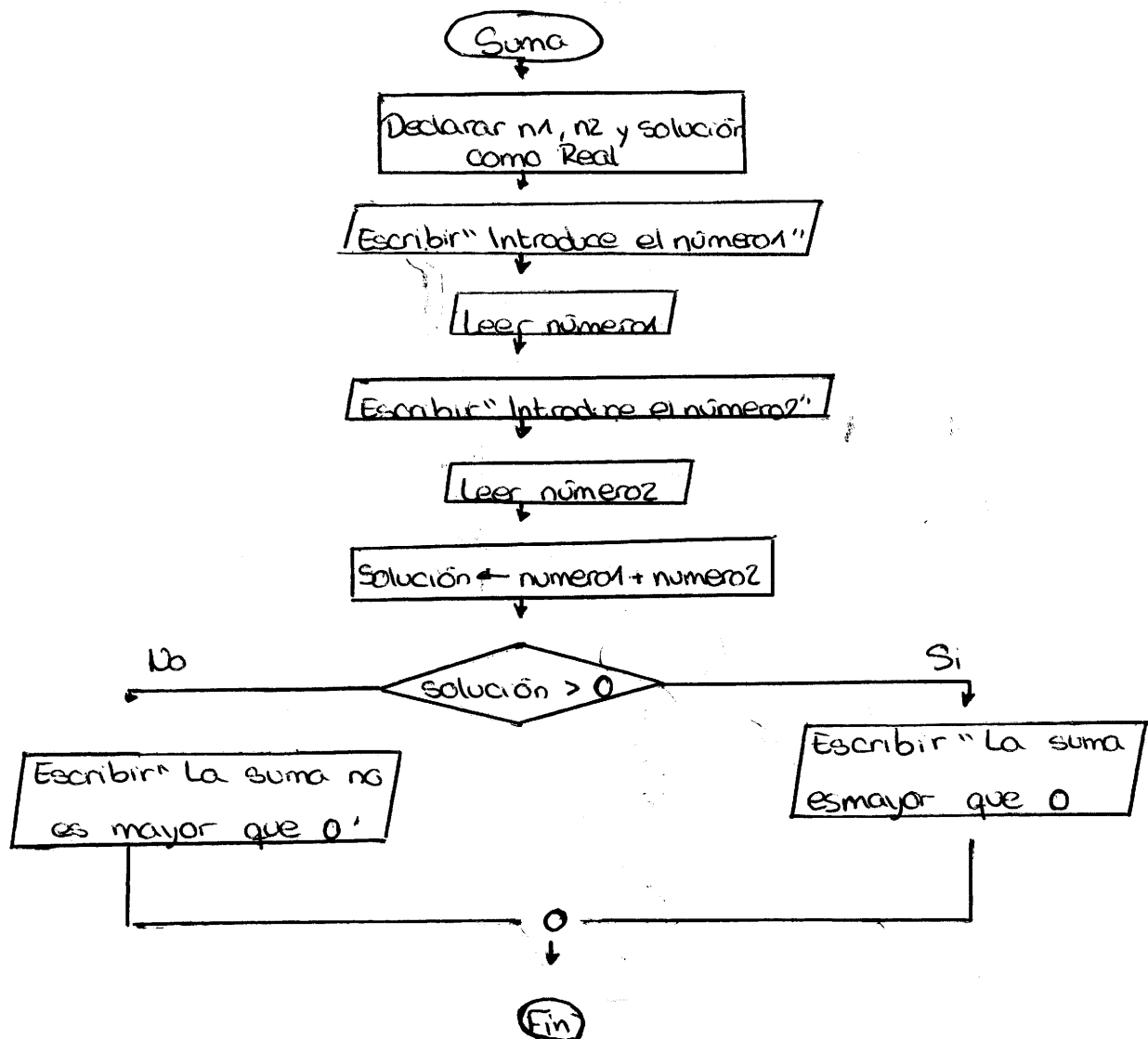
5.3 Condicional múltiple

Se ejecuta la rama de instrucciones que coincide con la evaluación de la expresión.

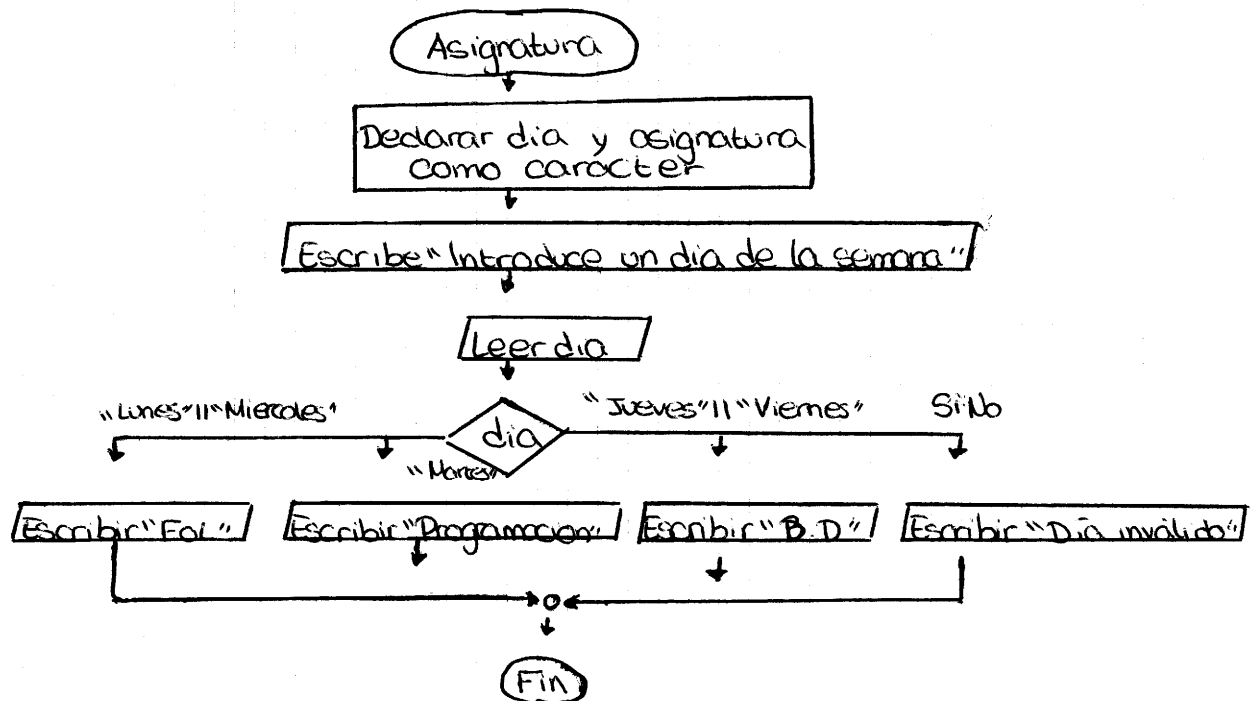


Ejercicio 11. Salario semanal 2. Ampliaremos el ejercicio anterior para considerar los horas extras. Las primeras 40h se pagan a 12 euros. Las siguientes se pagan a 16 euros.

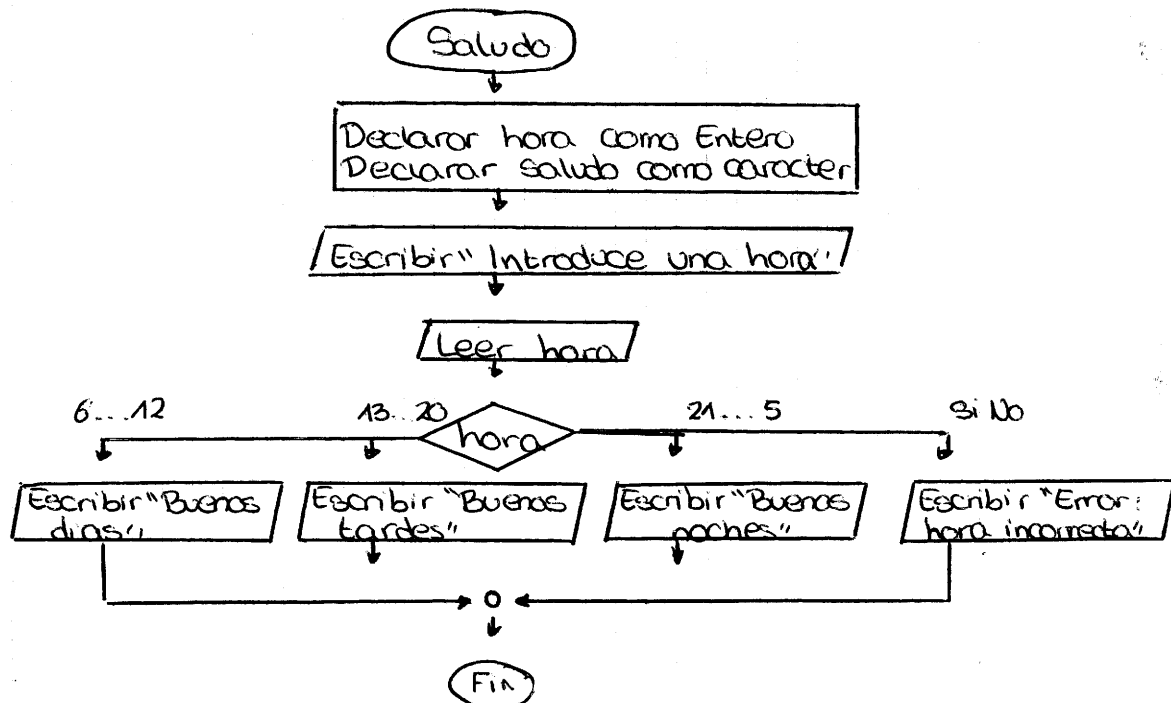
Ejercicio 8. Suma dos números. Escribe un programa que pida dos números y calcule la suma de ellos. Mostrará por pantalla "La suma es mayor que 0" o "La suma es menor que 0" en función del resultado.



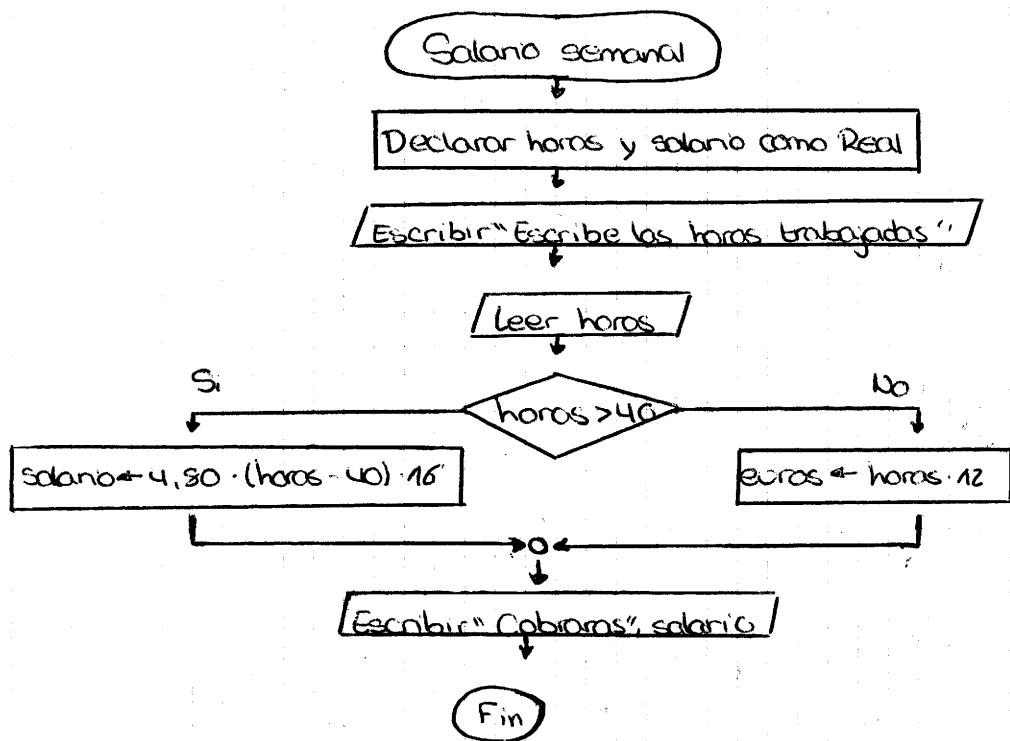
Ejercicio 9. Asignatura. Escribe un programa que pide el día de la semana y muestre que asignatura toca a primera hora.



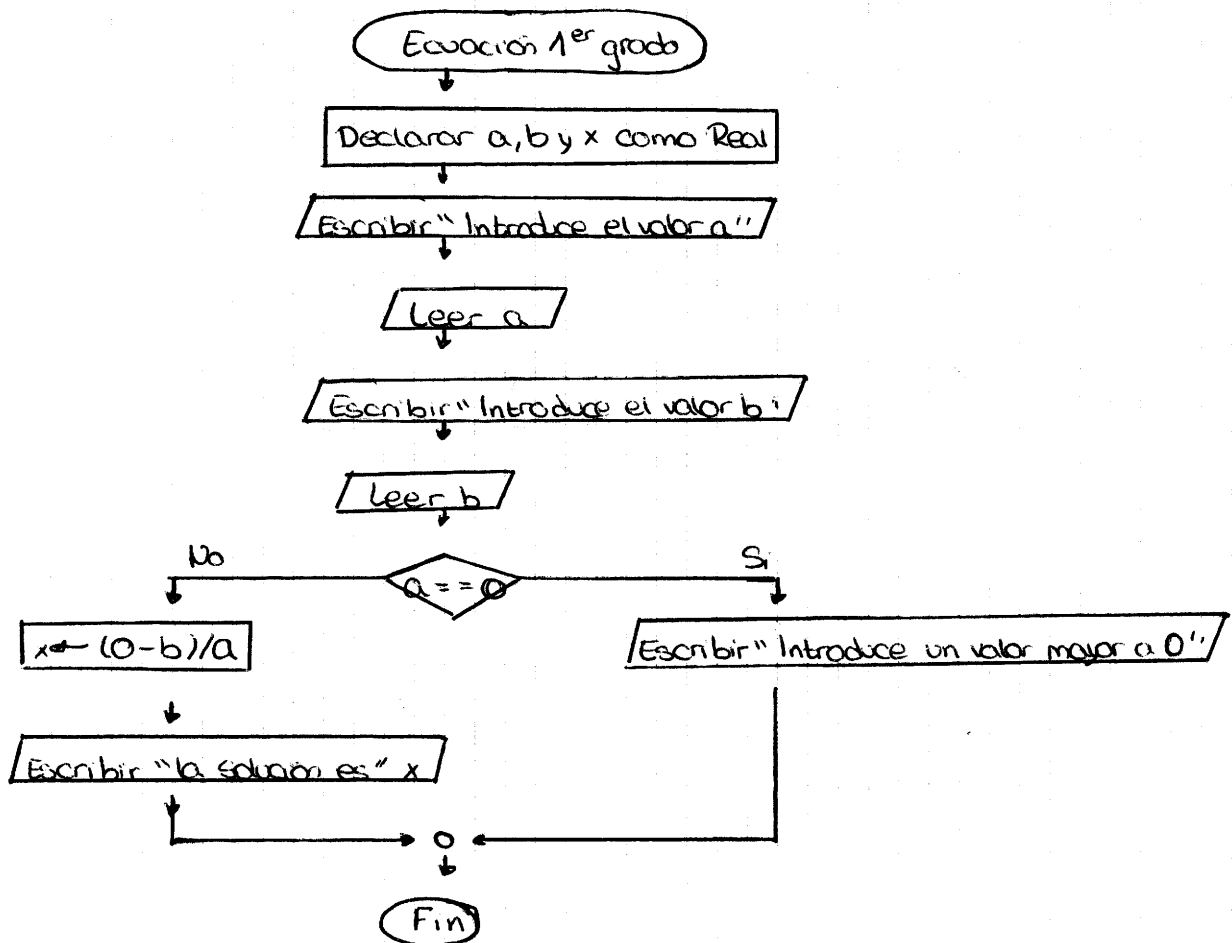
Ejercicio 10 Saludo. Escribe un programa que pida una hora (0-23) y muestre "Buenos días", "Buenas tardes", "Buenas noches" según la hora. Se utilizarán los tramos 6 a 12, 13 a 20 y 21 a 5 respectivamente



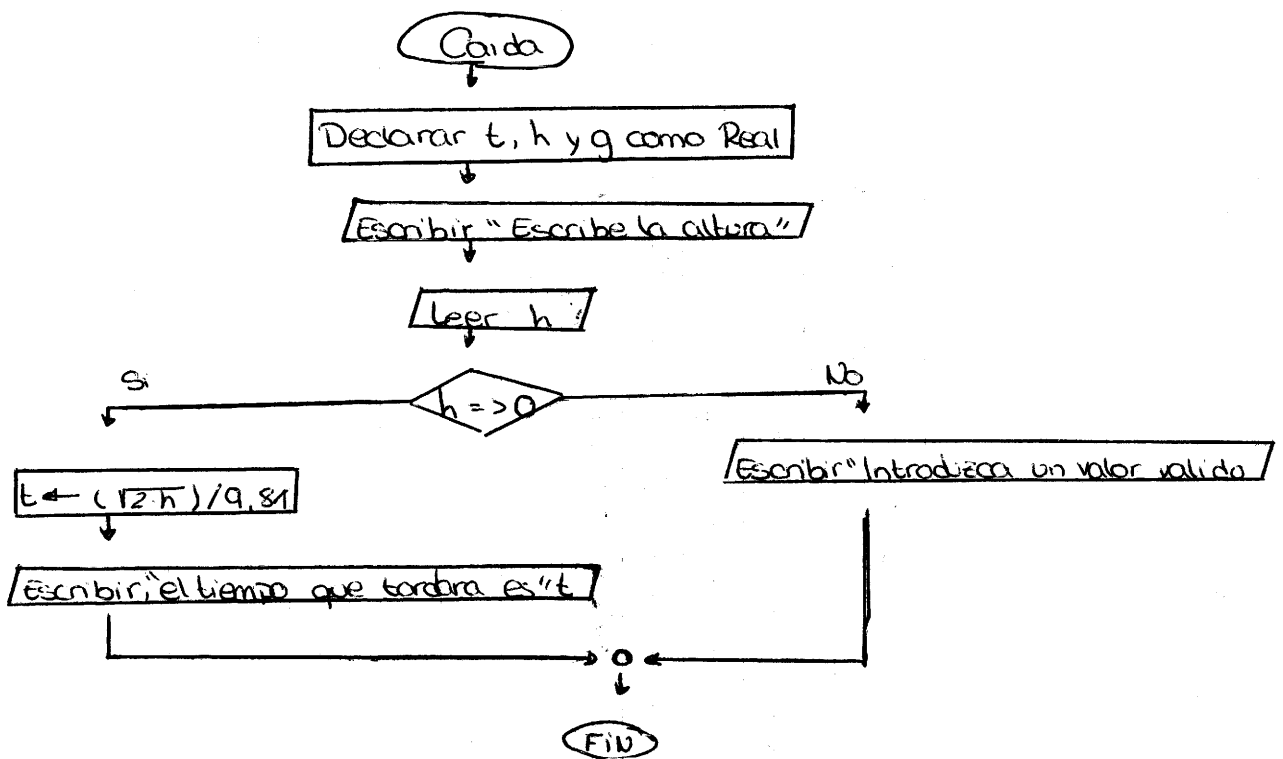
Ejercicio 11. Salario Semanal 2. Ampliaremos el ejercicio anterior para considerar las horas extras. Las primeras 40h se pagan a 12 euros. Las siguientes pagan a 16 euros.



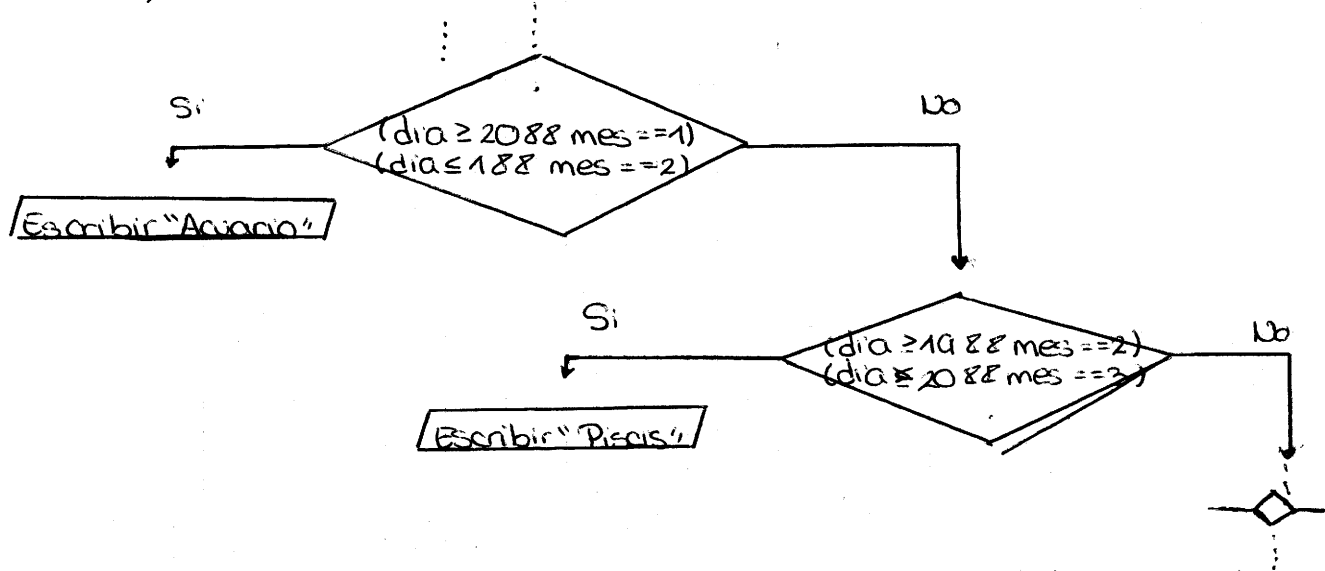
Ejercicio 12. Ecuación primer grado. Escribe un programa que resuelva una ecuación de primer grado $(ax+b)=0$



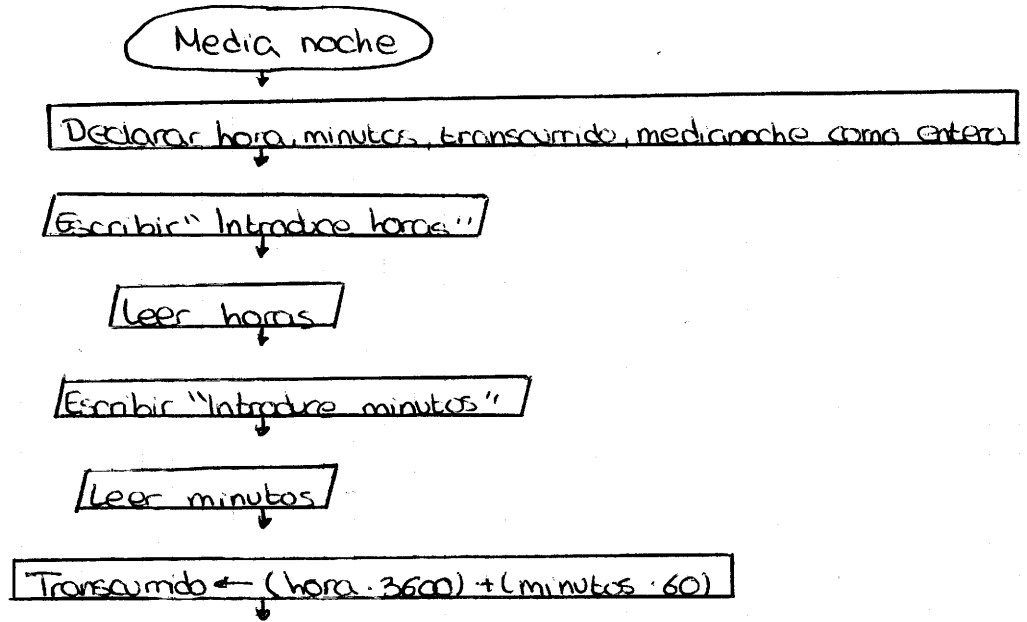
Ejercicio 13 Caída. Escribe un programa que calcule el tiempo que tardará en caer un objeto desde una altura (h). Aplica la fórmula $t = \sqrt{\frac{2h}{g}}$ siendo $g = 9,81 \text{ m/s}^2$.

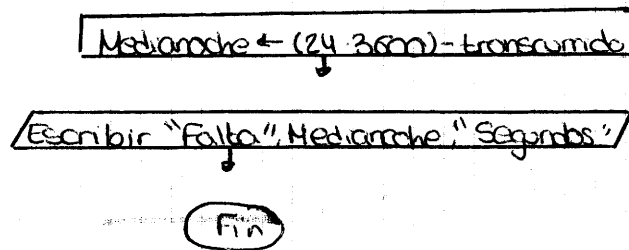


Ejercicio 14. Horóscopo. Escribe un programa que nos muestre el horóscopo a partir del día y mes de nacimiento (incompleto)



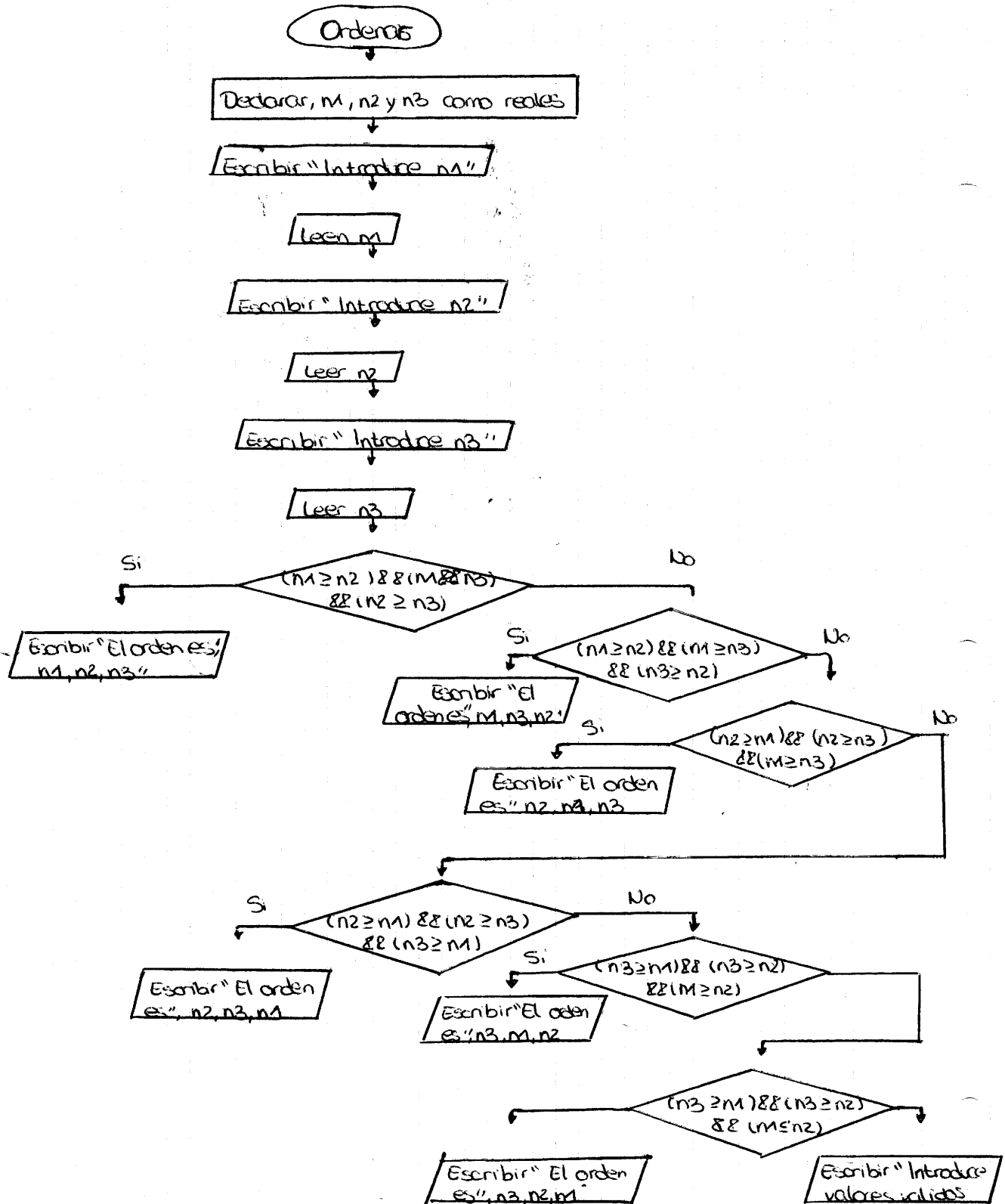
Ejercicio 15. Medianoche. Escribe un programa que dada una hora determinada (horas y minutos), calcule el número de segundos que faltan para llegar a media noche.



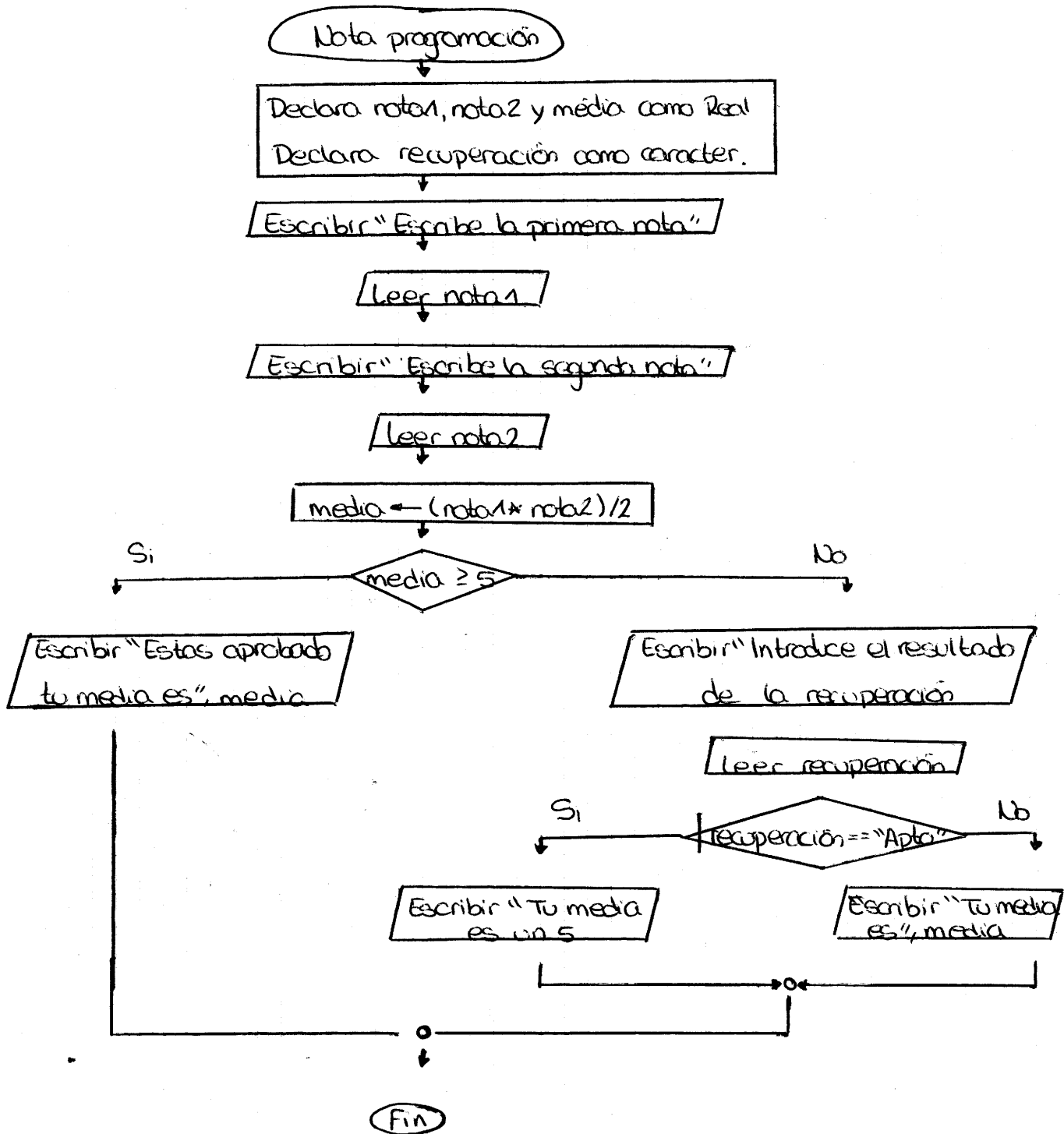


5.5 Condicionales anidados

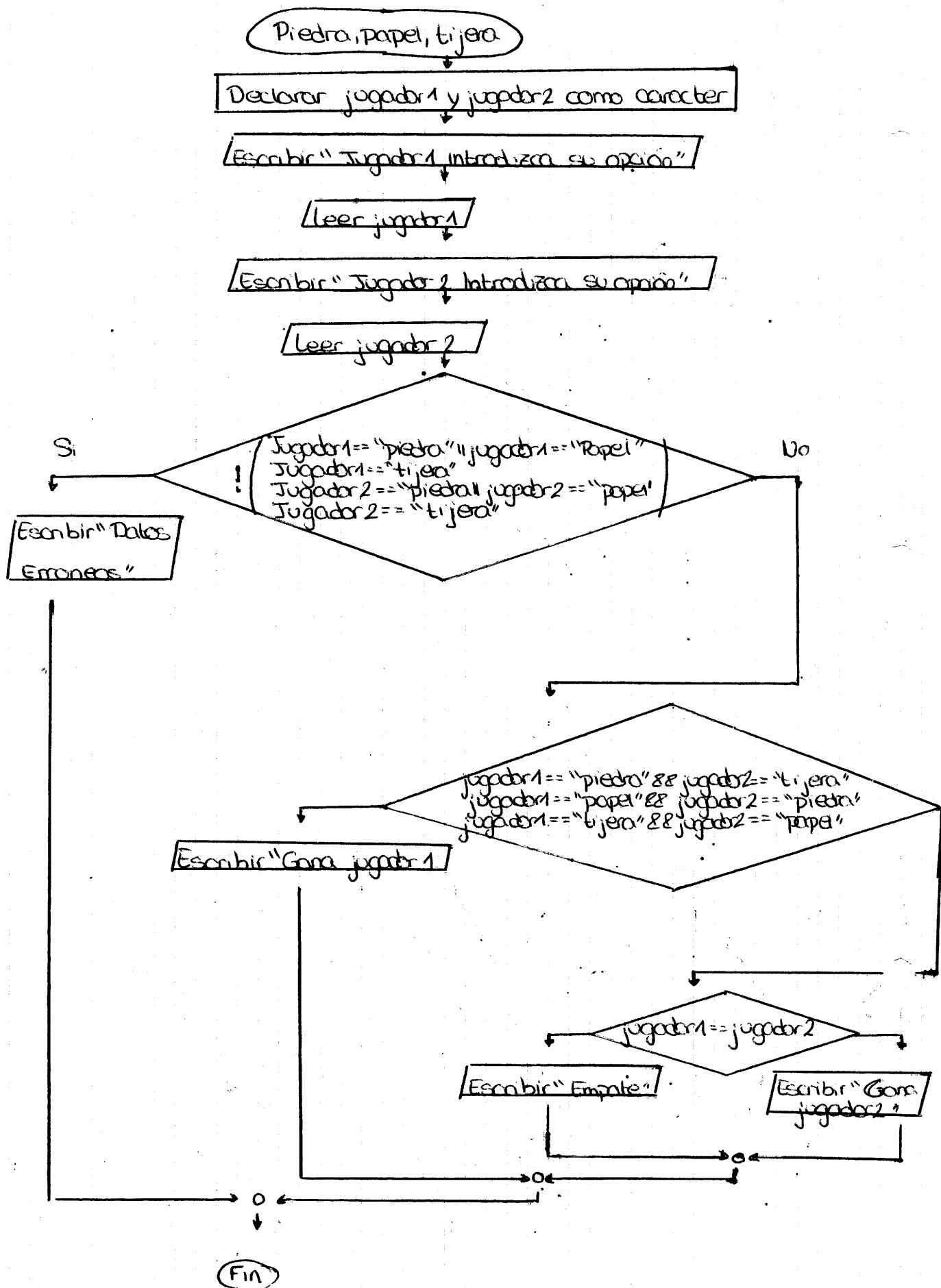
Ejercicio 16. Ordenar 3. Escribe un programa que orden 3 números enteros introducidos por teclado.



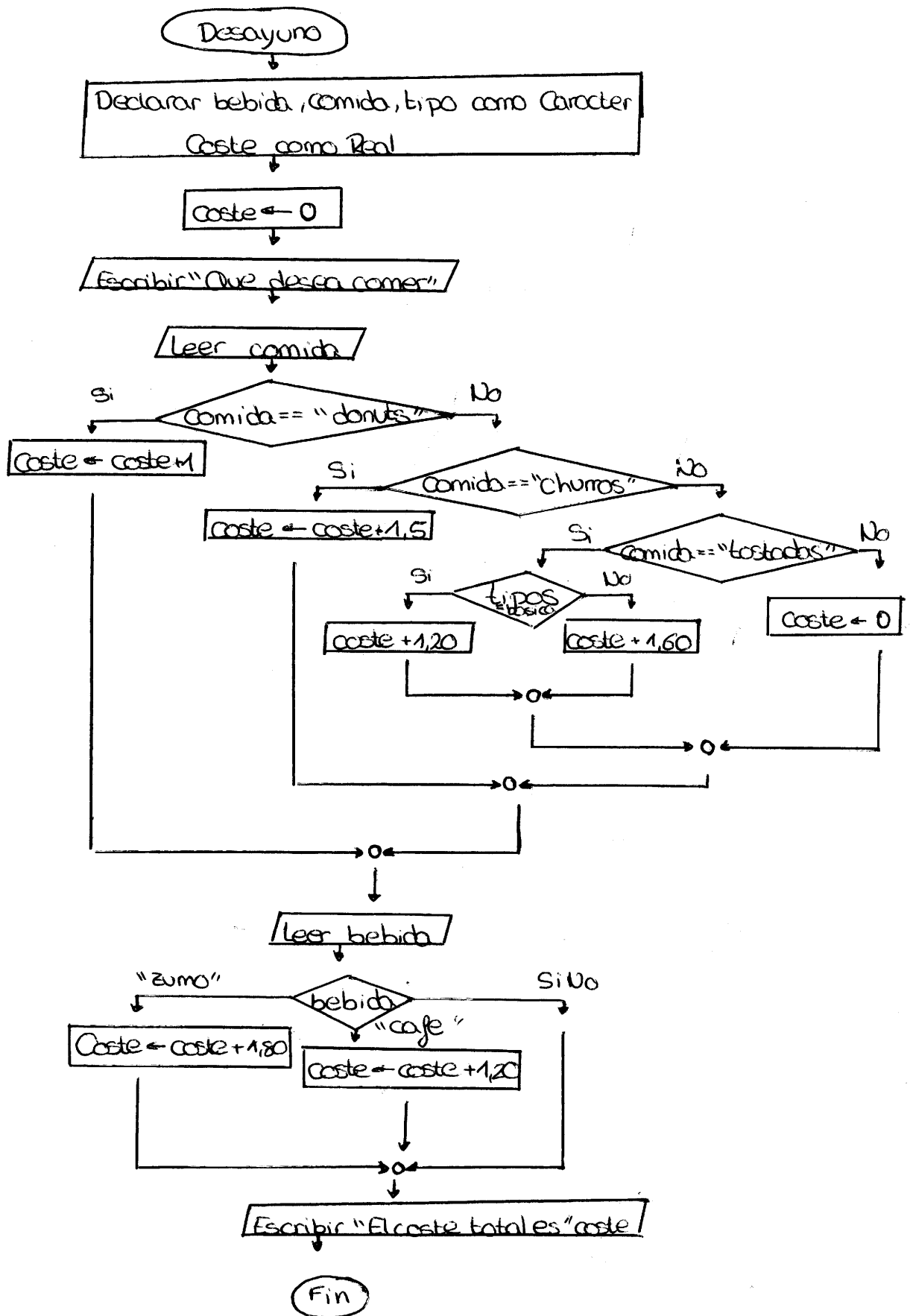
Ejercicio 17. Nota programación: Escribe un programa que pida la nota de dos exámenes de programación. Si la media es mayor o igual a 5, el alumno estará aprobado y se mostrará la media. Si no, se preguntará al usuario ¿Cual ha sido el resultado de la recuperación? (Apto/No apto). Si el resultado es apto, la nota será 5, en caso contrario se mantendrá la media anterior.



Ejercicio 18. Piedra, papel, tijera: Escribe un programa que implementen este juego para dos usuarios. Si alguno introduce una opción incorrecta se mostrará un mensaje de error.

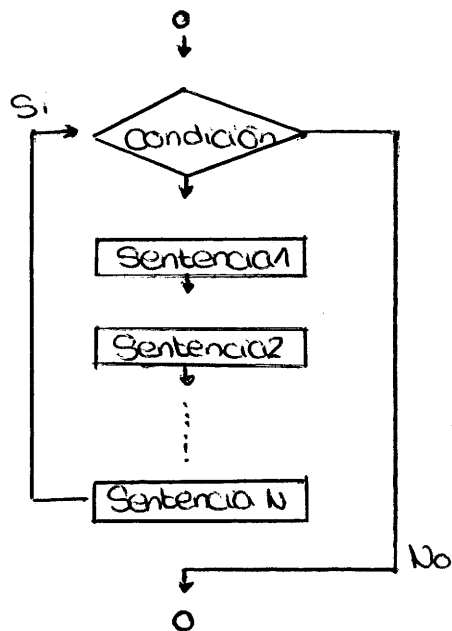


Ejercicio 19 Desayuno: Escribe un programa que calcule el precio de un desayuno. Primero pregunta al usuario que ha tomado para comer: Los churros valen 1,50€, donut 1€ y en caso de tomar tostada debe preguntar si es básica (1,20€) o especial (1,60€). Por último preguntará la bebida, zumo (1,80€) o café (1,20€).

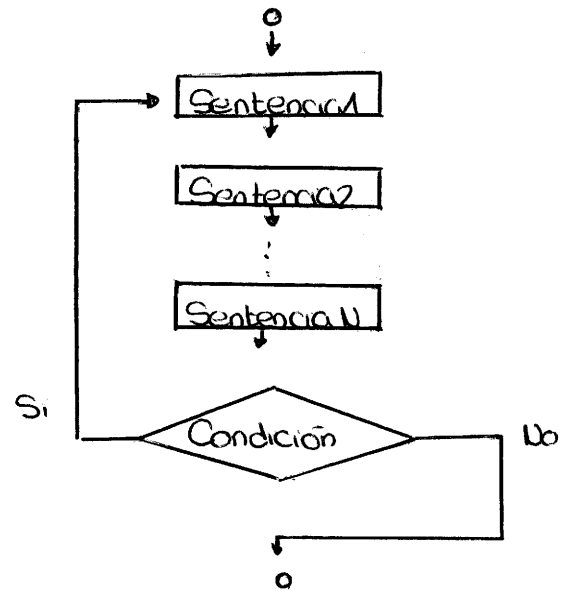


5.6 Estructuras cíclicas

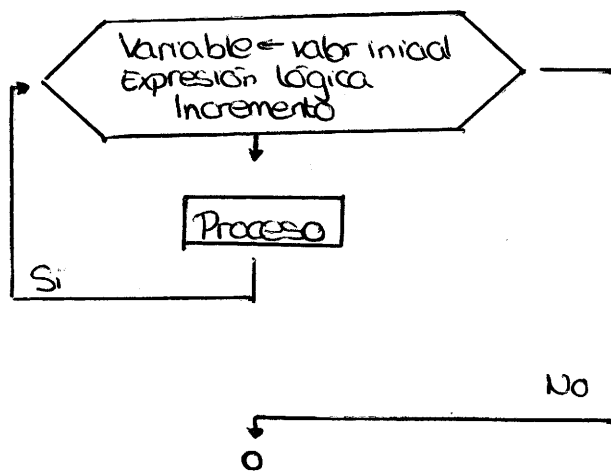
Mientras (while)



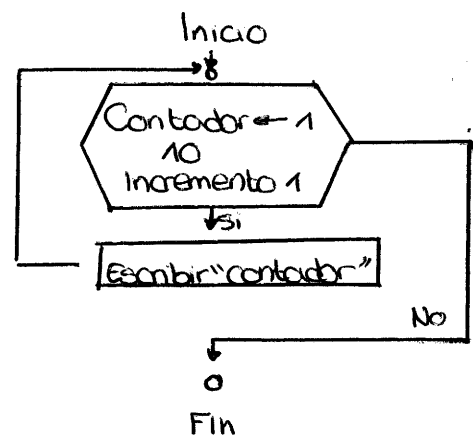
Hace... Mientras (Do while)



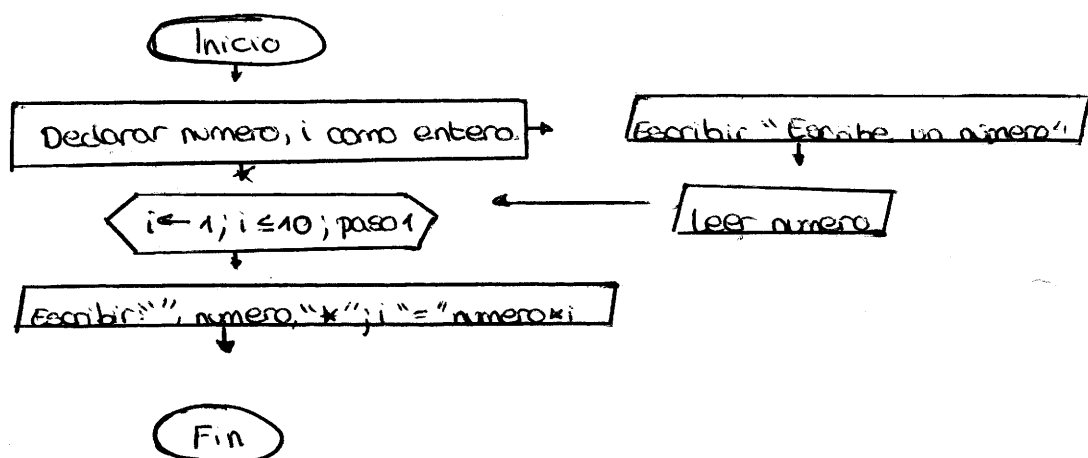
Para (For)

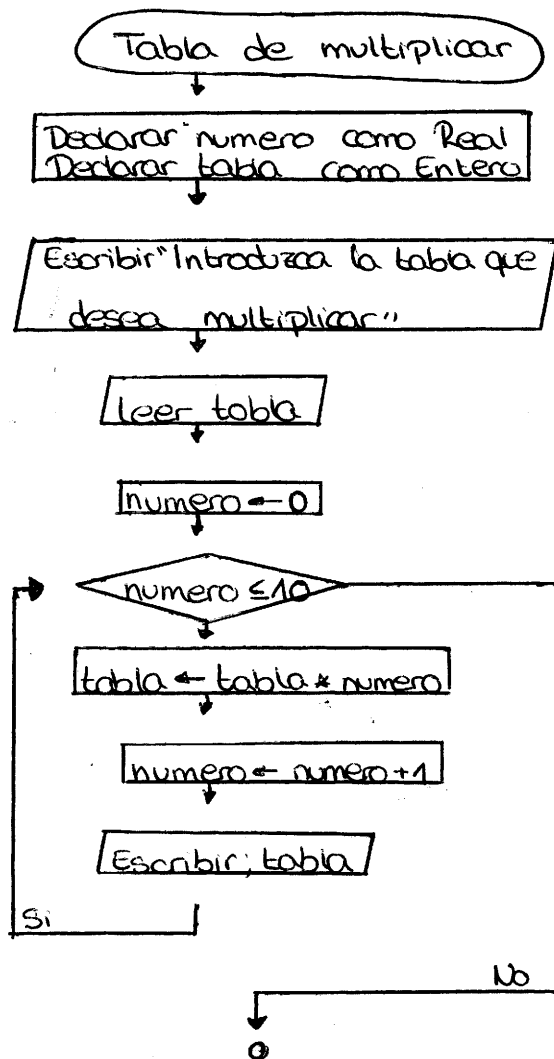


Ejemplo

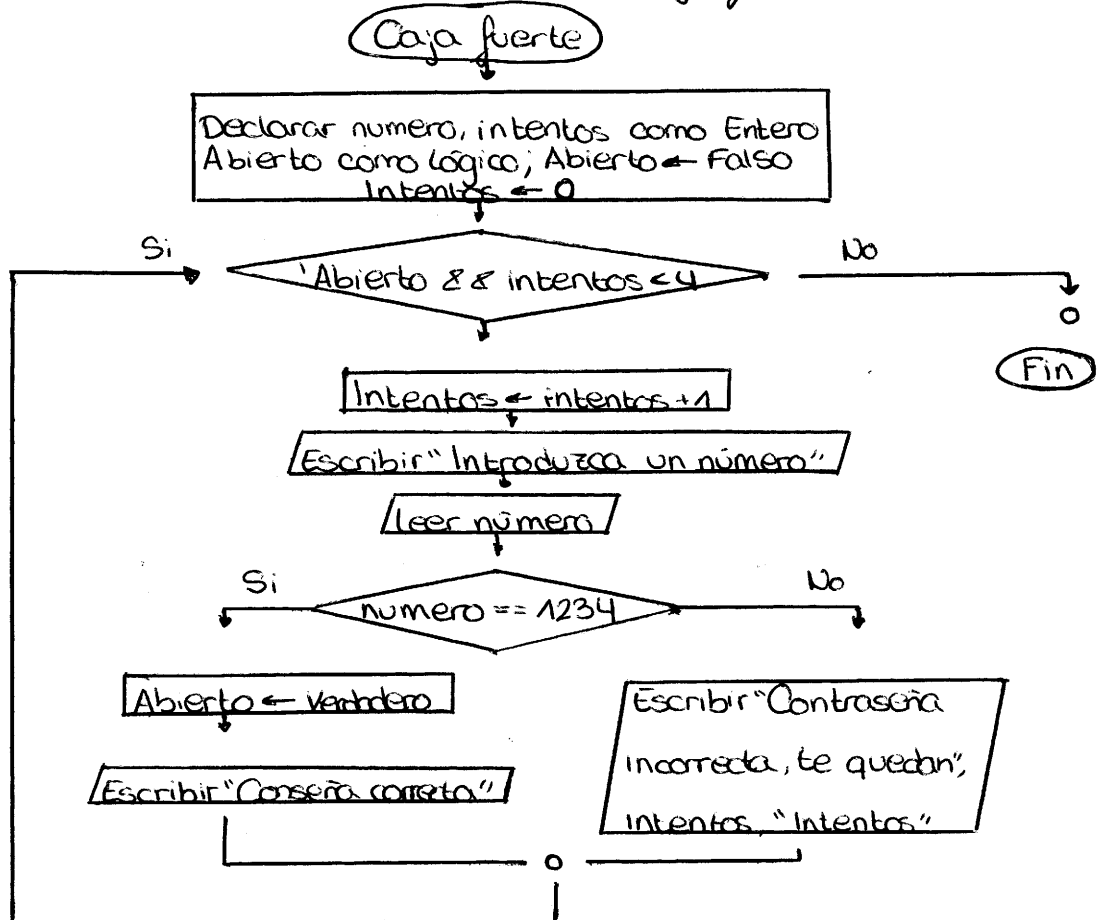


Ejercicio 20. Tabla de multiplicar. Escribe un programa que muestre la tabla de multiplicar de un número introducido por teclado.

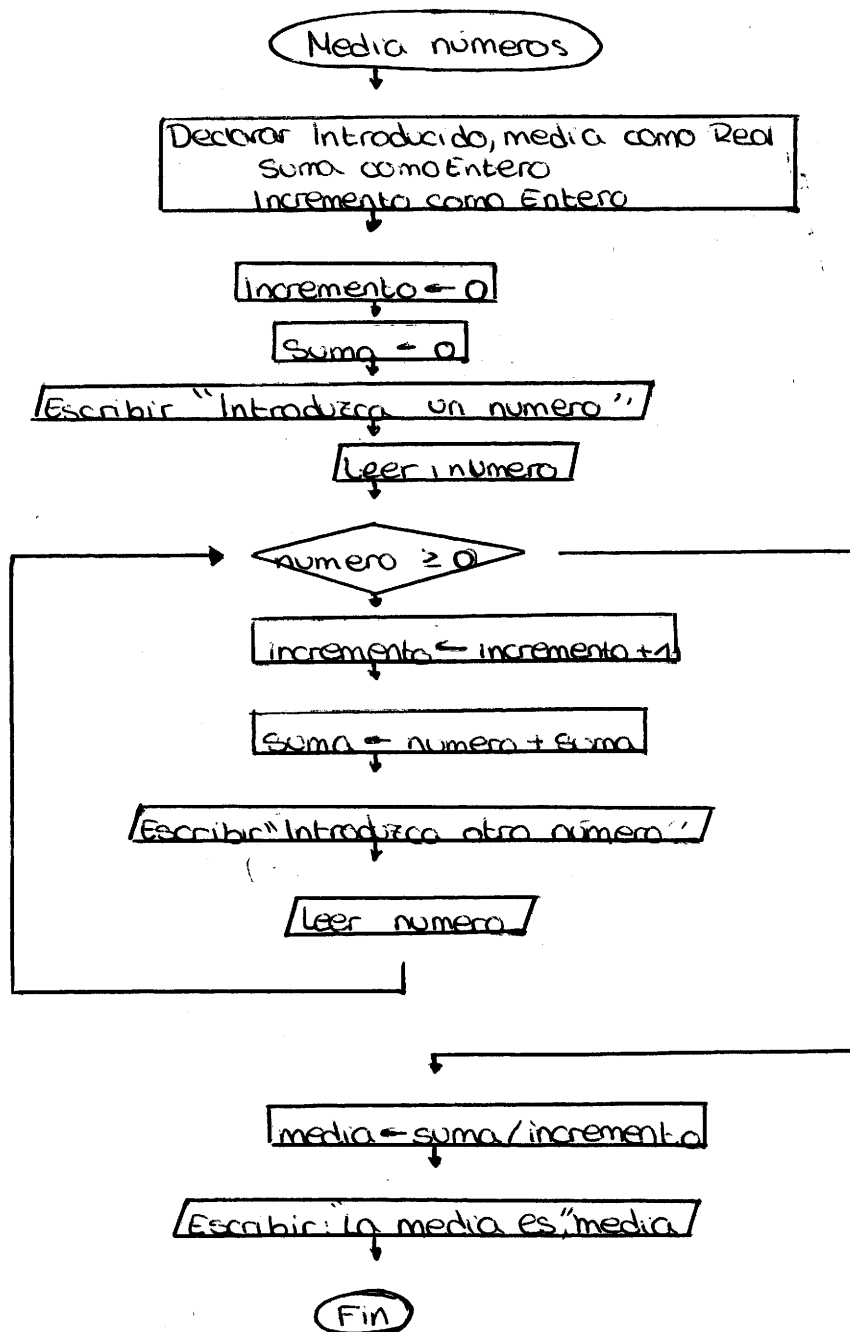




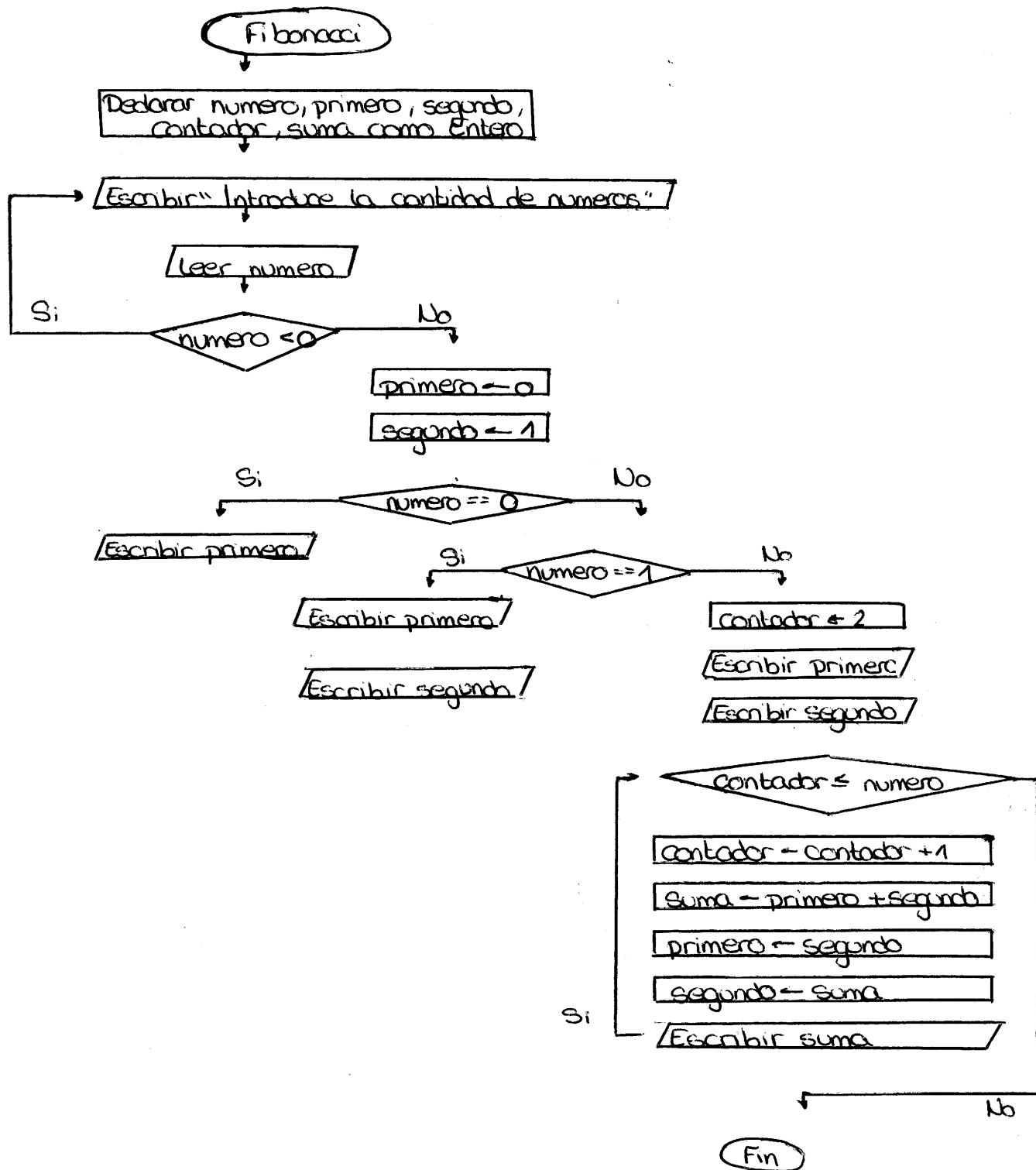
Ejercicio 21. Caja fuerte. Realiza un programa que pida un número de 4 cifras. Si no acertamos se mostrara el mensaje "lo siento esa no es la combinación" y si acertamos sen nos dira, "la caja fuerte se ha abierto". Hay 4 intentos.



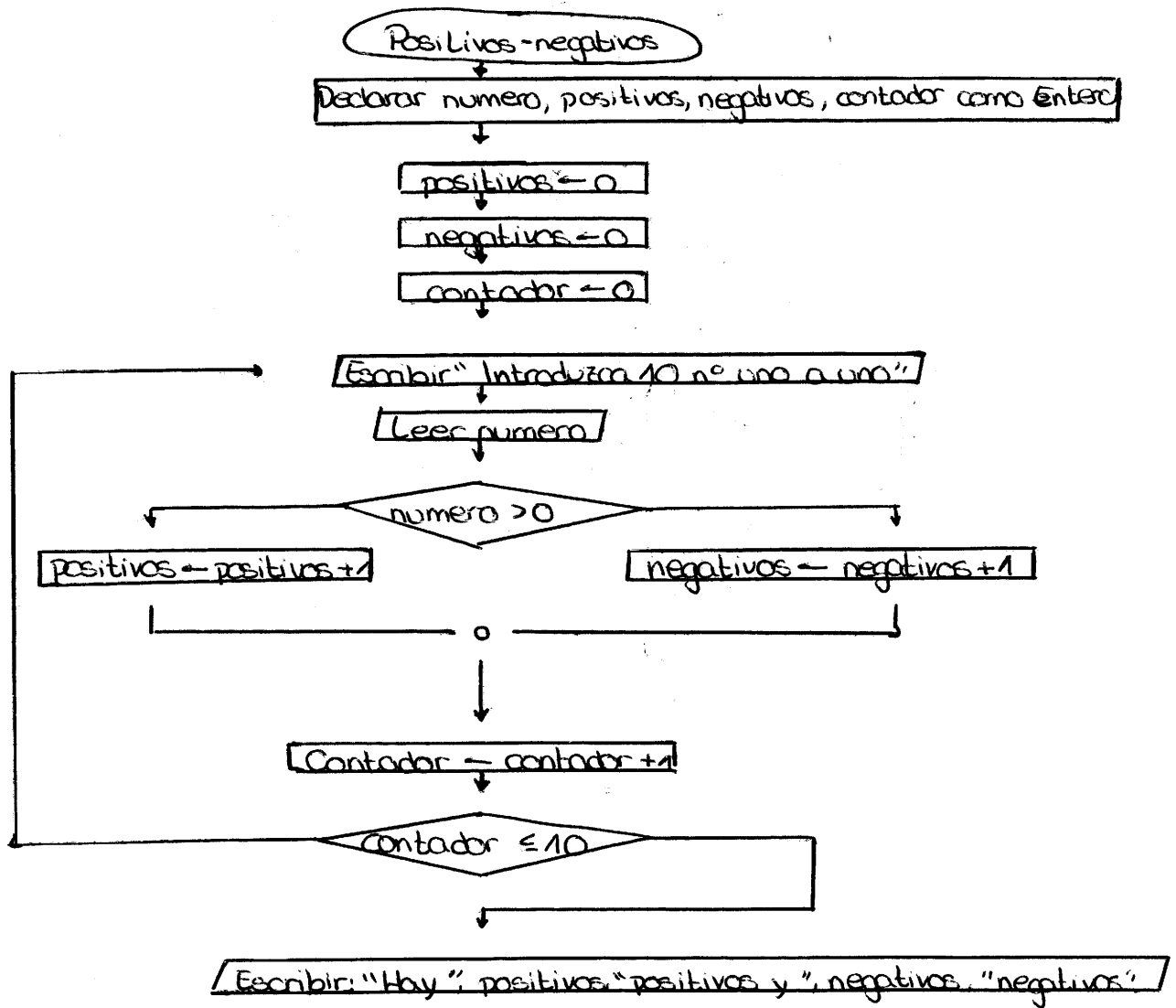
Ejercicio 22. Media números. Escribe un programa que calcule la media de una serie de números posibles. El usuario indicará que ha terminado de introducir datos cuando introduzca un número negativo.



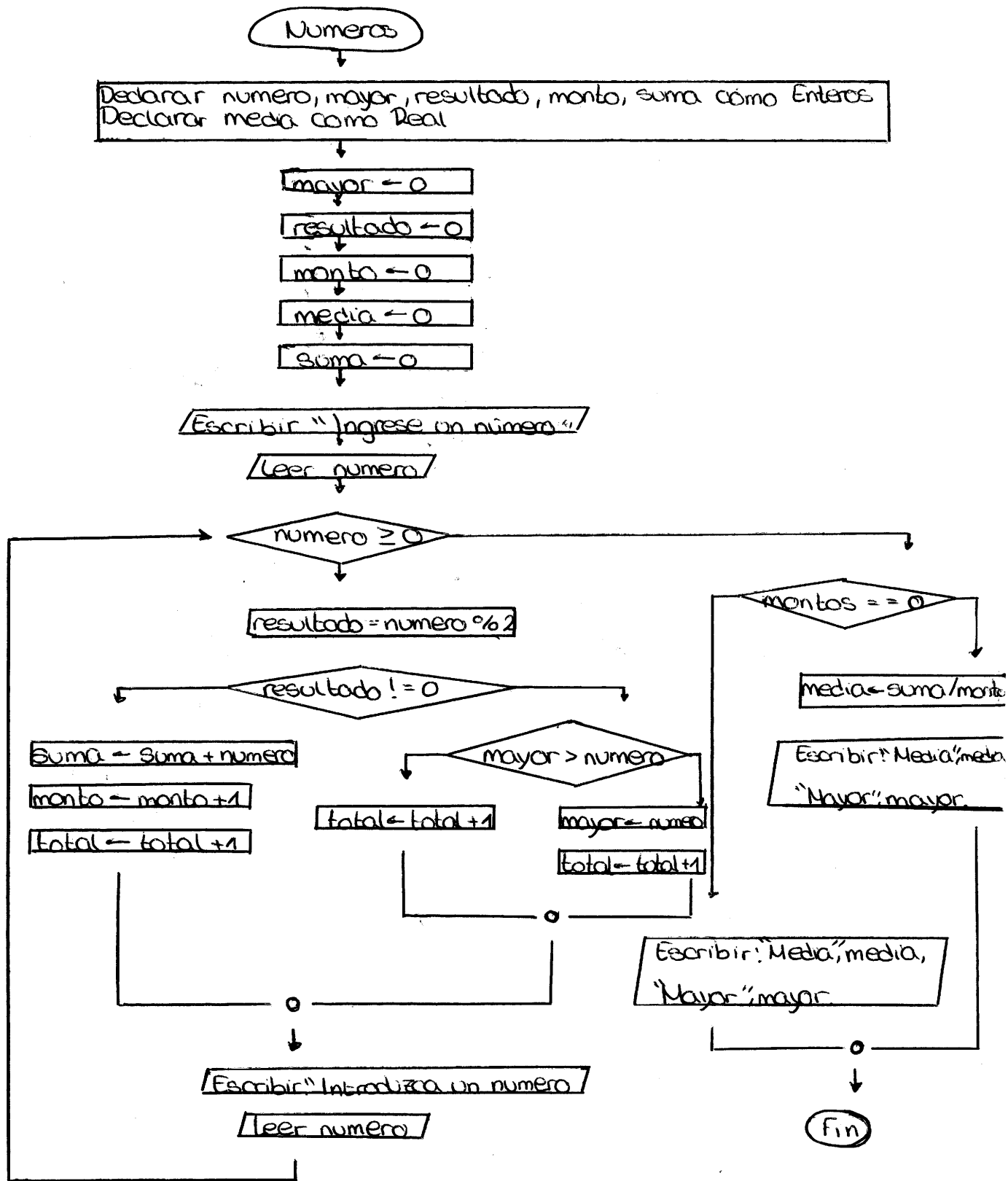
Ejercicio 23. Fibonacci. Escribe un programa que muestre los n primeros términos de la serie de fibonacci. El valor de n será introducido por teclado.



Ejercicio 24 Positivos - negativos: Escribe un programa que pida al usuario 10 números y muestre por pantalla cuántos son positivos y cuántos negativos.



Ejercicio 25. Numeros. Escribe un programa que pida números hasta que se introduzca un número negativo. Se mostrarán cuantos números se han introducido, la media de los impares y el mayor de los pares. El n° negativo solo se utilizará para salir.



6 Pseudocódigo

Es un código intermedio entre el lenguaje de programación y el lenguaje natural.
Evitas los detalles de sintaxis de un lenguaje concreto.

No se rige por normas estrictas.

No son tan visuales como los diagramas de flujo.

Son adecuados para una descripción formal de un algoritmo, independientemente del lenguaje.

Estructura de un algoritmo en pseudocódigo:

Algoritmo Area del triángulo

Definir base, altura como Real
Definir area como Real

Escribir "Introduce la base del triángulo".
Leer base
Escribir "Introduce la altura del triángulo".
Leer altura

$area \leftarrow (base * altura) / 2$
Escribir "El área es", area

Fin algoritmo.

Ejercicio 1 Media aritmética

Algoritmo Media aritmética

Definir a, b y media como Real

Escribir "Introduce la 1ª nota"
Leer a
Escribir "Introduce la 2ª nota"
Leer b

$media \leftarrow (a + b) / 2$
Escribir: "La media es", media

Fin algoritmo

Ejercicio 2 Salario

Algoritmo Salario

Definir horas como Entero
Definir salario como Real

Escribir "Introduce las horas trabajadas"
Leer horas

$Salario \leftarrow horas * 12$
Escribir: "Tu salario es", salario

Fin algoritmo

- Condicional simple y doble

(Simple)

Si (<expresión_lógica>) entonces
 <bloque_de_instrucciones>
FinSi

(doble)

Si (<expresión_lógica>) entonces
 <bloque_de_instrucciones>
SiNo
 <bloque_de_instrucciones>
FinSi

- Condicional anidado

Si (<expresión_lógica>) entonces
 <bloque_de_instrucciones_1>
SiNoSi (<expresión_lógica>) entonces
 <bloque_de_instrucciones_2>
SiNoSi (<expresión_lógica>) entonces
 <bloque_de_instrucciones_3>
.....
SiNo
 <bloque_de_instrucciones_N>
FinSi

- Condicional múltiple

SegunSea (<expresión>)
 <lista_de_valores_1>: <bloque_de_instrucciones_1>
 <lista_de_valores_2>: <bloque_de_instrucciones_2>
.....
 <lista_de_valores_N>: <bloque_de_instrucciones_N>
 [SiNo: <bloque_de_instrucciones_{N+1}>]
FinSegunSea

Ejemplo algoritmo

Algoritmo Saludo

Definir hora como Entero

Escribir "Introduce una hora"

Leer hora

Si $hora \geq 6$ y $hora \leq 12$ entonces

Escribir "Buenos días"

Si No Si $hora \geq 13$ y $hora \leq 20$ entonces

Escribir "Buenos tardes"

Si No Si $hora \geq 21$ y $hora \leq 24$ || $hora \geq 0$ y $hora \leq 5$

Escribir "Buenas noches"

Si No

Escribir "hora no valida"

Fin Si

Fin Algoritmo

Ejercicio 15

Algoritmo Medianoche

Definir horas, minutos, transcurrido, medianoche como Entero

Escribir "Introduce las horas"

Leer horas

Escribir "Introduce los minutos"

Leer minutos

Si $(horas \geq 0$ y $horas \geq 23)$ y $(minutos \geq 0$ y $minutos \leq 59)$

$minutos \leftarrow 60 - minutos$

$horas \leftarrow 23 - horas$

$segundos \leftarrow (minutos \cdot 60) + (horas \cdot 3600)$

Escribir: "Faltan", segundos, "segundos para medianoche."

Si No

Escribir "Datos no validos"

Fin Si

Fin Algoritmo

Ejercicio 18.

Algoritmo piedra, papel, tijera

Definir jugador1, jugador2 como caracter

Escribir "Jugador 1, introduzca su opción"
Leer jugador1

Escribir "Jugador 2, introduzca su opción"
Leer jugador2

Si (jugador1 == "piedra" || jugador1 == "papel" || jugador1 == "tijera" &&
jugador2 == "piedra" || jugador2 == "papel" || jugador2 == "tijera")

Escribir "Datos erroneos"

Si No Si (jugador1 == "piedra" && jugador2 == "tijera")
jugador1 == "papel" && jugador2 == "piedra"
jugador1 == "tijera" && jugador2 == "papel")

Escribir "Gana el jugador1"

Si No Si (jugador1 == jugador2)

Escribir "Empate"

Si No

Escribe "Gana jugador2"

FinSi

FinAlgoritmo

Ejercicio 19

Algoritmo Desayuno

Definir bebida, comida, tipo como Caracter
Definir coste como Real.

Coste \leftarrow 0

Escribir "Que comida ha elegido"
Leer comida

Si comida == "donuts" entonces

 coste \leftarrow coste + 1

Si No Si comida == "churros" entonces

 coste \leftarrow coste + 1,5

Si No Si comida == "tostadas" entonces

 Si tipo == "básica"

 coste \leftarrow coste + 1,20

 Si No

 coste \leftarrow coste + 1,60

Si No

 coste \leftarrow 0

Fin Si

Escribir "Que bebida ha elegido"
Leer bebida

Segun sea bebida:

 "zumos": coste \leftarrow coste + 1,80

 "cafe": coste \leftarrow coste + 1,20

Si No

 Escribir "El coste total es ", coste

Fin Segun Sea

Fin Algoritmo

- Bucle mientras

Mientras ($\langle \text{expresión_lógica} \rangle$) entonces
 ($\langle \text{bloque_de_instrucciones} \rangle$)

Fin Mientras

Ejemplo

contador \leftarrow 1

Mientras (contador \leq 10)

 Escribir contador

 contador \leftarrow contador + 1

Fin Mientras

- Bucle Hacer mientras

Hacer

 ($\langle \text{bloque_de_instrucciones} \rangle$)

Mientras ($\langle \text{expresión_lógica} \rangle$)

Ejemplo

contador \leftarrow 1

Hacer

 Escribir contador

 contador \leftarrow contador + 1

Mientras (contador \leq 10)

- Bucle para

Para $\langle \text{variable} \rangle$, $\leftarrow \langle \text{valor_inicial} \rangle$, $\langle \text{condición} \rangle$, $\langle \text{incremento} \rangle$, $\langle \text{valor_incremento} \rangle$, hacer
 $\langle \text{bloque_de_instrucciones} \rangle$

Fin Para

Ejemplo

Para contador, contador \leftarrow 1, contador \leq 10, incremento 1 hacer

 Escribir contador

Fin Para

Ejercicio 21

Algoritmo caja fuerte

Declarar numero, intentos como Entero
Abierto como lógico

Abierto \leftarrow Falso
Intentos \leftarrow 0

Mientras !(Abierto & intentos < 4)

Intentos \leftarrow intentos + 1

Escribir "Introduzca el número secreto"
Leer numero

Si numero == "1234"

Abierto \leftarrow verdadero

Escribir "Caja fuerte abierta"

Si No

Escribir "Contraseña incorrecta"

FinSi

FinMientras

FinAlgoritmo

Ejercicio 22

Algoritmo MediaNumeros

Ejercicio 23

Algoritmo Fibonacci

Declarar numero, primero, segundo, incremento, suma como Entero

Hacer

Escribir "Introduce la cantidad de números"
Leer numero

Mientras (numero < 0)

Primero \leftarrow 0

Segundo \leftarrow 1

Si numero == 0

Escribir primero

Si No Si numero == 1

Escribir primero

Escribir segundo

Si No incremento \leq 2

Escribir primero

Escribir segundo

Mientras (incremento \leq numero)

incremento \leftarrow incremento + 1

suma \leftarrow primero + segundo

primero \leftarrow segundo

segundo \leftarrow suma

Escribir suma

Fin Mientras

Fin Si

Fin Algoritmo

Falta inicializar la
variable incremento
a 0

Ejercicio 24

Algoritmo positivos-negativos.

Declarar numero, positivo, negativo, contador como Entero

positivo $\leftarrow 0$

negativo $\leftarrow 0$

contador $\leftarrow 0$

Hacer

 Escribir "Introduzca 10 números uno a uno"
 Leer numero

 Si numero > 0

 positivo \leftarrow positivo + 1

 Si No Si numero < 0

 negativo \leftarrow negativo + 1

 Si No

 Escribir "numero no valido"

 Fin Si

 Contador \leftarrow contador + 1

Mientras contador ≤ 10

 Escribir " Hay ", positivo, " número positivos y ", negativo, " negativos".

Fin Algoritmo

Ejercicio 25

Algoritmo Numeros

Declarar numero, par mayor, contador, impar, media como Entero

par mayor $\leftarrow 0$

contador $\leftarrow 0$

impar $\leftarrow 0$

Hacer

Escribir "Introduzca un numero, negativo si quiere acabar"
Leer numero

Si (numero > 0)

contador \leftarrow contador + 1

Si (numero % 2 != 0)

media \leftarrow media + numero

impar \leftarrow impar + 1

Si No Si (numero \geq par mayor)

par mayor \leftarrow numero

Fin Si

Fin Si

Mientras (numero > 0)

media \leftarrow media / impar

Escribir media

Escribir par mayor

Escribir contador

Fin Algoritmo

Relación de ejercicios

1. Mostrar los numeros del 1 al 100