

# CAPÍTULO 1

**EJEMPLO 2:** Programa que lee dos números y muestra el mayor de ellos en pantalla, si son iguales deberá mostrar un mensaje indicándolo.

Se utiliza la estructura condicional **Si** <condición> **Entonces** <instrucciones> **Si no** <instrucciones> **Fin si**.

Además, se muestra una estructura condicional dentro de otra.

PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<p><b>Inicio</b></p> <p>Leer A, B</p> <p><b>Si</b> <math>A &gt; B</math> <b>entonces</b></p> <p>    Visualizar “El mayor es: “ A</p> <p><b>Sino</b></p> <p>    <b>Si</b> <math>A = B</math> <b>entonces</b></p> <p>        Visualizar “Son iguales”</p> <p>    <b>Sino</b></p> <p>        Visualizar “El mayor es: “ B</p> <p>    <b>Fin Si</b></p> <p><b>FinSi</b></p> <p><b>Fin</b></p>	<pre> graph TD     INICIO([INICIO]) --&gt; Input[/A, B/]     Input --&gt; Cond1{A &gt; B}     Cond1 -- Si --&gt; OutA([El mayor es: A])     Cond1 -- NO --&gt; Cond2{A = B}     Cond2 -- Si --&gt; OutEq([Son iguales])     Cond2 -- NO --&gt; OutB([El mayor es: B])     OutA --&gt; Join(( ))     OutEq --&gt; Join     OutB --&gt; Join     Join --&gt; FIN([FIN])     </pre>

**EJEMPLO 3:** Programa que lee dos números en un proceso repetitivo. Este proceso terminará cuando los números leídos sean iguales.

Se utiliza la estructura repetitiva **Repetir** <instrucción> **Hasta que** <condición>.

PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<p><b>Inicio</b></p> <p><b>Repetir</b></p> <p>Visualizar “Escribe dos números”</p> <p>Leer A, B</p> <p><b>Hasta</b> A=B</p> <p><b>Fin</b></p>	<pre>graph TD; Inicio([Inicio]) --&gt; Escribe{{Escribe dos números}}; Escribe --&gt; AB[/A, B/]; AB --&gt; ABEq{A= B}; ABEq -- Si --&gt; FIN([FIN]); ABEq -- NO --&gt; Escribe;</pre>

**EJEMPLO 4:** Programa que lee diez números en un proceso repetitivo y muestra la suma. Es necesario declarar variables para contar los números que se van leyendo y para ir guardando la suma. Se deben inicializar a 0. Se utiliza la estructura repetitiva **Repetir** <instrucción> **Hasta que** <condición>.

PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<p><b>Inicio</b></p> <p>Declarar Cuenta=0</p> <p>Declarar Suma=0</p> <p><b>Repetir</b></p> <p>Visualizar “Escribe un número”</p> <p>Leer A</p> <p>Cuenta = Cuenta + 1</p> <p>Suma = Suma + A</p> <p><b>Hasta</b> cuenta = 10</p> <p>Visualizar “La suma es: ” Suma</p> <p><b>Fin</b></p>	<pre> graph TD     INICIO([INICIO]) --&gt; Init[Cuenta = 0&lt;br/&gt;Suma = 0]     Init --&gt; Input([Escribe un número])     Input --&gt; Read[/A/]     Read --&gt; Process[Cuenta = Cuenta + 1&lt;br/&gt;Suma = Suma + A]     Process --&gt; Decision{Cuenta = 10}     Decision -- SI --&gt; Output([La suma es: "&lt;br/&gt;Suma])     Decision -- NO --&gt; Input     Output --&gt; FIN([FIN])     </pre>

**EJEMPLO 5:** Programa que lee registros de un fichero secuencial. Cada registro contiene información de un alumno: Nombre, Curso, Nota.

El programa debe mostrar por cada registro leído el Nombre, el Curso y la Nota.

Al final del proceso de lectura debe mostrar la nota media. Esta se calcula sumando las notas de todos los alumnos y dividiéndola por el número de alumnos que hay.

Se utiliza la estructura repetitiva **Mientras** < condición> **Hacer** < instrucciones> **Fin mientras**.

Se declaran tres variables, una para contar alumnos, otra para sumar notas y una tercera para guardar la nota media.

PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<p><b>Inicio</b></p> <p>Declarar Cuenta=0</p> <p>Declarar Suma=0</p> <p>Declarar Media</p> <p>Abrir Fichero Notas</p> <p>Leer Registro (Nombre, Curso, Nota)</p> <p><b>Mientras</b> NO sea Final de Fichero <b>Hacer</b></p> <p>    Visualizar Nombre, Curso, Nota</p> <p>    Cuenta = Cuenta + 1</p> <p>    Suma = Suma + Nota</p> <p>    Leer Registro (Nombre, Curso, Nota)</p> <p><b>Fin mientras</b></p> <p>Media =Suma/Cuenta</p> <p>Visualizar “Nota media: “ Media</p> <p>Cerrar Fichero Notas (eof)</p> <p><b>Fin</b></p>	<pre> graph TD     INICIO([INICIO]) --&gt; Init[Cuenta=0&lt;br/&gt;Suma=0]     Init --&gt; Open[Abrir fichero]     Open --&gt; Read[/Nombre,&lt;br/&gt;Curso, Nota/]     Read --&gt; Decision{¿NO es Fin de&lt;br/&gt;Fichero?}     Decision -- SI --&gt; Process[Cuenta=Cuenta+1&lt;br/&gt;Suma=Suma+Nota]     Process --&gt; Read     Decision -- NO --&gt; Calc[Media=Suma/Cuenta]     Calc --&gt; DisplayMedia[/Nota media:&lt;br/&gt;Media/]     DisplayMedia --&gt; Close[Cerrar fichero]     Close --&gt; FIN([FIN])     </pre>

**EJEMPLO 6:** Programa que lee un número de teclado y muestra a qué día de la semana se corresponde.

Si el número leído es 1 visualizará Lunes, si es 2, visualizará Martes, si es 3 visualizará Miércoles, si es 4 Jueves, si es 5 Viernes, si es 6 Sábado y si es 7 Domingo.

Para cualquier otro valor visualizará “No válido”.

Se utiliza la estructura **Según sea** <variable> **Hacer**

**Caso** valor1: < instrucciones>

**Caso** valor2: < instrucciones>

**Caso** valor3: < instrucciones>

**Otro caso**

**Fin según.**

PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<p><b>Inicio</b></p> <p>Leer N</p> <p><b>Según sea N Hacer</b></p> <p><b>Caso 1:</b></p> <p>Visualizar “Lunes”</p> <p><b>Caso 2:</b></p> <p>Visualizar “Martes”</p> <p><b>Caso 3:</b></p> <p>Visualizar “Miércoles”</p> <p><b>Caso 4:</b></p> <p>Visualizar “Jueves”</p> <p><b>Caso 5:</b></p> <p>Visualizar “Viernes”</p> <p><b>Caso 6:</b></p> <p>Visualizar “Sábado”</p> <p><b>Caso 7:</b></p> <p>Visualizar “Domingo”</p> <p><b>Otro Caso:</b></p> <p>Visualizar “No válido”</p> <p><b>Fin según</b></p> <p><b>Fin</b></p>	<pre> graph TD     INICIO([INICIO]) --&gt; N[/N/]     N --&gt; N1{N=1}     N1 -- SI --&gt; L1([Lunes])     N1 -- NO --&gt; N2{N=2}     N2 -- SI --&gt; L2([Martes])     N2 -- NO --&gt; N3{N=3}     N3 -- SI --&gt; L3([Miércoles])     N3 -- NO --&gt; N4{N=4}     N4 -- SI --&gt; L4([Jueves])     N4 -- NO --&gt; N5{N=5}     N5 -- SI --&gt; L5([Viernes])     N5 -- NO --&gt; N6{N=6}     N6 -- SI --&gt; L6([Sábado])     N6 -- NO --&gt; N7{N=7}     N7 -- SI --&gt; L7([Domingo])     N7 -- NO --&gt; L8([No válido])     L1 --&gt; FIN([FIN])     L2 --&gt; FIN     L3 --&gt; FIN     L4 --&gt; FIN     L5 --&gt; FIN     L6 --&gt; FIN     L7 --&gt; FIN     L8 --&gt; FIN </pre>

## ACTIVIDADES DE AMPLIACIÓN

---

1. Modifica el pseudocódigo y el diagrama de flujo de los Ejemplos 3 y 4, de forma que se utilice la estructura *Mientras <condición> Hacer <instrucciones> Fin mientras*.

### Ejemplo 3

Inicio

Visualizar “escribe dos números”

Leer A, B,

Mientras <condición>

sean distintas [repetir]

Hacer <instrucciones>

Visualizar “escribe dos números”

Leer A, B,

Hasta que Sean igual [Fin]

Corrección

Inicio

Visualizar “escribe dos números”

Leer A, B

Mientras A!=B Hacer

Visualizar “escribe dos números”

Leer A, B

Fin Mientras

Fin

### Ejemplo 4

Inicio

Declarar Contador

Declarar Suma

While(contador<=10) ----- Hacer

Visualizar “Escribe un número”

Declarar número

Leer número

Suma=Suma+número

Contador++ (+1 cada vez)

Fin Mientras

El contador llegue a 10

Hacer la suma

Sysio leer ("La suma de todos los números es " +suma)

### Corrección

#### Inicio

Declarar cuenta=0

Declarar Suma=0

Mientras Cuenta < 10 ----- Hacer

Visualizar "Escribe un número"

Leer A

Cuenta = Cuenta +1

Suma = Suma + A

Fin Mientras

Visualizar "La suma es: Suma"

#### Fin

2. Modifica el pseudocódigo y el diagrama de flujo del Ejemplos 5, de forma que se utilice la estructura *Repetir <instrucciones> Hasta <condición>*.

#### Inicio

Declarar Nombre, Curso, Nota

Declarar N (número alumnos total)

Declarar Contador=0

Declarar Suma=0

Declarar Media

**Repetir** (Contador<=N)<instrucciones>

Abrir Fichero Notas

Leer Registro y Mostrar (Nombre, Curso, Nota)

Contador++ (Contador +1)

Suma+=Nota (Suma=Suma+Nota)

**Hasta** (que Contador = N)

Media=Suma/N

Mostar la media por pantalla

**Fin**

Corrección

Inicio

Declarar Cuenta = 0

Declarar Suma = 0

Declarar media

Abrir Fichero Notas

Leer registro(Nombre, Curso, Nota)

Repetir

Visualizar (Nombre, Curso, Nota)

Cuenta = Cuenta+1

Suma = Suma+Nota

Leer registro (Nombre, Curso, Nota)

Hasta (Final de Fichero)

Media = Suma/Cuenta

Visualizar ("Nota Media: " Media)

Cerrar fichero

Fin



3. Utilizando como modelo el Ejemplo 6, realiza un pseudocódigo y un diagrama de flujo que lea un número y muestre la nota a la que corresponde. Si el valor es 5 debe mostrar *Suficiente*, si es 6 *Bien*, si es 7 *Notable bajo*, si es 8 *Notable alto*, si es 9 *Sobresaliente*, si es 10 *Matrícula de honor*; para cualquier otro valor debe mostrar *Suspense*.

PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<p><b>Inicio</b></p> <p>Leer N</p> <p><b>Switch (Según sea N) &lt;Hacer&gt;</b></p> <p><b>Caso 1:</b> para N = 5</p> <p>Visualizar “Suficiente”</p> <p><b>Caso 2:</b> para N = 6</p> <p>Visualizar “Bien”</p> <p><b>Caso 3:</b> para N = 7</p> <p>Visualizar “Notable Bajo”</p> <p><b>Caso 4:</b> para N = 8</p> <p>Visualizar “Notable Alto”</p> <p><b>Caso 5:</b> para N = 9</p> <p>Visualizar “Sobresaliente”</p> <p><b>Caso 6:</b> para N = 10</p> <p>Visualizar “Matrícula de Honor”</p> <p><b>Default (Otro Caso):</b></p> <p>Visualizar “Suspense”</p> <p><b>Fin según</b></p> <p><b>Fin</b></p>	<pre> graph TD     INICIO([INICIO]) --&gt; N[/N/]     N --&gt; C1[Caso 1 para N=5]     C1 -- SI --&gt; V1[Visualizar Suficiente]     C1 -- NO --&gt; C2[Caso 2 para N=6]     C2 -- SI --&gt; V2[Visualizar Bien]     C2 -- NO --&gt; C3[Caso 3 para N=7]     C3 -- SI --&gt; V3[Visualizar Notable Baio]     C3 -- NO --&gt; C4[Caso 4 para N=8]     C4 -- SI --&gt; V4[Visualizar Notable Alto]     C4 -- NO --&gt; C5[Caso 5 para N=9]     C5 -- SI --&gt; V5[Visualizar Sobresaliente]     C5 -- NO --&gt; C6[Caso 6 para N=10]     C6 -- SI --&gt; V6[Visualizar Matrícula de Honor]     C6 -- NO --&gt; D1[Default Otro Caso]     D1 -- SI --&gt; N5{N&lt;5}     N5 -- SI --&gt; V7[Visualizar Suspense]     N5 -- NO --&gt; V7     D1 -- NO --&gt; V7     V1 --&gt; FIN([FIN])     V2 --&gt; FIN     V3 --&gt; FIN     V4 --&gt; FIN     V5 --&gt; FIN     V6 --&gt; FIN     V7 --&gt; FIN   </pre>

4. Modifica el pseudocódigo del Ejemplo 6 de manera que se utilice la estructura *Si <condición> Entonces <Instrucciones> Si no <Instrucciones> Fin si*, en lugar de Según sea.

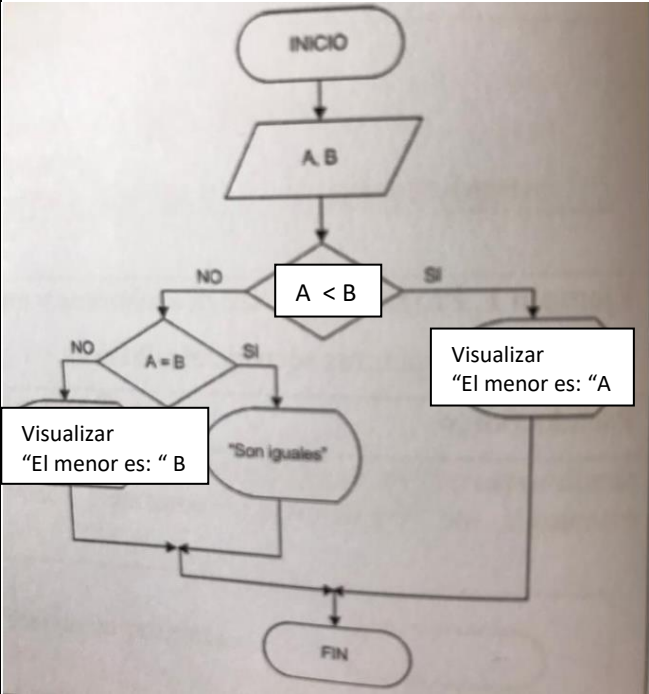
PSEUDOCÓDIGO
<b>Inicio</b>
Declarar N
Leer N
<b>If</b> (N=5)<condición>
[entonces] Mostrar “Suficiente”
<b>If</b> (N=6) <condición>
[entonces] Mostrar “Bien”
<b>If</b> (N=7) <condición>
[entonces] Mostrar “Notable Bajo”
<b>If</b> (N=8) <condición>
[entonces] Mostrar “Notable Alto”
<b>If</b> (N=9) <condición>
[entonces] Mostrar “Sobresaliente”
<b>If</b> (N=10) <condición>
[entonces] Mostrar “Matrícula de Honor”
<b>Else</b> [si no](Mostrar “Insuficiente”)<instrucciones>
<b>Fin si</b>

Esto esta mal porque el ejemplo 6 es el de los días de la semana no las notas...

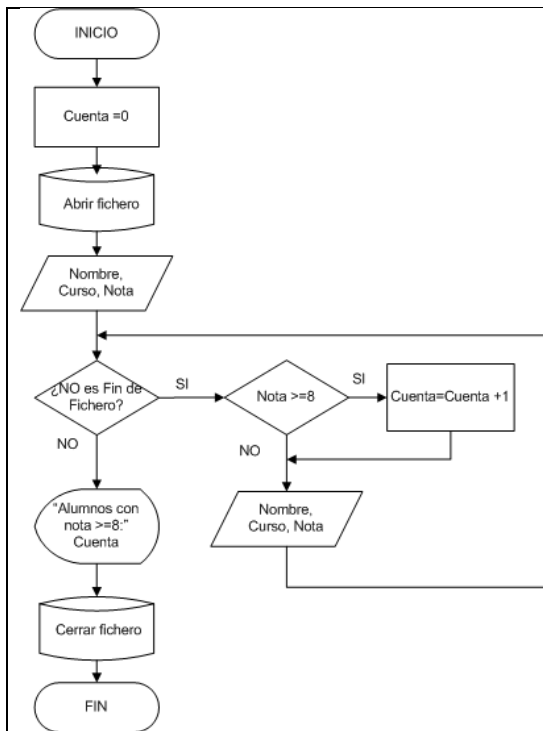
PSEUDOCÓDIGO
<b>Inicio</b>
Declarar N
Leer N
<b>If</b> (N=1)<condición>
[entonces] Mostrar “Lunes”
Else If
<b>If</b> (N=2) <condición>
[entonces] Mostrar “Martes”
Else If
<b>If</b> (N=3) <condición>
[entonces] Mostrar “Miercoles”
Else If
<b>If</b> (N=4) <condición>
[entonces] Mostrar “Jueves”
Else If
<b>If</b> (N=5) <condición>
[entonces] Mostrar “Viernes”
Else If
<b>If</b> (N=6) <condición>
[entonces] Mostrar “Sábado”
Else If
<b>If</b> (N=7)
[entonces] Mostrar “Domingo”
<b>Else</b> [si no](Mostrar “No válido”)<instrucciones>
Fin si
<b>Fin si</b>
<b>Fin si... etc</b>
<b>FIN</b>

5. Utilizando como modelo el Ejemplo 2, realiza un pseudocódigo y un diagrama de flujo que lea un\* número y muestre el menor de ellos. Si los números son iguales debe mostrar un mensaje indicándolo.

*\*Nota: profesora esto está bien? No deberían de ser dos números?*

PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<p><b>Inicio</b></p> <p>Declarar A, B</p> <p>Leer A, B</p> <p><b>Si</b> <math>A &lt; B</math> <b>entonces</b></p> <p>    Visualizar “El menor es: “ A</p> <p><b>Sino</b></p> <p>    <b>Si</b> <math>A = B</math> <b>entonces</b></p> <p>        Visualizar “Son iguales”</p> <p>    <b>Sino</b></p> <p>        Visualizar “El menor es: “ B</p> <p>    <b>Fin Si</b></p> <p><b>FinSi</b></p> <p><b>Fin</b></p>	 <pre> graph TD     INICIO([INICIO]) --&gt; Input[/A, B/]     Input --&gt; Cond1{A &lt; B}     Cond1 -- Si --&gt; OutA[Visualizar "El menor es: "A]     Cond1 -- NO --&gt; Cond2{A = B}     Cond2 -- Si --&gt; OutEq([Son iguales])     Cond2 -- NO --&gt; OutB[Visualizar "El menor es: "B]     OutA --&gt; Join(( ))     OutEq --&gt; Join     OutB --&gt; Join     Join --&gt; FIN([FIN])     </pre>

6. Escribe el pseudocódigo para el siguiente diagrama de flujo:



**Inicio**

Declarar Nombre, Curso, Nota

Declarar Contador=0

Declarar ContadorNota=0

Declarar N (alumnos)

**While** (Contador<=N) { <No sea fin de fichero>

Abrir Fichero

Leer Nombre, Curso, Nota

**If** (Nota>=8) {

ContadorNota++ (contador +1) }

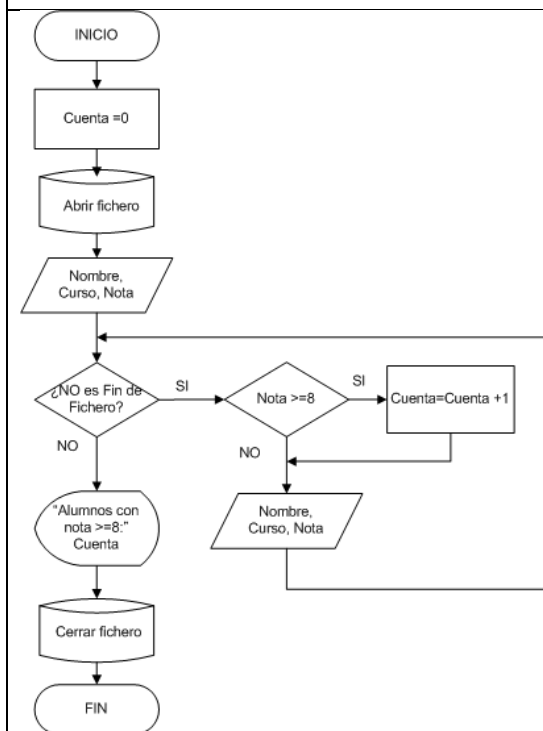
Contador ++ (Contador +1)

**Fin mientras** [Contador=N]

Mostrar ("Alumnos con nota igual o superior a 8: "+ContadorNota)

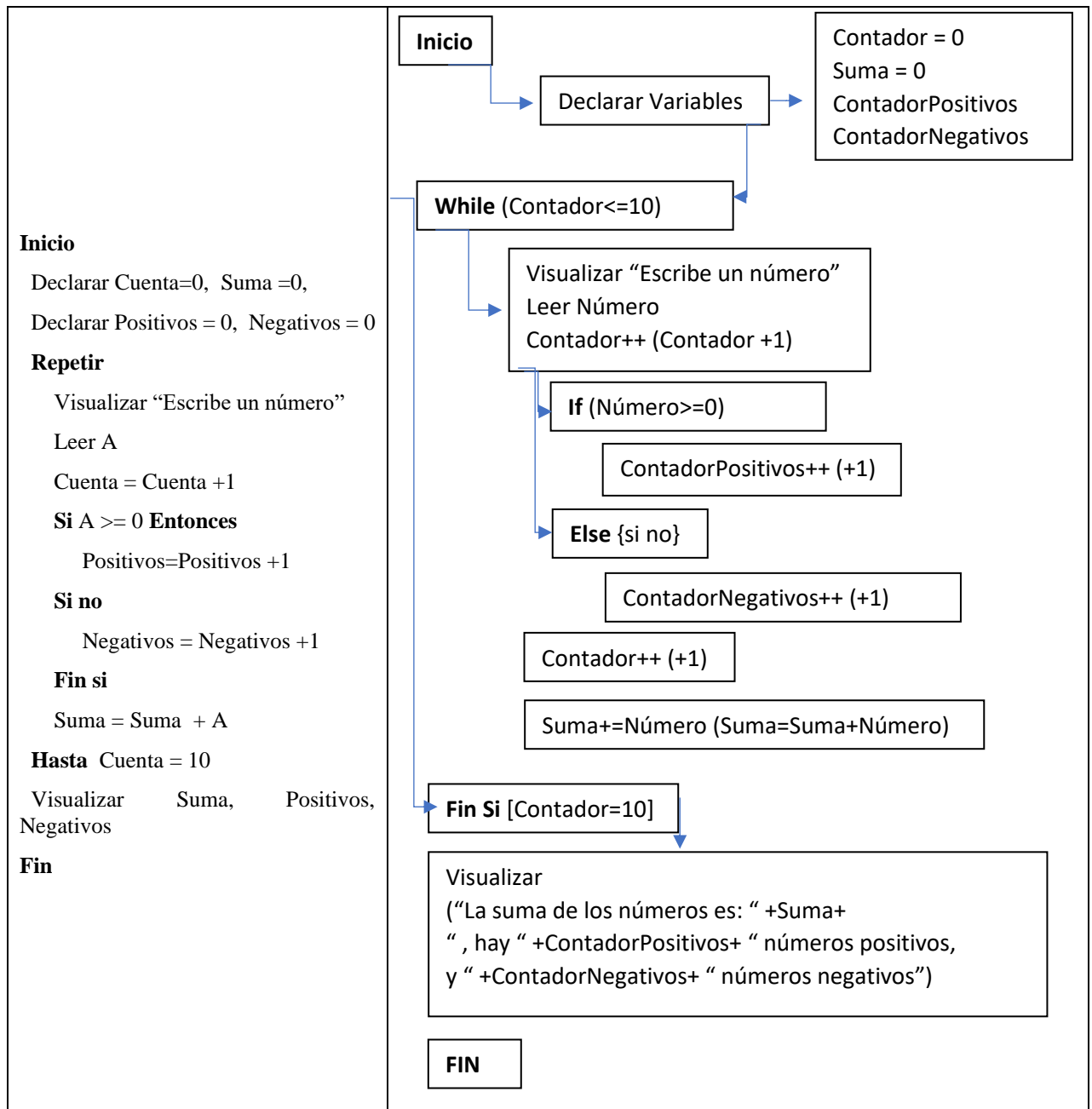
Cerrar Fichero

**FIN**



**Inicio**

7. Escribe el diagrama de flujo para el siguiente pseudocódigo:



8. Se desea realizar un sencillo sistema de gestión de notas en el que van a interactuar profesores y alumnos. El sistema permitirá a un alumno consultar la nota de las asignaturas en que está matriculado y a un profesor introducir las notas de los cursos a los que imparte clase. Tanto el profesor como el alumno deben identificarse para entrar al sistema mediante un nombre de usuario y una clave, por lo tanto, debe existir un almacén con los usuarios que pueden entrar en el sistema. Si la identificación no es correcta el sistema rechaza al usuario. Los procesos para alumno y profesor son los siguientes:

- Para alumno, 2 procesos: comprobar si el usuario está registrado y consultar la nota de la asignatura solicitada, estas notas estarán en un almacén. Cuando el usuario se identifica en el sistema puede introducir el nombre de la asignatura para que este le devuelva la nota. Para nombrar el flujo de datos entre el proceso de comprobación de usuario y de consulta de notas puedes usar: *pantalla de usuario*.
- Para profesor, 3 procesos: comprobar si el usuario está registrado, solicitar el curso al que introducirá las notas e introducir las notas del curso solicitado. Para nombrar el flujo de datos entre el proceso de comprobación y de solicitud de curso puedes usar: *pantalla de profesor*, y para el de solicitud de curso e introducción de notas: *pantalla de curso*. Cuando el profesor se identifica envía al sistema el curso del que desea introducir las notas y las notas, estas se irán registrando en un almacén.

Se pide realizar los DFDs de nivel 0, 1 y 2. Define dos almacenes, uno para los usuarios que se identifican en el sistema y otro para las notas.

