

DAM/DAW Programación

UNIDAD 4.1

ESTRUCTURAS DE CONTROL I

SOLUCIONES

1. ESTRUCTURAS DE SELECCIÓN

Escribe el PSEUDOCÓDIGO, dibuja su DIAGRAMA DE FLUJO e implementa en JAVA de los siguientes supuestos

1.1.

PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<pre>Inicio Leer N1 Leer N2 Si (N1 mod N2) = 0 Mostrar "N1 es múltiplo de N2" SiNo Mostrar "N1 no es múltiplo de N2" Fin</pre>	<pre>graph TD Start([start]) --> Input[/Introduce dos números/] Input --> Read[/N1, N2/] Read --> Decision{N1%N2 = 0} Decision -- false --> OutputFalse[/N1 no múltiplo de N2/] Decision -- true --> OutputTrue[/N1 múltiplo de N2/] OutputFalse --> End([end]) OutputTrue --> End</pre>

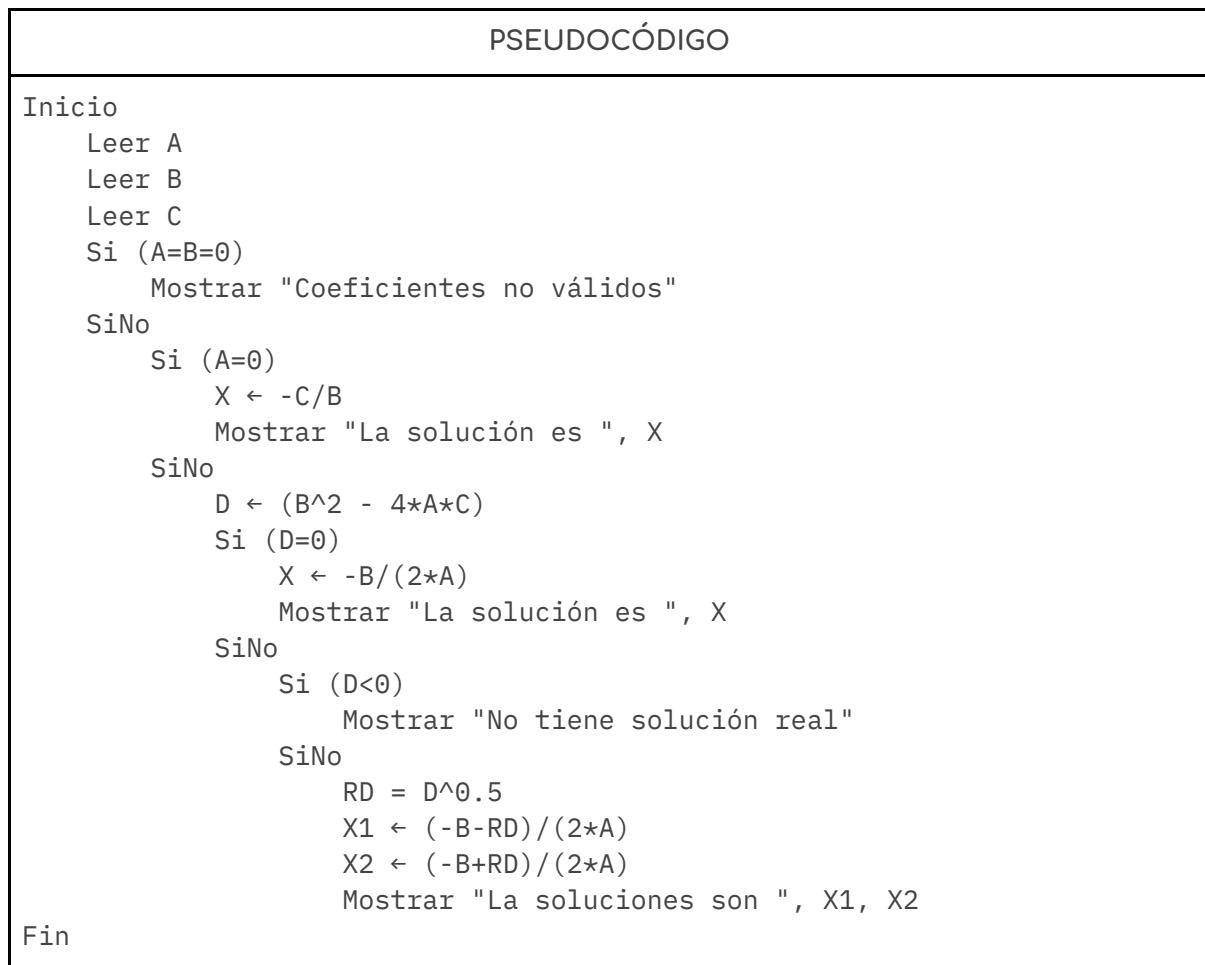
1.2.

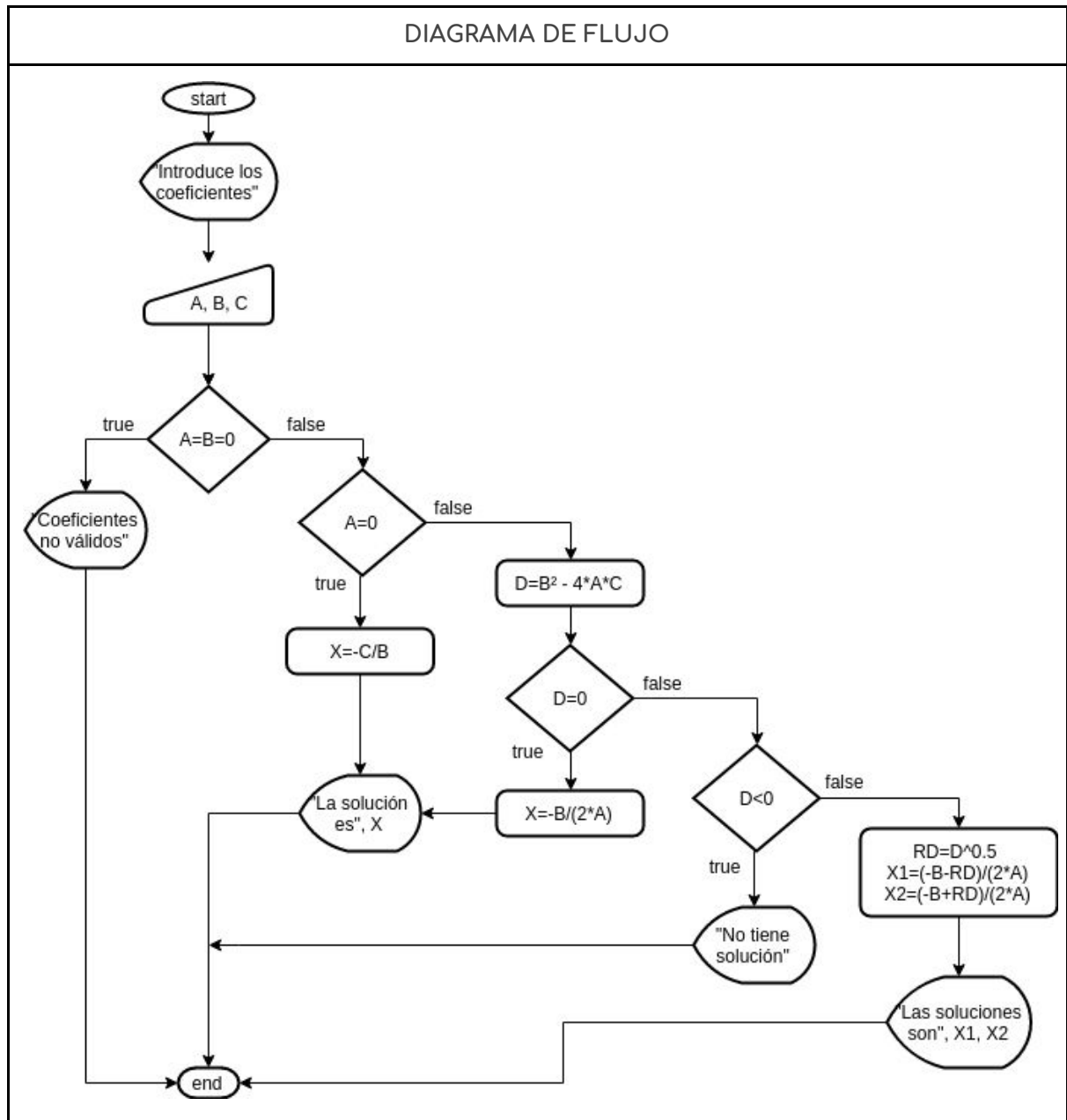
PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<div>Inicio Leer Num Si (Num=0) OR (Num mod 2=0) Mostrar "Num es PAR" SiNo Mostrar "Num es IMPAR" Fin</div>	<pre>graph TD; Start([start]) --> Input[/Introduce un número/]; Input --> Num[/Num/]; Num --> Decision{Num=0 OR Num%2=0}; Decision -- true --> Output1[/Num es PAR/]; Decision -- false --> Output2[/Num es Impar/]; Output1 --> End([end]); Output2 --> End;</pre>

1.3.

PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<div>Inicio</div> <div>Leer N1</div> <div>Leer N2</div> <div>Leer N3</div> <div>Si (N1=N2=N3)</div> <div>Mostrar "3 iguales"</div> <div>SiNo</div> <div>MAX ← N1</div> <div>Si (N2>MAX)</div> <div>MAX ← N2</div> <div>Si (N3>MAX)</div> <div>MAX ← N3</div> <div>Mostrar "El mayor es ", MAX</div> <div>Fin</div>	<pre>graph TD; Start([start]) --> Input[/Introduce tres números/]; Input --> Read[/N1, N2, N3/]; Read --> Cond1{N1=N2=N3}; Cond1 -- true --> Out1([3 iguales]); Out1 --> End([end]); Cond1 -- false --> Assign1[MAX = N1]; Assign1 --> Cond2{N2 > MAX}; Cond2 -- true --> Assign2[MAX = N2]; Assign2 --> Cond3{N3 > MAX}; Cond3 -- true --> Assign3[MAX = N3]; Assign3 --> Out2([El mayor es MAX]); Cond2 -- false --> Out2; Cond3 -- false --> Out2; Out2 --> End;</pre>

1.4.

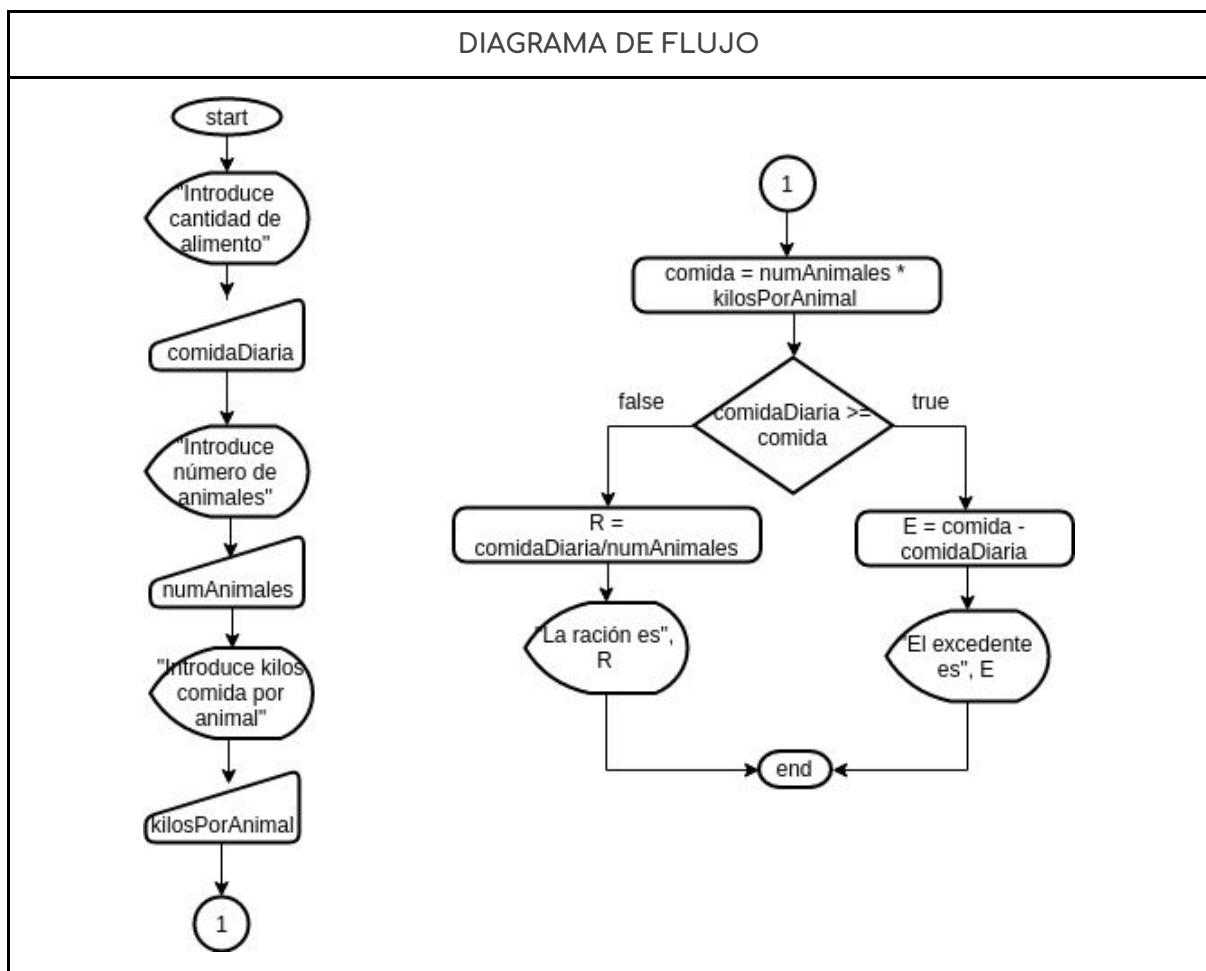
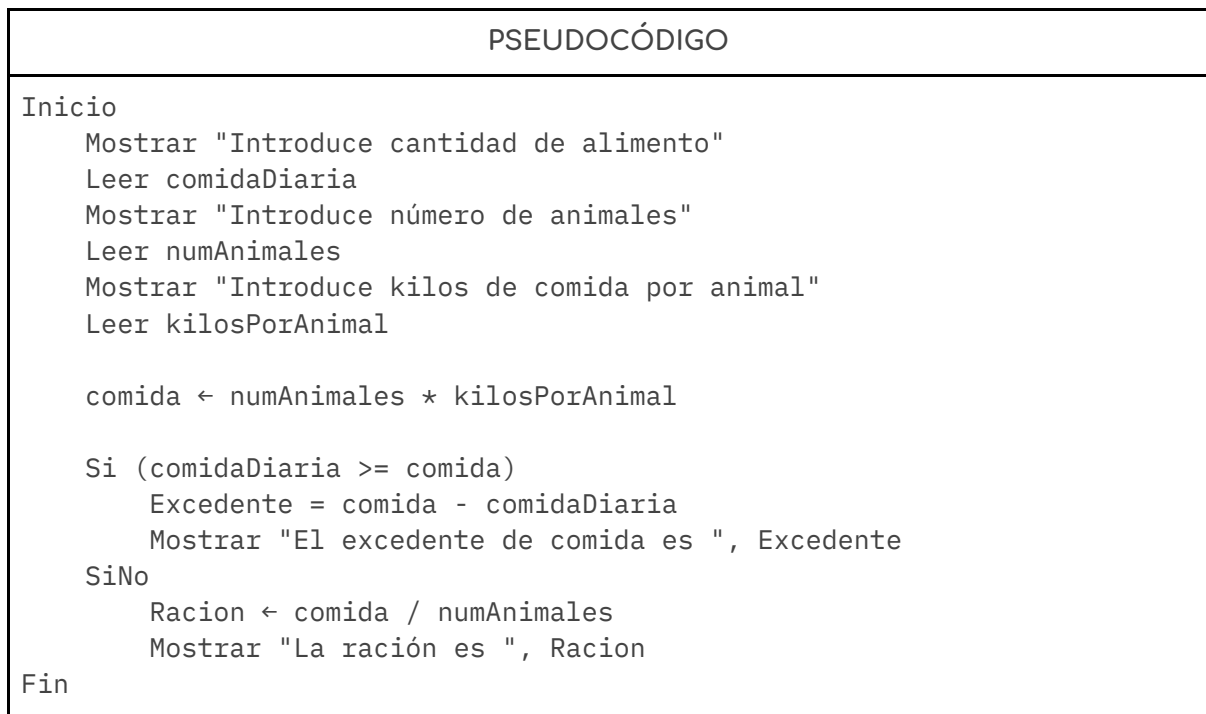




1.5.

PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<p>Inicio</p> <p>Mostrar "Introduce DNI"</p> <p>Leer DNI</p> <p>Si (DNI<10000000 OR DNI>99999999)</p> <p>Mostrar "Número DNI no válido"</p> <p>SiNo</p> <p>id ← DNI mod 23</p> <p>L ← "TRWAGMYFPDXBNJZSQVHLCKE"[id]</p> <p>Mostrar "La letra es", L</p> <p>Fin</p>	<pre>graph TD; start([start]) --> IO1[/Introduce DNI/]; IO1 --> DNI[/DNI/]; DNI --> Decision{DNI < 10000000 OR DNI > 99999999}; Decision -- true --> IO2[/Número DNI no válido/]; IO2 --> end([end]); Decision -- false --> Process1[id = DNI % 23]; Process1 --> Process2[L = "TRWAGMYFPDXBNJZSQVHLCKE" [id]]; Process2 --> IO3[/"La letra es", L/]; IO3 --> end;</pre>

1.6.



2. ESTRUCTURAS ITERATIVAS

Escribe el PSEUDOCÓDIGO, dibuja su DIAGRAMA DE FLUJO e implementa en JAVA de los siguientes supuestos

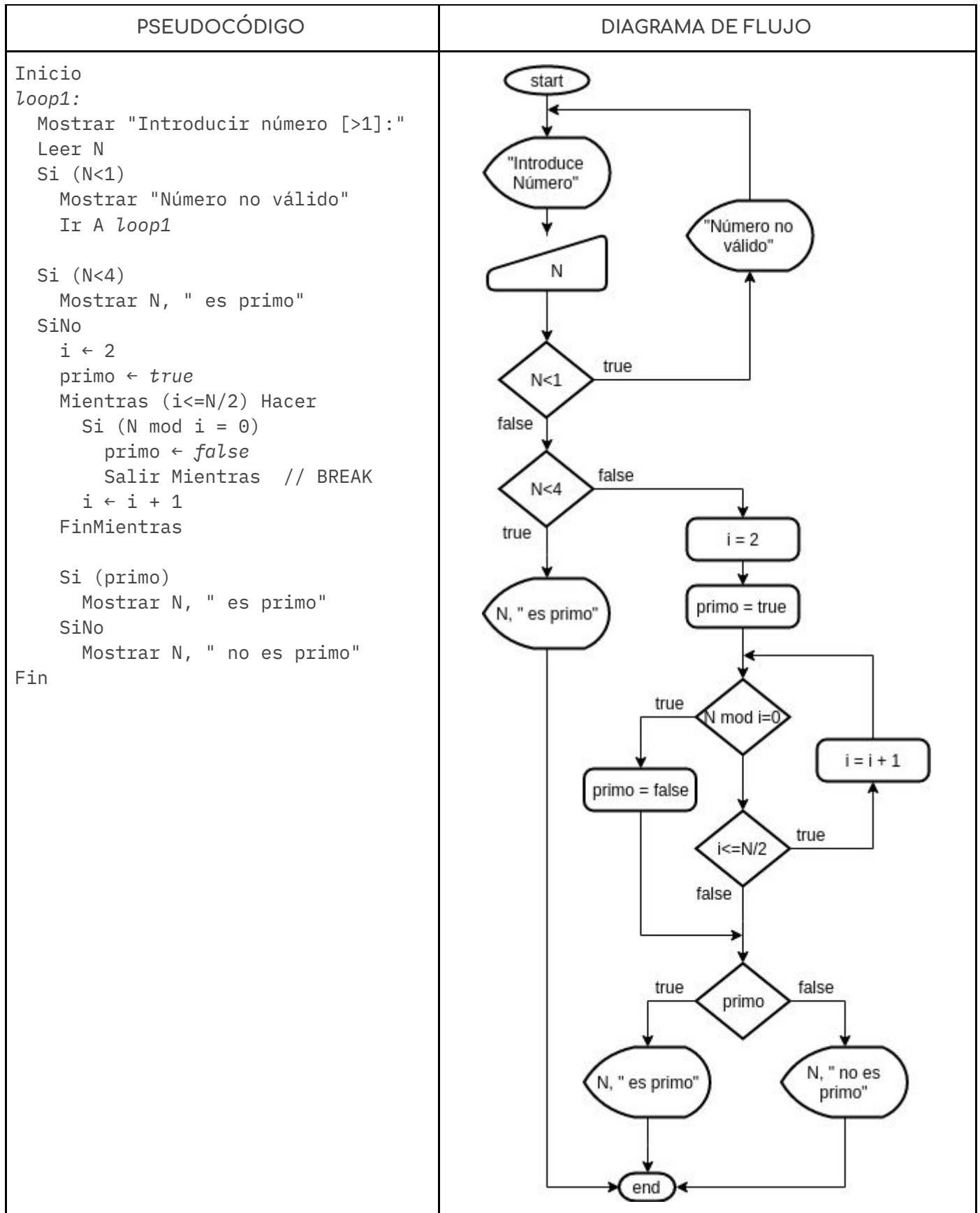
2.1.

PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<pre>Inicio loop1: Mostrar "Introducir número [0-9]:" Leer N Si (N<0 OR N>9) Mostrar "Número no válido" Ir A loop1 i ← 0 loop2: M ← i * N Mostrar N, " x ", i, " = ", M Si (i<=10) i ← i + 1 Ir A loop2 Fin</pre>	<pre>graph TD Start([start]) --> Input[/Introduce Número/] Input --> Read[/N/] Read --> Cond1{N >= 0 AND N <= 9} Cond1 -- false --> Output1[/Número no válido/] Output1 --> Input Cond1 -- true --> Init[i = 0] Init --> Calc[M = i * N] Calc --> Print[/N, "x", i, "=", M/] Print --> Cond2{i <= 10} Cond2 -- true --> Inc[i = i + 1] Inc --> Calc Cond2 -- false --> End([end])</pre>

2.2.

PSEUDOCÓDIGO	DIAGRAMA DE FLUJO
<pre>Inicio Mostrar "*****" Mostrar "* TABLA DE MULTIPLICAR *" Mostrar "*****" N ← 0 Mientras (N<=10) Hacer Mostrar "-----" Mostrar " TABLA DEL ", N Mostrar "-----" i ← 0 Mientras (i<=10) Hacer M ← i * N Mostrar N, " x ", i, " = ", M i ← i + 1 FinMientras FinMientras Fin</pre>	

2.3.



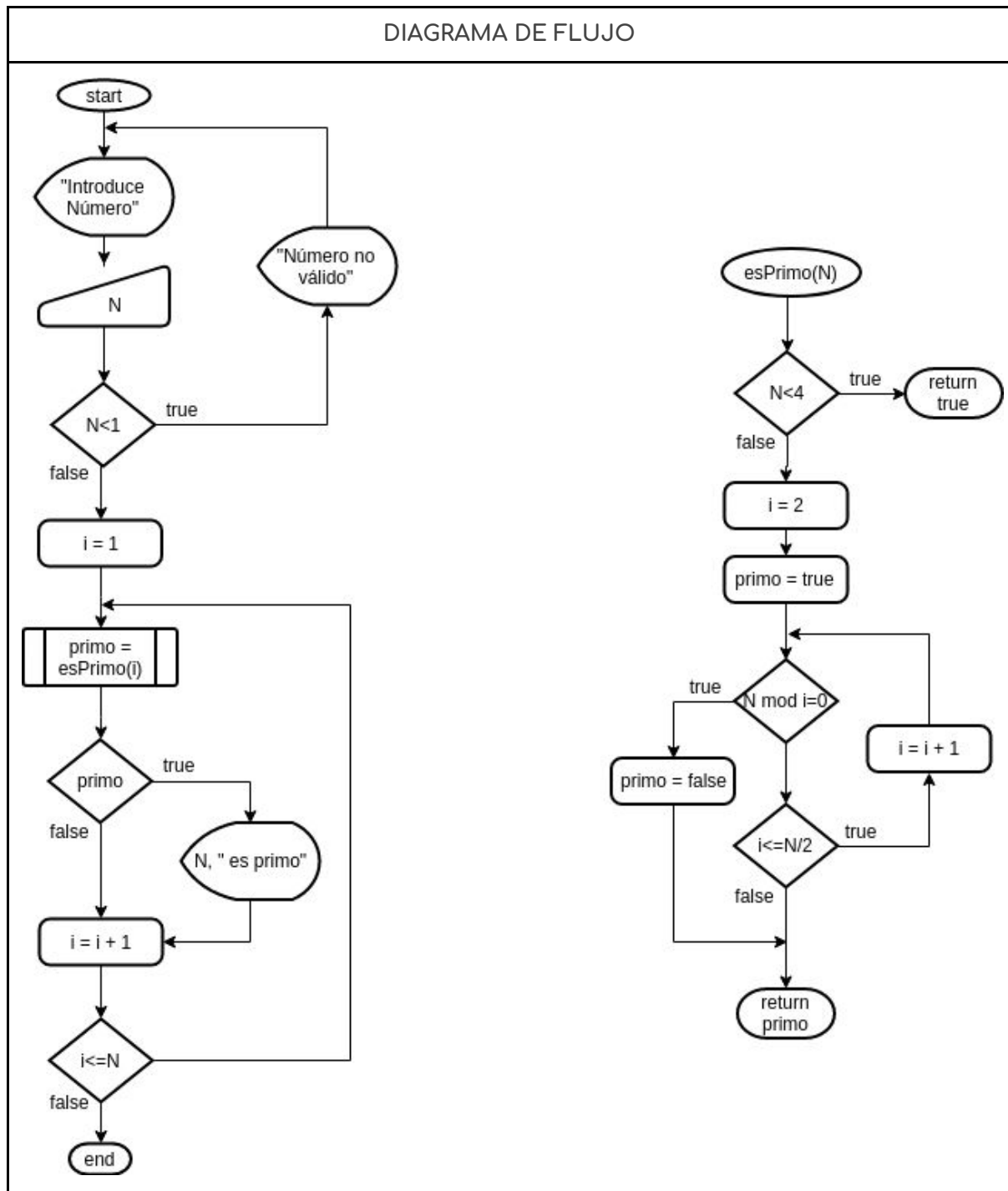
2.4.

PSEUDOCÓDIGO

```
Funcion esPrimo(N)
  Si (N<4)
    Retorna true
  SiNo
    i ← 2
    primo ← true
    Mientras (i<=N/2) Hacer
      Si (N mod i = 0)
        primo ← false
        Salir Mientras // BREAK
      i ← i + 1
    FinMientras
    Retorna primo
FinFuncion

Inicio
loop1:
  Mostrar "Introducir número [>1]:"
  Leer N
  Si (N<1)
    Mostrar "Número no válido"
    Ir A loop1

  Para i ← 1 Hasta N Hacer // Equivale a for
    Si esPrimo(i)
      Mostrar i
    FinSi
  FinPara
Fin
```

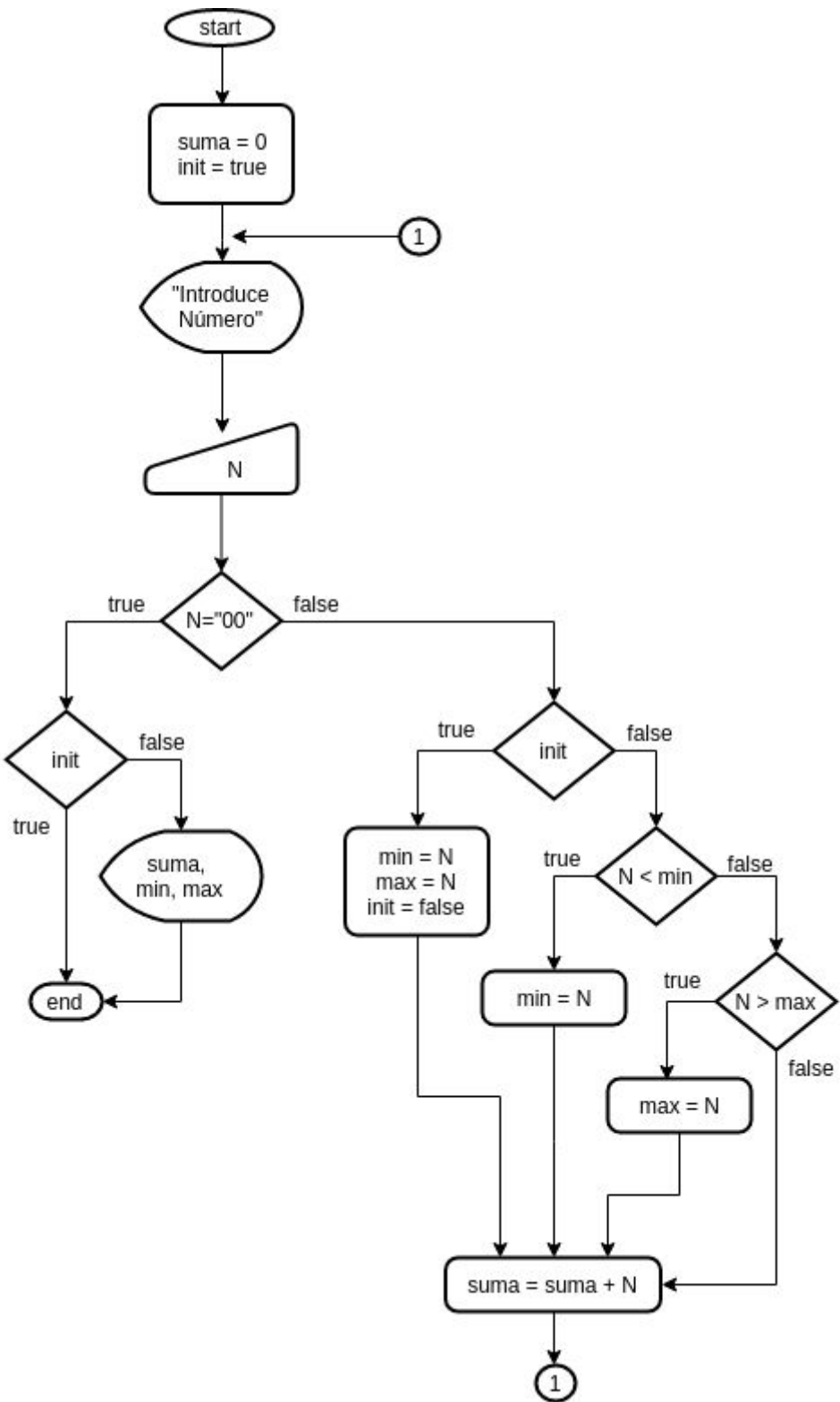


2.5.

PSEUDOCÓDIGO

```
Inicio
  primero ← true
  suma ← 0
  Mientras (N != "00") Hacer
    Mostrar "Introducir número:"
    Leer N
    Si (N = "00")
      Si (no primero)
        Mostrar suma, min, max
      FinSi
    SiNo
      Si (primero)
        min = N
        max = N
        primero ← false
      SiNo
        Si (N<min)
          min ← N
        SiNo
          Si (N>max)
            max ← N
          FinSi
        FinSi
      FinSi
    suma ← suma + N
  FinMientras
Fin
```

DIAGRAMA DE FLUJO

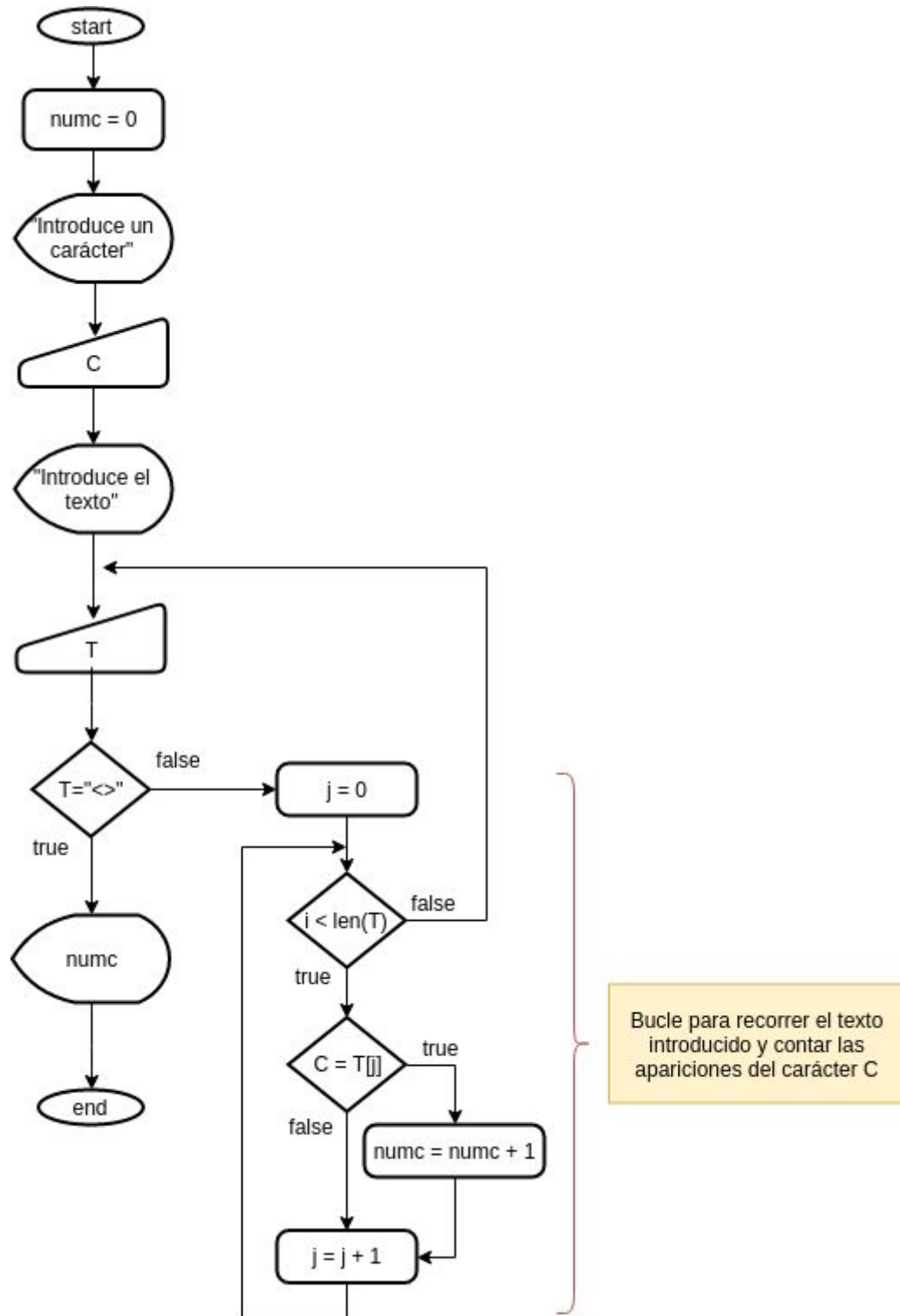


2.6.

PSEUDOCÓDIGO

```
Inicio
  numc ← 0
  Mostrar "Introduce un carácter:"
  Leer C
  Mostrar "Introducir texto:"
  Mientras (T != "<>") Hacer
    Leer T // Lee un línea de texto
    Para j ← 0 Hasta LEN(T)-1 Hacer // LEN: Función de librería
      Si (C=T[j])
        numc ← numc + 1
      FinSi
    FinPara
  FinMientras
  Mostrar numc
Fin
```

DIAGRAMA DE FLUJO

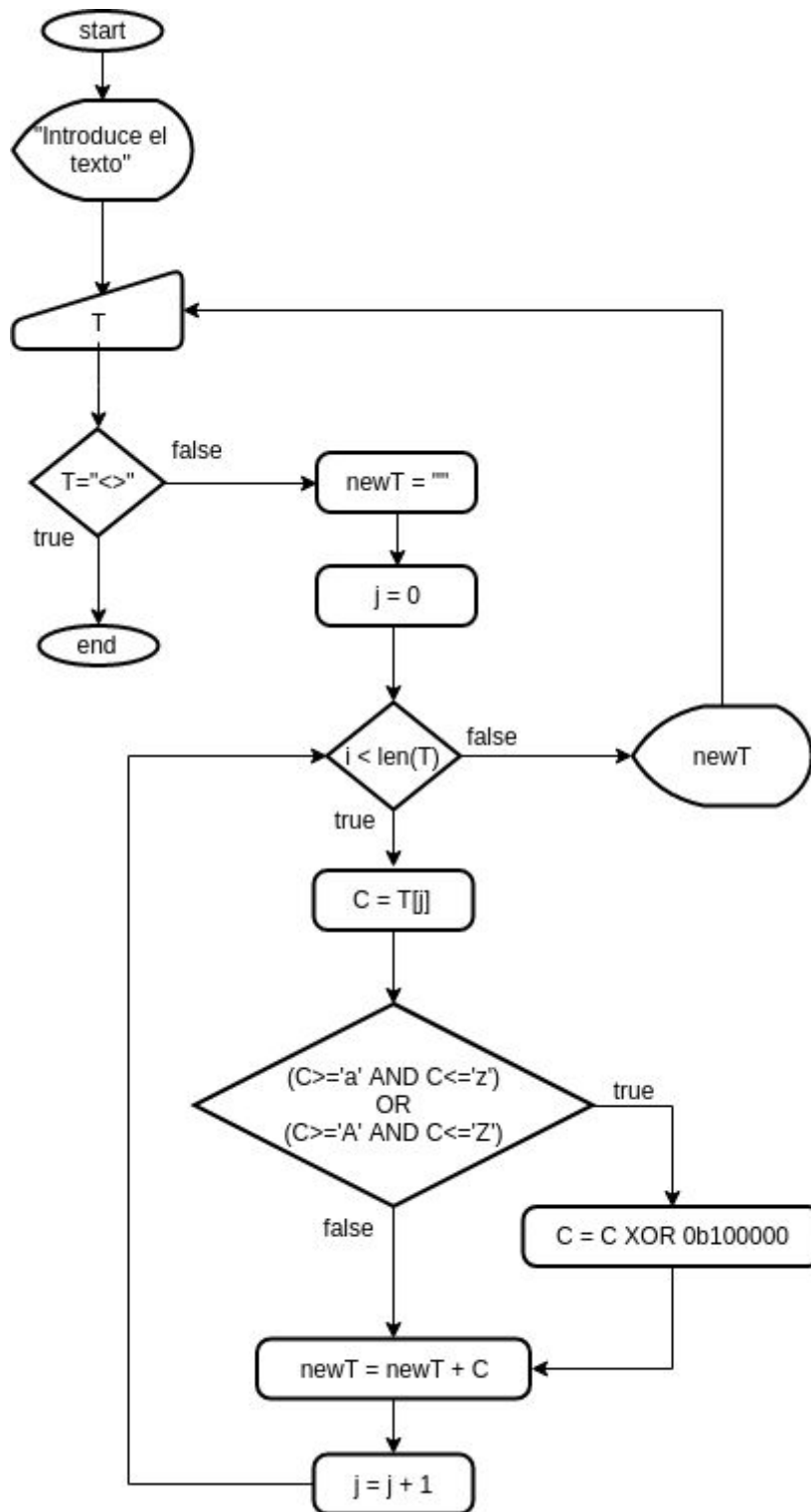


2.7.

PSEUDOCÓDIGO

```
Inicio
  Mostrar "Introducir texto:"
  Mientras (T != "<>") Hacer
    Leer T          // Lee un línea de texto
    newT ← ""       // Cadena vacía
    Para j ← 0 Hasta LEN(T)-1 Hacer // LEN: Función de librería
      C ← T[j]
      Si ((C>='a' && C<='z') || (C>='A' && C<='Z'))
        C ← C XOR 0b100000
      FinSi
      newT ← newT + C // Concatenar carácter
    FinPara
    Mostrar newT
  FinMientras
Fin
```

DIAGRAMA DE FLUJO



2.8.

PSEUDOCÓDIGO

```

Funcion eliminaTildes(S)
  VS ← "aeiouAEIOU"
  VC ← "áéíóúÁÉÍÓÚ"
  newS ← S
  Para i ← 0 Hasta LEN(VC)-1 Hacer // LEN: Función de librería
    newS→REPLACE(VC[i], VS[i]) // REPLACE: Función de librería
  FinPara
  Retorna newS
FinFuncion

Inicio
  Mostrar "Introducir texto:"
  Leer T
  T ← eliminaTildes(T)
  S ← " ,;.!:¿?"
  pal ← true
  i ← 0
  j ← LEN(T)-1 // LEN: Función de librería
loop:
  Mientras ( i < j ) Hacer
    A ← T[i]
    B ← T[j]
    pos ← S→FIND(A) // FIND: Función de librería
    Si (pos!=-1)
      i ← i + 1
      Ir A loop // CONTINUE
    FinSi
    pos ← S→FIND(B) // FIND: Función de librería
    Si (pos!=-1)
      j ← j - 1
      Ir A loop // CONTINUE
    FinSi
    Si(A!=B)
      pal ← false
      Salir Mientras // BREAK
    FinSi
    i ← i + 1
    j ← j - 1
  FinMientras
  Si (pal)
    Mostrar "Es palíndromo"
  SiNo
    Mostrar "No es palíndromo"
  FinSi
Fin

```

DIAGRAMA DE FLUJO

