

BUAP[®]



VIRTUAL BUAP®

Metodología de la Programación

UNIDAD 1. Diseño estructurado usando diagramas de flujo y pseudocódigo

Estructuras de Control

- Si
 - Si – Si_no
 - Según (expresion) - Sea
 - Para
 - Mientras
 - Hacer - Mientras.
- Decision
- Repetición
-
- A diagram showing control structures grouped into two categories. The first category, 'Decision', is indicated by a red bracket on the right side and includes the items 'Si', 'Si – Si_no', and 'Según (expresion) - Sea'. The second category, 'Repetición', is indicated by a red bracket on the right side and includes the items 'Para', 'Mientras', and 'Hacer - Mientras.'

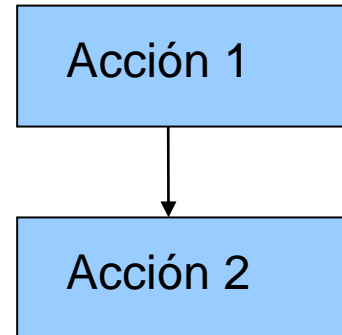
Estructuras básicas

Secuencial

- Una acción después de la otra
- El flujo coincide con el orden físico en que se colocan las instrucciones.
- Es una secuencia lineal.

Sintaxis

D.F.



Pseudocodigo

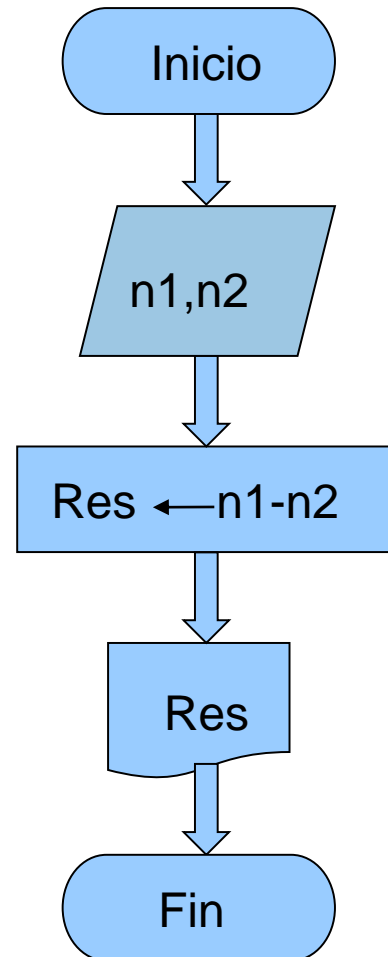
Acción 1

Acción 2

Ejemplo 1: Restar 2 números

Pseudocodigo

Diagrama
de Flujo



Algoritmo: Resta

Entrada: n1,n2

Salida: Res

Inicio

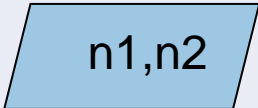
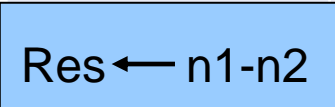

Leer(n1,n2)

Res ← n1-n2

Escribir(Res)

Fin Resta

Prueba

instrucción	Dato de entrada	operaciones	Memoria Variable y contantes	Dato de salida
			n1 n2 Res	
	5 7		n1 n2 Res 5 7	
		5-7 = -2	n1 n2 Res 5 7 -2	
			n1 n2 Res 5 7 -2	-2
			n1 n2 Res 5 7 -2	

Inicio

Fin

Selectivas

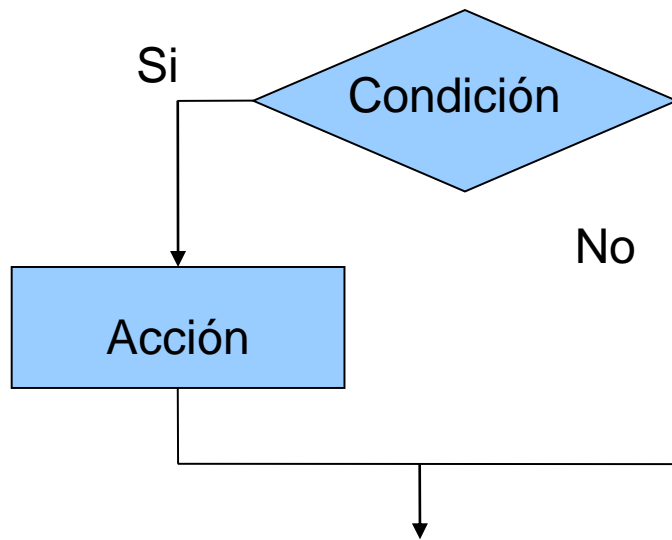
- Permiten controlar la ejecución de acciones que requieran ciertas condiciones para su realización.
- Aplicaciones a:
 - Opciones excluyentes
 - Para elegir de entre varias
 - Verificación de datos

Simples

- Si la condición es verdadera se ejecuta una acción (es). En caso contrario seguir con el flujo.

Sintaxis

D.F.



Pseudocódigo

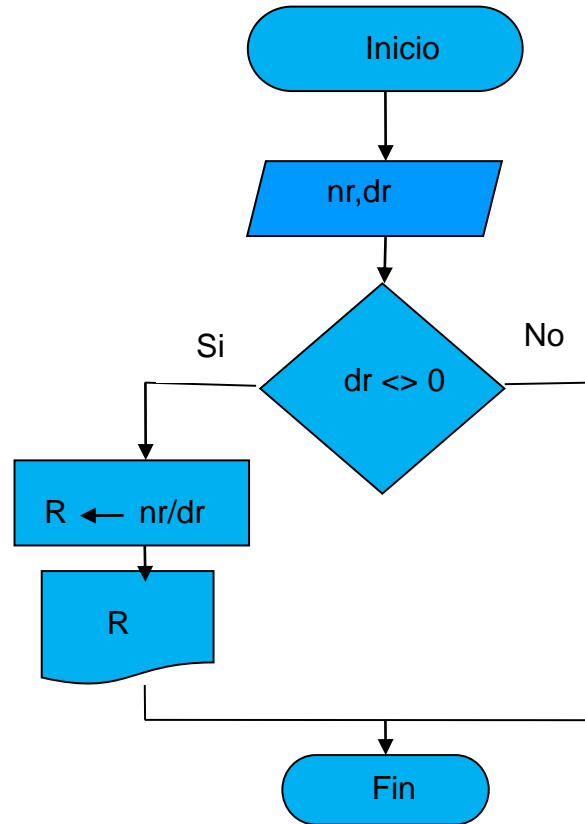
Si <condición> entonces

acción

Fin_si

Ejemplo: Dividir 2 números

- Considerar que el divisor es $\neq 0$



Algoritmo: Divide

Entrada: $nr \leftarrow 5, dr \leftarrow 3$

Salida: Res

Inicio

Leer (nr, dr)

Si $dr \neq 0$ entonces

$Res \leftarrow nr/dr$

Fin_si

Escribir (Res)

FinDivide

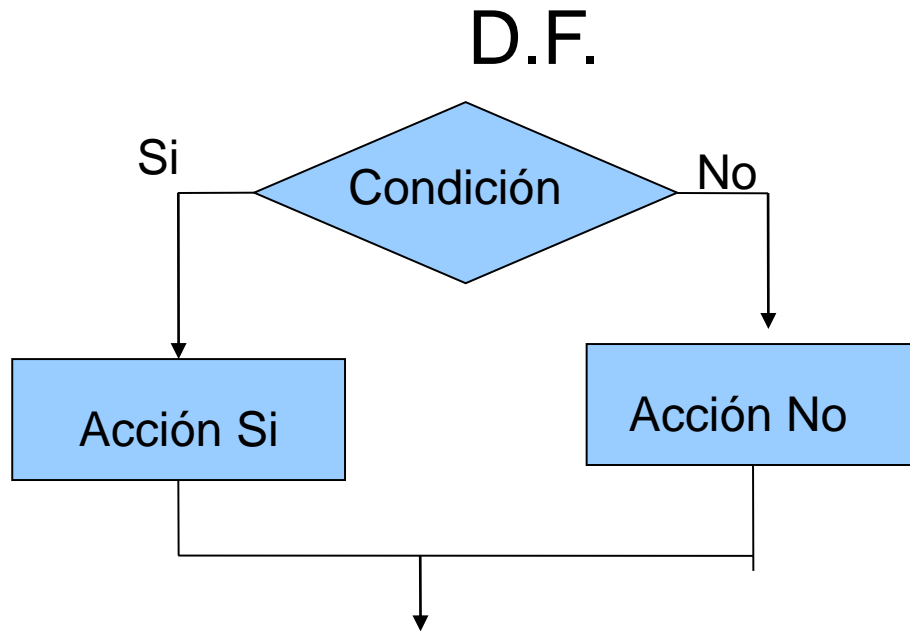
Ejemplo (todos)

- Hacer el D.F. y Pseudocódigo para que dada una calificación si ésta es aprobatoria (mayor o igual a 7) escribir 'FELICIDADES'.

Dobles

- Si la condición es verdadera se ejecuta una acción (es). Si es falsa se ejecuta otra acción(es).

Sintaxis



Pseudocódigo

Si <condición> entonces

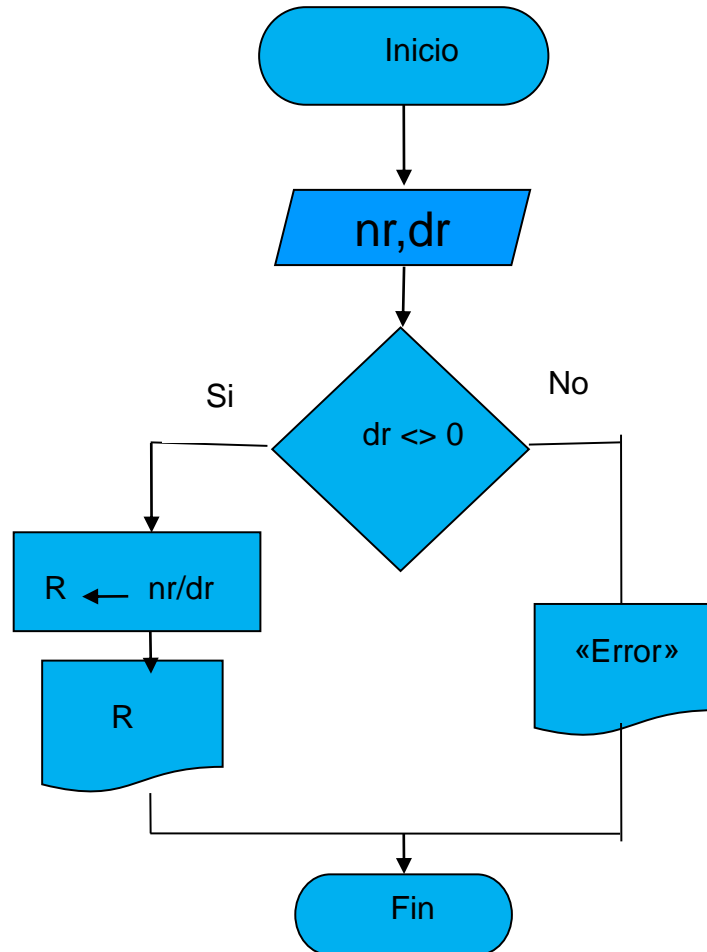
 acción_si

Si_no

 acción_no

Fin_si

Ejemplo: Dividir 2 números



Algoritmo: Divide

Entrada: nr,dr

Salida: R

Inicio

Leer (nr,dr)

Si $dr \neq 0$ entonces

$R \leftarrow nr/dr$

escribir (R)

Si_no

Escribir ("Error")

Fin_si

FinDivide

Ejercicio

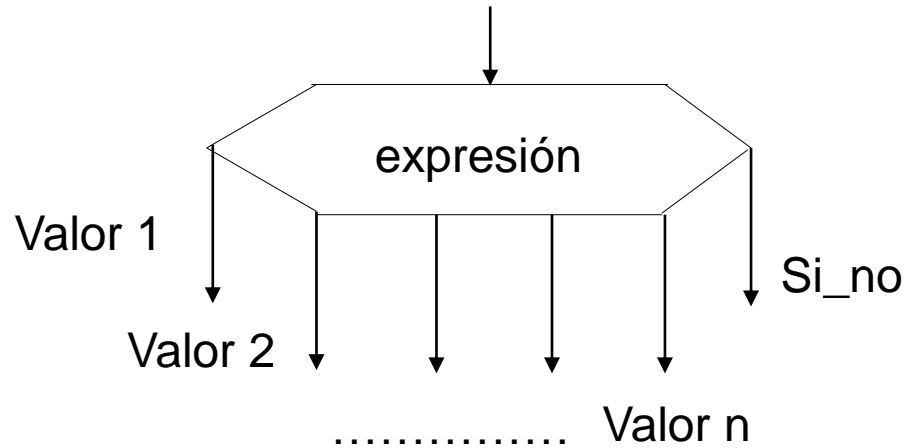
- Hacer el DF y Pseudocódigo para que dada una cantidad que representa una compra si ésta es menor a \$1000 entonces escribir 'Sigue participando' si no lo es, entonces realizar un descuento del 20% a la compra.

Múltiples

- Se evalúa una expresión para ejecutar una acción de mas de dos opciones.
- Cada acción (es) está ligado a una constante.
- Si el valor obtenido no está presente entonces se ejecuta la acción (es) en la cláusula **Si_no**.

Sintaxis

Diagrama de Flujo.



Pseudocódigo

Según <expresión> Sea

Caso<valor1>:acción 1

Caso<valor2>:acción n

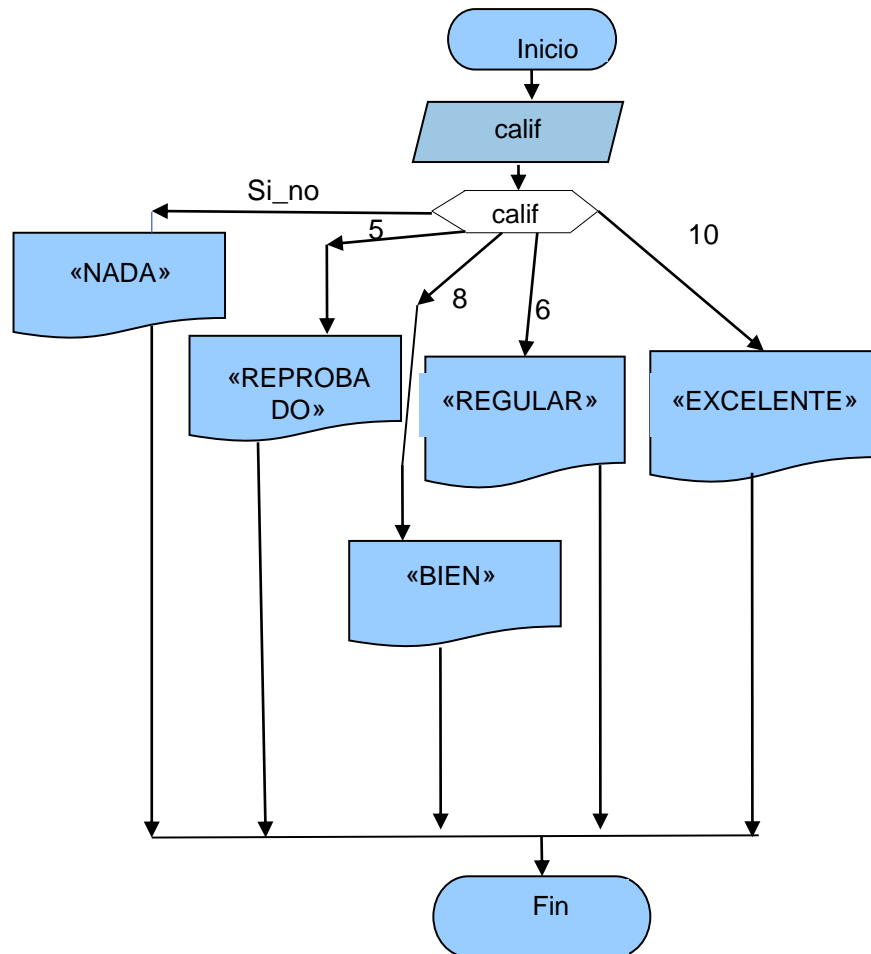
.....

Si_no

acción_no

Fin_según

Ejemplo:



Algoritmo: Escribe

Entrada: calif

Salida: Letreros

Inicio

Variables: calif de tipo entero

Leer (calif)

Según calif Sea

Caso 5: Escribir("REPROBADO") romper

Caso 6: Escribir("REGULAR") romper

Caso 8: Escribir("BIEN") romper

Caso 10: Escribir("EXCELENTE")

Si_No

Escribir("NADA QUE DECIR")

Fin_Según

Fin

Decisión múltiple o anidada

Algoritmo: Escribe

Entrada: calif

Salida: letreros

Inicio

Variables: calif de tipo Real

Leer (calif)

Si (calif \geq 0.0) AND (calif $<$ 6.0) entonces
Escribir("REPROBADO")

Si_No

Si (calif \geq 6.0) AND (calif $<$ 7.0) entonces
Escribir("REGULAR")

Si_No

Si (calif \geq 7.0) AND (calif $<$ 9.0) entonces
Escribir("BIEN")

Si_No

Si (calif \geq 9.0) AND (calif \leq 10.0) entonces
Escribir("EXCELENTE")

Si_No

Escribir("NADA QUE DECIR")

Fin_Si

Fin_Si

Fin_Si

Fin_Si

Fin_Escribe

Ejercicios

1. Hacer el DF y Pseudocódigo para que dado un carácter si es una letra mayúscula o minúscula y además coincide con la primera letra de los 7 días, entonces imprimir el nombre del día.

2. Hacer diagrama de flujo y pseudocodigo para calcular el sueldo de un empleado bajo las siguientes condiciones.
 - a. Si trabaja menos de 10 horas el pago por hora será de \$50.
 - b. Si trabaja 10 horas o mas pero menos de 40 el pago por hora será de \$ 80
 - c. Si trabaja 40 horas o mas su pago es de \$100

Estructuras de repetición

Definición

Repetir

- Ejecutar una instrucción o conjunto de instrucciones y luego regresar y ejecutarla (s) una y otra vez hasta terminar la tarea.



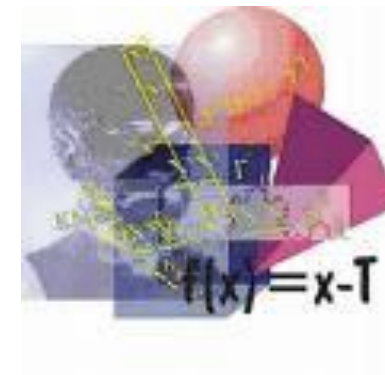
Aplicaciones

La **introducción** de muchos datos.

La **clasificación** periódica de una gran colección de datos.

Aplicaciones

- La **búsqueda** de un dato en una gran colección.
- **Fórmulas científicas** que sólo se pueden calcular por aproximaciones sucesivas.



Mecanismos de iteración

1. **Para** todos los valores de la progresión.
2. **Mientras** se valida una condición.
3. **Repetir- hasta** que se satisfaga una condición.

PARA

- Es usado *cuando se conoce de antemano*, el **número de veces** que debe repetirse una instrucción o conjunto de ellas.



PARA

- Los valores de la progresión deben ser asignados a una variable, la cual se denomina *variable de control*.



Variables contadoras

- Se usan para **contar**, por lo tanto deben ser de tipo **entero**.
- Ejemplo: Variable de control en un Ciclo Para.



Variables acumuladoras

- La función de una variable acumuladora es **almacenar** valores numéricos que generalmente se Suman (o multiplican) en cada iteración.
- La variable debe ser de tipo **entero o real**.

-
-



$sum \leftarrow sum + 1$

$sum \leftarrow 1$

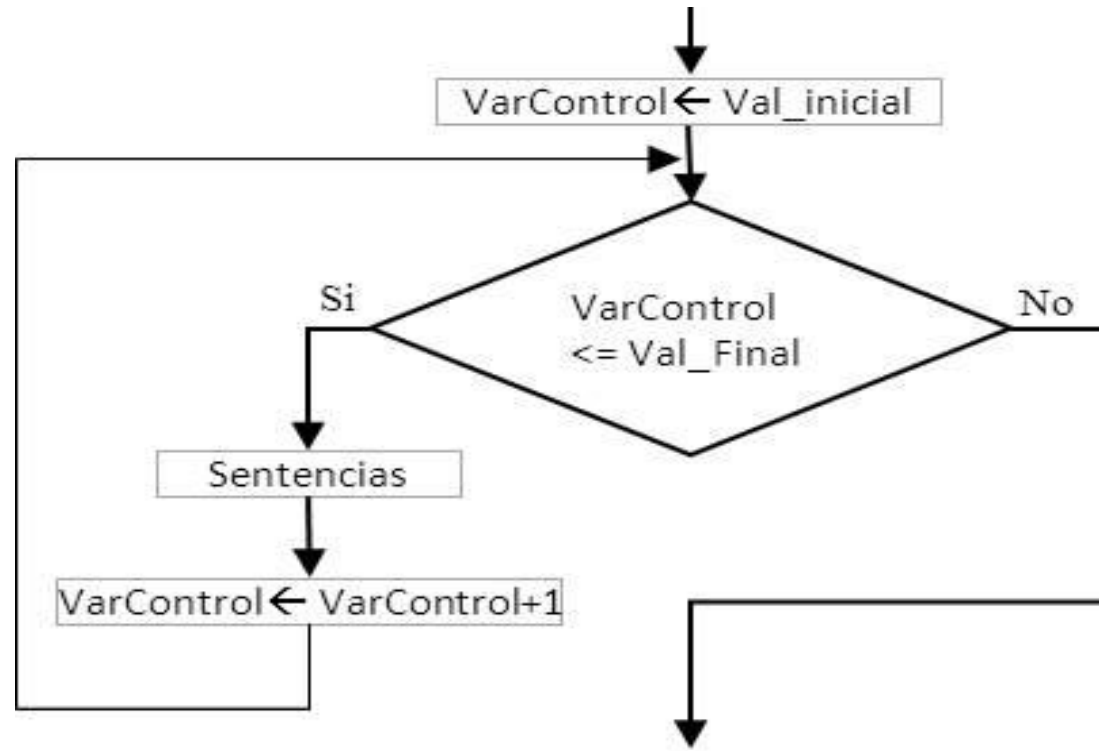


Variables bandera

- Utilizada dentro de la condición de un ciclo, para determinar cuándo un Ciclo se sigue iterando o cuando no.
- Debe ser de tipo **booleano o entero**.



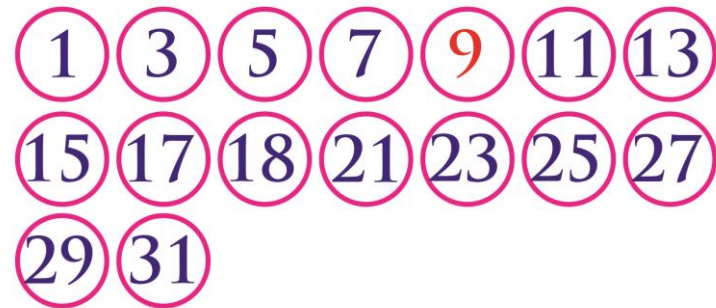
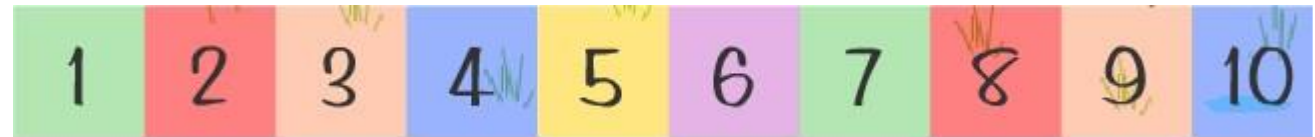
Para



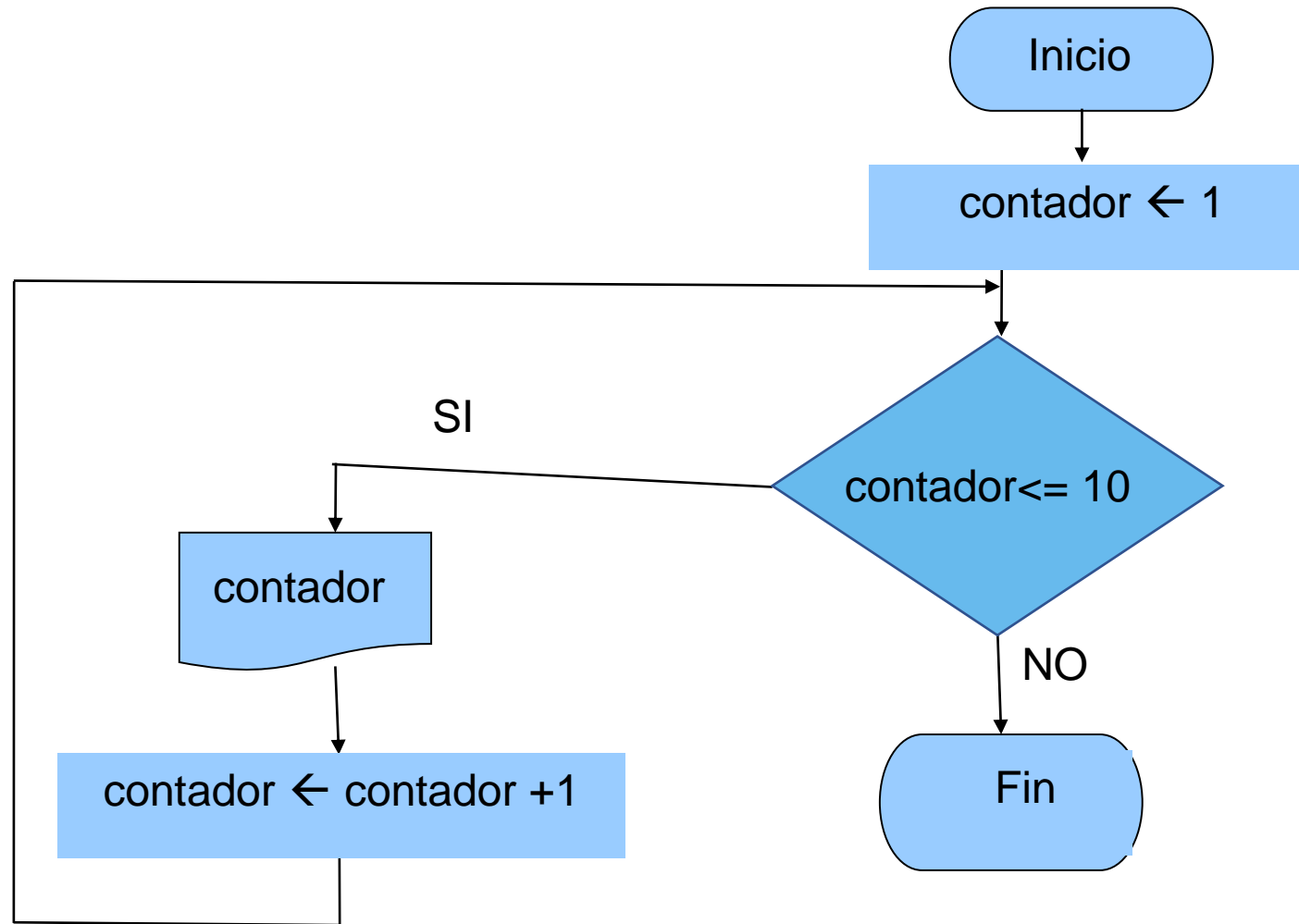
Pseudo código

Para VAR ← Expresion Hasta Expresión_FINAL Paso Valor
Sentencias

Finpara



Ejemplo: Imprime los primeros 10 números naturales



Pseudo código

Algoritmo: Num_1a10

Entrada:

Salida: contador

Inicio

Variables: contador: Entero

Para contador \leftarrow 1 Hasta 10 Paso1

Escribir(contador)

Fin_para

FinNum_1a10

Ejemplo

Algoritmo: promedio

Entrada: calif

Salida: prom

Inicio

Variables: calif, prom, sum \leftarrow 0.0 Real
cant Entero

Escribir("Dame 3 numeros")

Para(cant \leftarrow 1 Hasta 3 paso 1)

Leer(calif)

sum \leftarrow sum+calif

Fin_Para

prom \leftarrow sum/3

Fin_promedio

Traza

cant	calif	sum	prom
1	10	0.0	
		10.0	
2	8	18	
3	10	28	
4			9.3

Ejemplo

Algoritmo: descuento

Entrada: compra

Salida: pago

Inicio

Variables: compra, pago \leftarrow 0.0, desc Real

cant Entero

Para(cant \leftarrow 1 Hasta 3 paso 1)

Escribir("Dame la compra")

Leer(compra)

Si(compra > 5000) Entonces

desc \leftarrow compra * .20

pago \leftarrow compra - desc

Sino

Escribir("sigue participando")

FinSi

Escribir("lo que vas a pagar es de", pago)

FinPara

Findescuento

Traza

cant	calif	sum	prom
1	10	0.0	
		10.0	
2	8	18	
3	10	28	
4			9.3

Ejemplo

Algoritmo:cal_sueldo

Entrada:pago, hrs

Salida:sueldo

Inicio

Variables: hrs, sueldo, pago Real

emp Entero

Escribir("cuanto cuesta la hora")

Leer(pago)

Para emp \leftarrow 1 Hasta 4 Paso1

Escribir("Dame las hrs trabajadas")

Leer(hrs)

sueldo \leftarrow pago*hrs

Escribir("el sueldo es", sueldo)

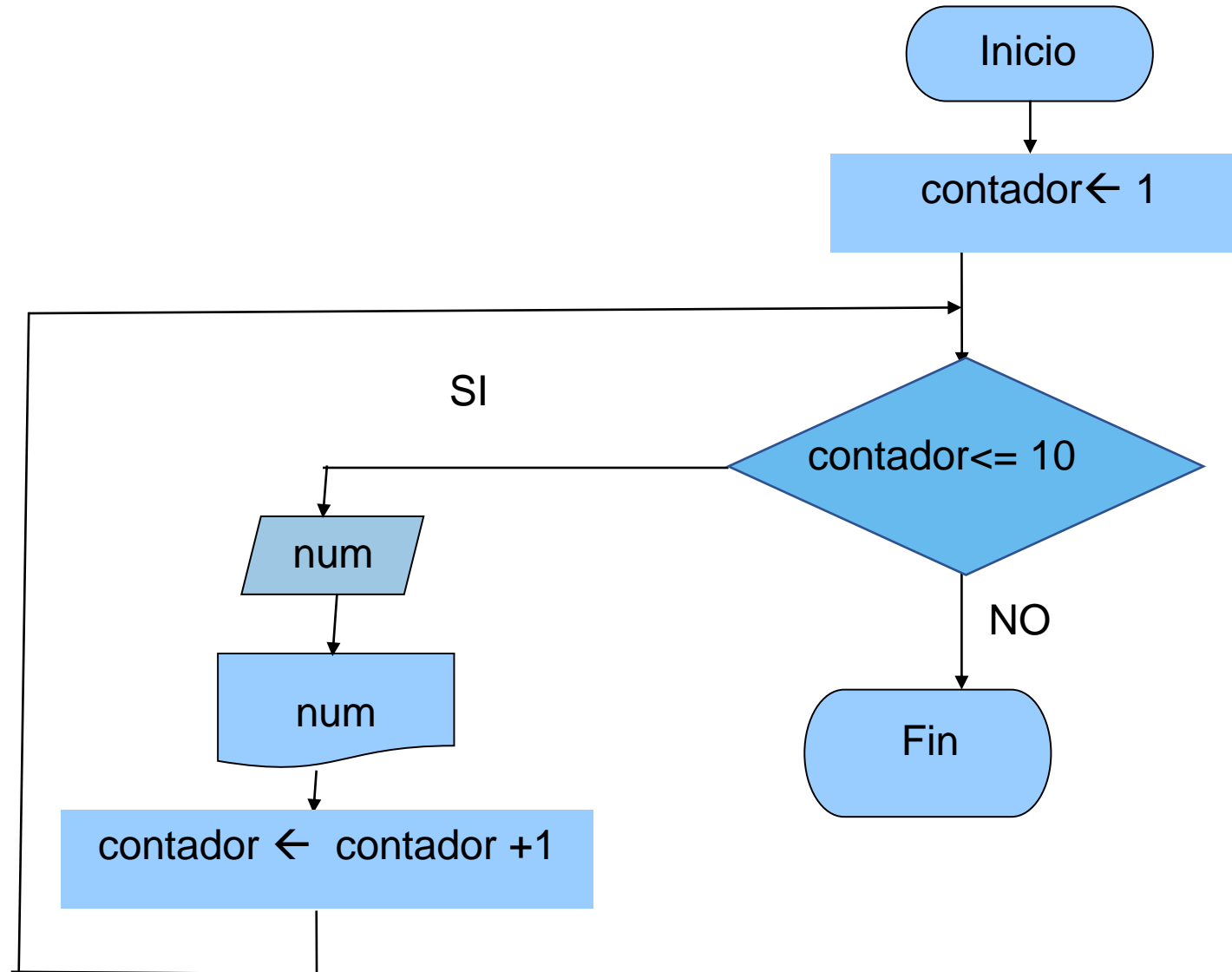
FinPara

Fincal_sueldo

Traza

con	num	sum
1	8	0X
		0+8
2	3	8+3
3	10	11+10

Ejemplo: Imprime 10 números enteros



Pseudo código

Algoritmo: dieznum

Entrada:num

Salida:num

Inicio

Variables: num: Entero

Para contador \leftarrow 1 Hasta 10 Paso 1

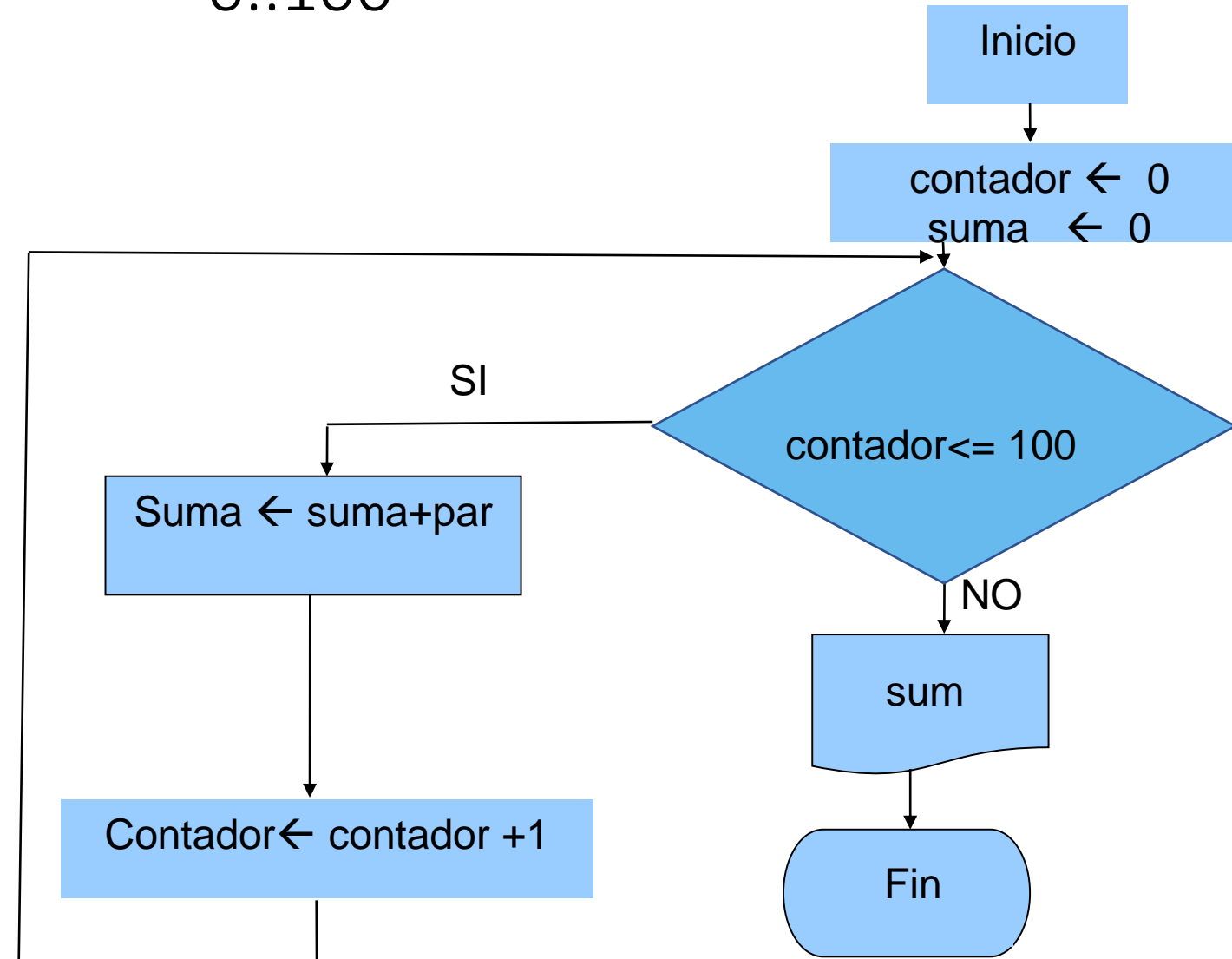
Leer(num)

Escribir(num)

Fin_Para

Findieznum

Ejemplo: Suma los números pares de 0..100



Pseudo código

- **Algoritmo:** num_1a100

- **Entrada:**

- **Salida:** suma

Inicio

Variables: suma, par:Entero

suma \leftarrow 0

Para par \leftarrow 0 Hasta 100 Paso 2

 suma \leftarrow suma + par

Fin_para

Escribir(suma)

Finnum_1a100

Ejercicio

Diseña el DF y pseudo código para los siguientes ejercicios



Nota: Utiliza la estructura para

1. Generar los primeros n números pares.
2. Generar los primeros n números primos.
3. Obtener el factorial de un número N.
4. Obtener los primeros N números de la serie de Fibonacci.
5. Sumar los cuadrados de los primeros N números naturales.
6. Dado n calcular La sumatoria

$$\sum_{i=1}^n i^2 * (n / 2)$$

Mientras

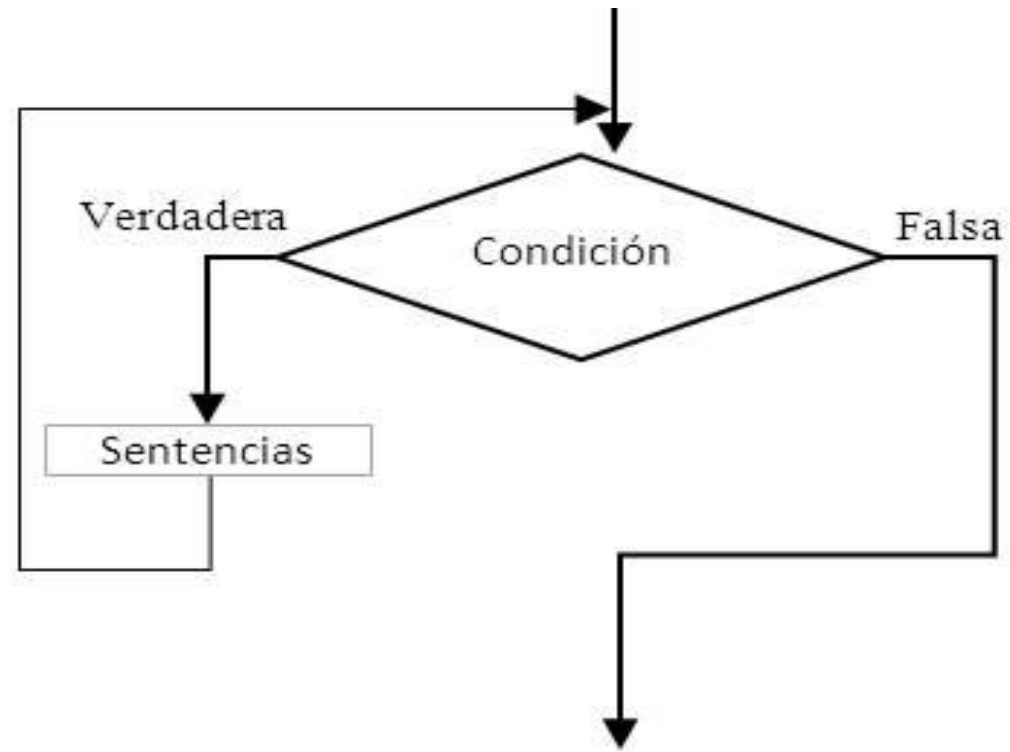
- Ejecuta un bloque de instrucciones, **mientras** una expresión lógica dada se cumpla.



Mientras

- Si la **condición se cumple**, el bloque se **ejecuta**, después, la condición se vuelve a evaluar.
- Si la condición se evalúa la primera vez como **falsa**, el bloque de instrucciones **no será ejecutado**.

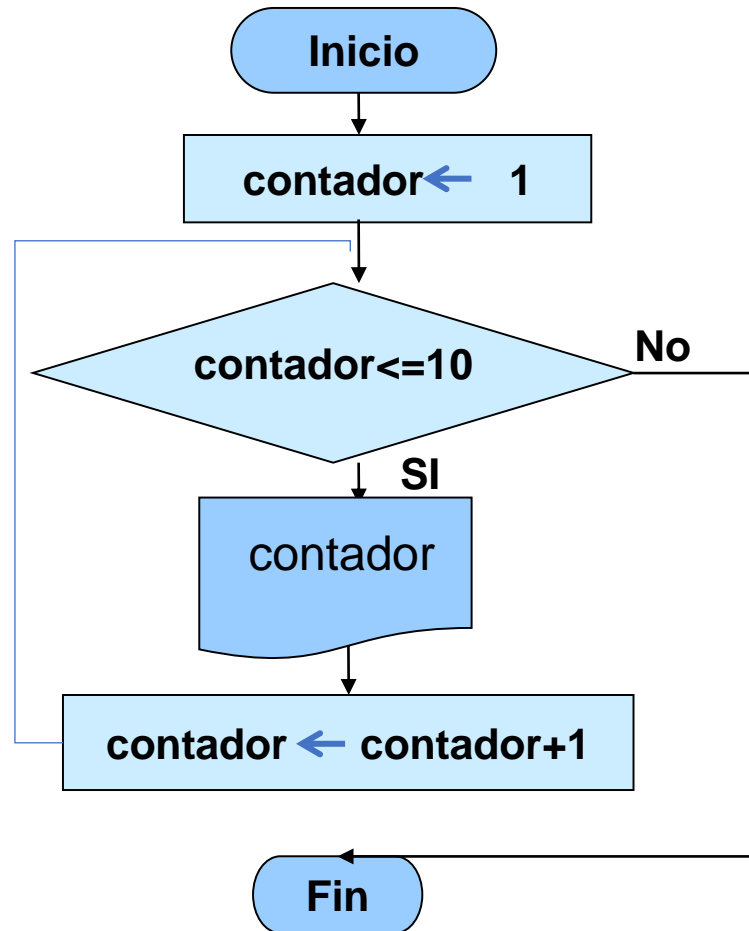
Diagrama de Flujo



Pseudo código

```
Mientras (expresión_lógica)  
    bloque_de_instrucciones  
Fin_mientras
```

Ejemplo: Escribe los primeros 10 números enteros



Algoritmo: Serie_1a10

Entrada:

Salida: contador

Inicio

Variables: contador: Entero

calif ← -8

Mientras (calif ≤ 0) or (calif > 10)

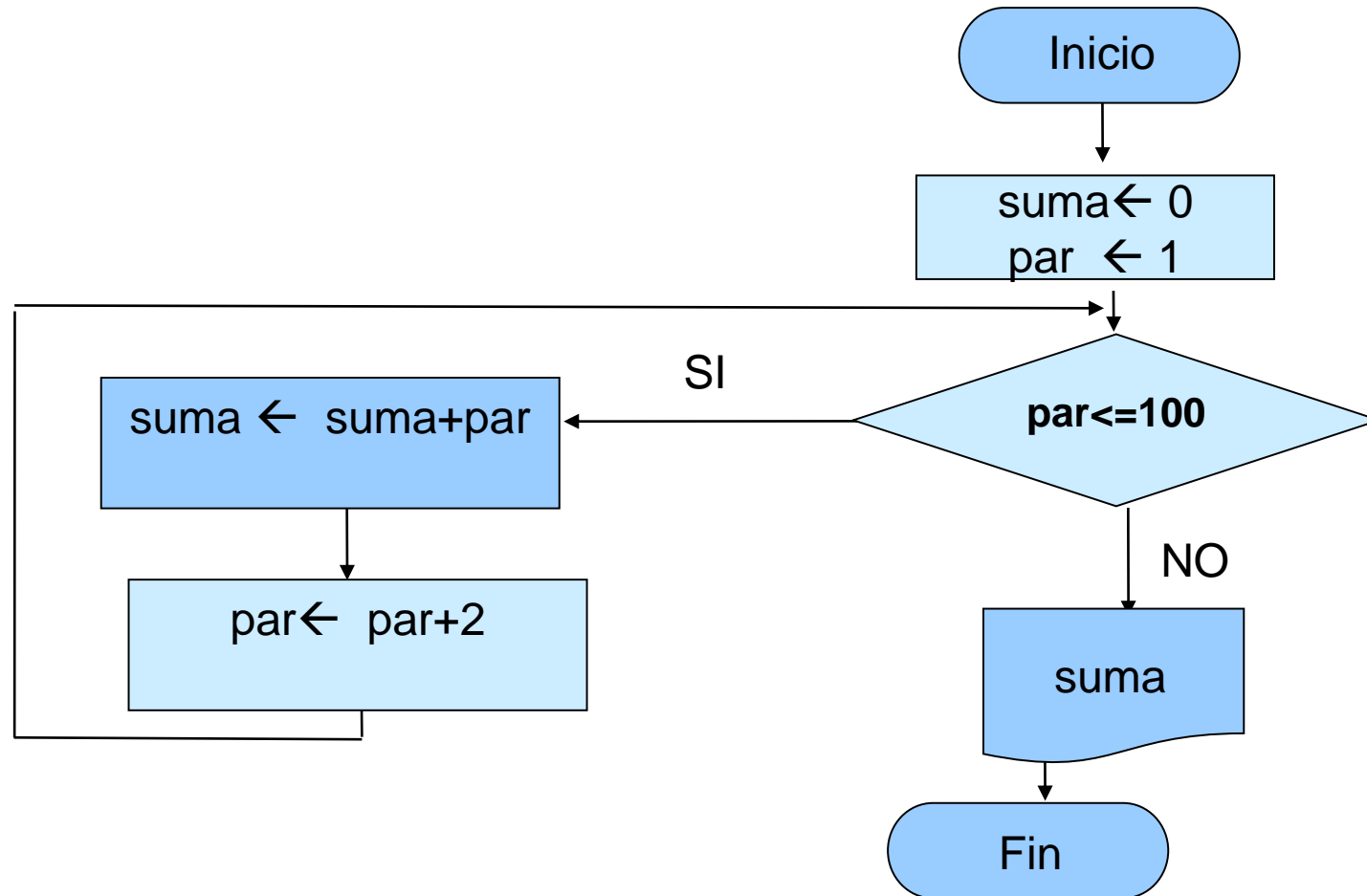
Ler(calif) 9

Fin_mientras

F

FinSerie_1a10

Ejemplo: Suma los números pares de 0..100



Pseudo código

Algoritmo: Num_1a100

Entrada:

Salida: suma

Inicio

Variables: par,suma:Entero

par \leftarrow 0 Para (par \leftarrow 1)

suma \leftarrow 0

Mientras (par \leq 100) Hasta 100

 suma \leftarrow suma + par

 par \leftarrow par+2 Paso 2

Fin_mientras

Escribir(suma)

FinNum_1a100

Ejercicios

- De forma individual analiza los siguientes algoritmos e indica que hace cada uno.
- Debes identificar las tres partes para cada ciclo de repetición:
 - Inicio
 - Actualización de la variable de control
 - Fin(condicional)

Ejercicio 1: Escribe los primeros 10 enteros

Algoritmo: ciclo

Entrada:

Salida: contador

Inicio

Variables: contador:Entero

```
contador ← 10
Mientras ( contador <= 10 )
    Escribir( contador )
    contador ← contador - 1
Fin_mientras
```

Finciclo

Ejercicio 2: Escribe los primeros 10 enteros

Algoritmo: imprime

Entrada:

Salida: contador

Inicio

Variables: contador: Entero

contador \leftarrow 1

Mientras (contador \geq 1)

 Escribir(contador)

 contador \leftarrow contador - 1

Fin_mientras

Finimprime

Ejercicio 3: Escribe los primeros 10 enteros

Algoritmo: ciclo

Entrada:

Salida: contador

Inicio

Variables: contador:Entero

contador \leftarrow 10

Mientras (contador \geq 1)
 Escribir(contador)

Fin_mientras

Finciclo

Ejercicio

Diseña el Diagrama de flujo y Pseudo código para los siguientes ejercicios.

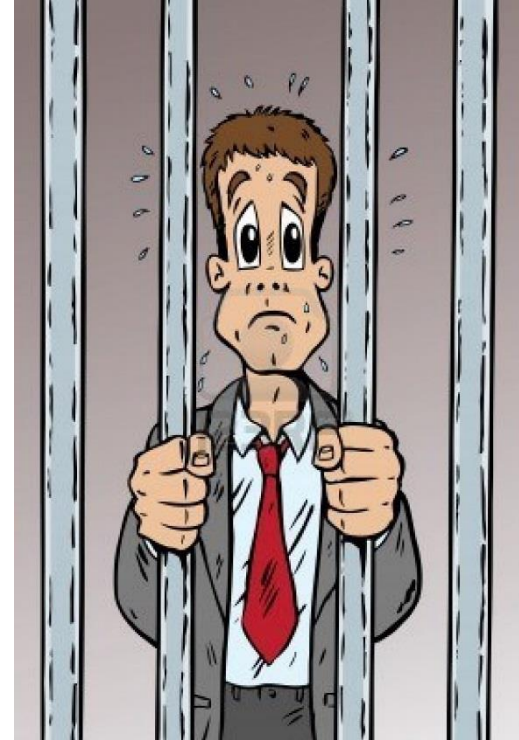


Usar Mientras

- Determinar si un número dado X es múltiplo de otro número dado Y .
- Determinar los dos números pares más próximos entre los que se encuentra un número entero X dado.
- Leer N números y obtener el promedio solo de los números pares de la lista.
- Introducir un conjunto de M números, determinar la cantidad de números positivos y negativos del conjunto.
- La sumatoria $\sum_{j=0}^k \frac{k^j}{k} * j$

Repetir-Hasta

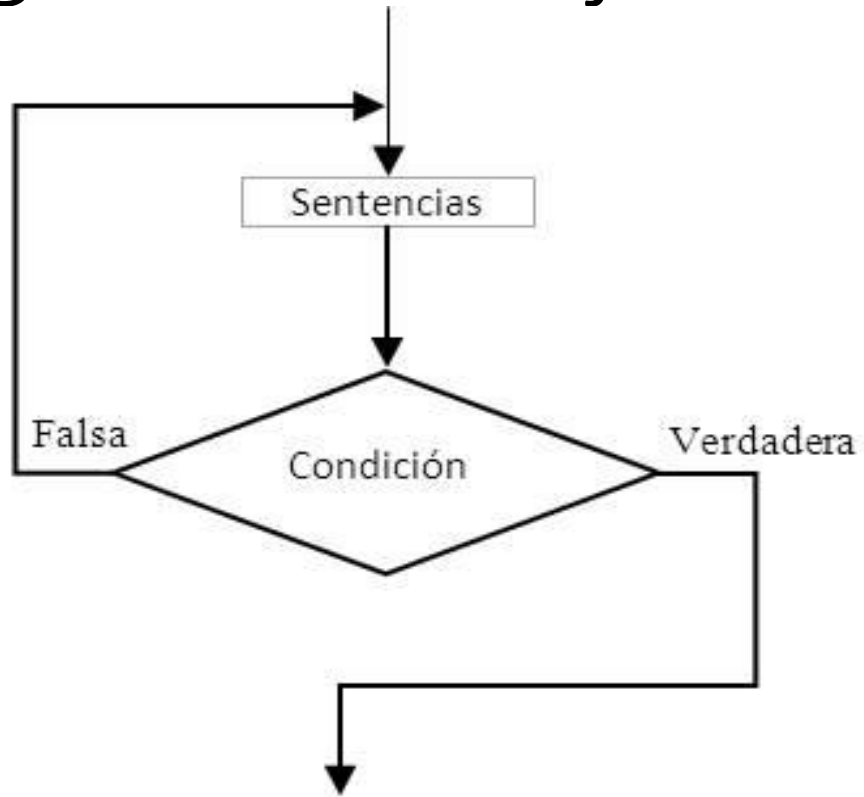
- Se **ejecuta** el bloque de instrucciones y después se **evalúa** la condición.



Repetir-hasta

- Si es verdadera, se vuelve a ejecutar el bloque de instrucciones.
- Y así sucesivamente, hasta que, la condición sea falsa.

Diagrama de Flujo



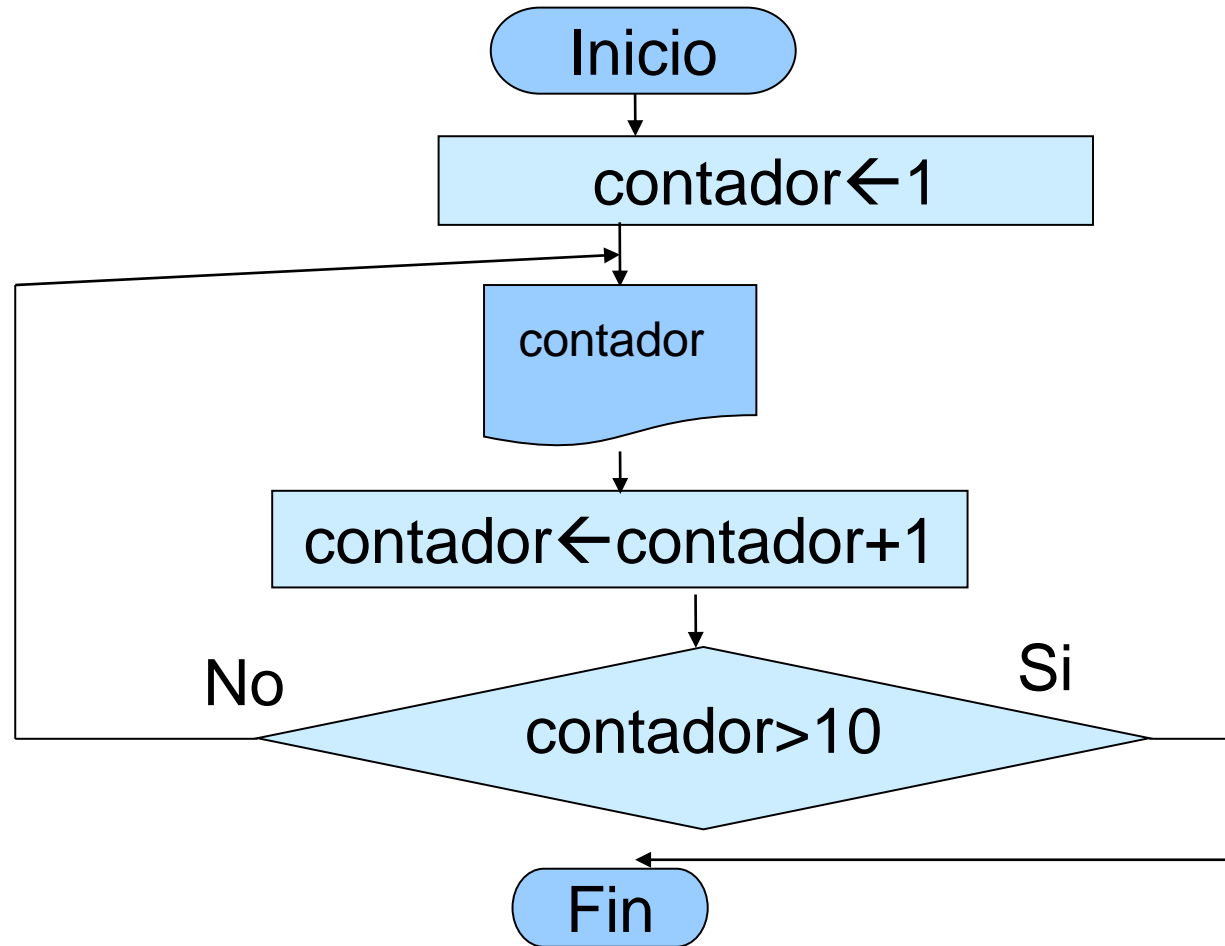
Pseudo código

Repetir

bloque_de_instrucciones

Hasta(expresión_lógica)

Ejemplo: Escribe los primeros 10 números enteros



Pseudo código

Algoritmo: Para

Entrada: caract

Salida: con_vo

Inicio

Variables: caract Carácter

con_vo ← 0, cont Entero

Para (cont ← 1 Hasta 10 Paso 1)

Escribir("Dame un caracter")

Leer(caract)

Según(caract) Sea

Caso 'a'

Caso 'e'

Caso 'i'

Caso 'o'

Caso 'u': con_voc ← con_voc + 1

FinSegun

FinPara

Escribir("Las vocales son ", con_voc)

Algoritmo: Para

Entrada: caract

Salida: con_vo

Inicio

Variables: caract Carácter

con_vo ← 0 Entero

cont ← -1

Mientras(cont ≤ 10)

Escribir("Dame un caracter")

Leer(caract)

Según(caract) Sea

Caso 'a'

Caso 'e'

Caso 'i'

Caso 'o'

Caso 'u': con_voc ← con_voc + 1

FinSegun

cont ← cont + 1

Fin_Mientras

Algoritmo: Para

Entrada: caract

Salida: con_vo

Inicio

Variables: caract Carácter

con_vo ← 0 Entero

cont ← 1

Repetir

Escribir("Dame un caracter")

Leer(caract)

Según(caract) Sea

Caso 'a'

Caso 'e'

Caso 'i'

Caso 'o'

Caso 'u': con_voc ← con_voc + 1

FinSegun

cont ← cont + 1

Hasta (cont > 10)

Ejercicios

En equipo de 4 diseña el Diagrama de flujo y Pseudo código para los siguientes ejercicios



1. Obtener el máximo y el mínimo de n números.
2. Realizar la suma de dos números con incrementos de 1.
3. Realizar la resta de dos números con decrementos de 1
4. Realizar la multiplicación de dos números con sumas sucesivas.
5. Realizar la división de dos números con restas sucesivas.
6. La sumatoria $\sum_{x=1}^n x + x^n$

Bibliografía

1. Cairó O (2005). Metodología de la programación, Algoritmos, diagramas de flujo y programas (3ª ed). México: Alfaomega.
2. Joyanes, A. (2008). Fundamentos de programación, Algoritmos, Estructuras de datos y Objetos (4ª. ed). España: Mc Graw Hill.
3. Skiena, S (2008). The algorithm design Manual (2nd ed). USA: Springer.

Responsables del Curso

De la Rosa Flores Rafael

Moyao Martínez Yolanda

Sánchez Román Guillermina

Es responsabilidad exclusiva de los autores el respeto de los derechos de autor sobre los contenidos e imágenes en el presente documento, en consecuencia, la **BUAP** no se hace responsable por el uso no autorizado, errores, omisiones o manipulaciones de los derechos de autor y estos serán atribuidos directamente al **Responsable de Contenidos**, así como los efectos legales y éticos correspondientes.

gracias.

BUAP ©2020

Es **responsabilidad exclusiva de los autores** el respeto de los derechos de autor sobre los contenidos e imágenes en el presente documento, en consecuencia, la **BUAP** no se hace responsable por el uso no autorizado, errores, omisiones o manipulaciones de los derechos de autor y estos serán atribuidos directamente al **Responsable de Contenidos, así como los efectos legales y éticos correspondientes.**