

GA3-220501093-AA2-EV01 fundamentos de programación estructurada y estructuras cíclicas

GA3-220501093-AA2-EV02 foro temático: Estructuras de programación

GA3-220501093-AA2-EV03 taller aplicando funciones y procedimientos en la solución de algoritmos

Yeisson Mauricio Torres Patiño

Ficha: 2977422

Instructor

William Cortés

Tecnólogo en análisis y desarrollo de software

Centro de la construcción regional valle, servicio nacional de aprendizaje

27 de noviembre de 2024

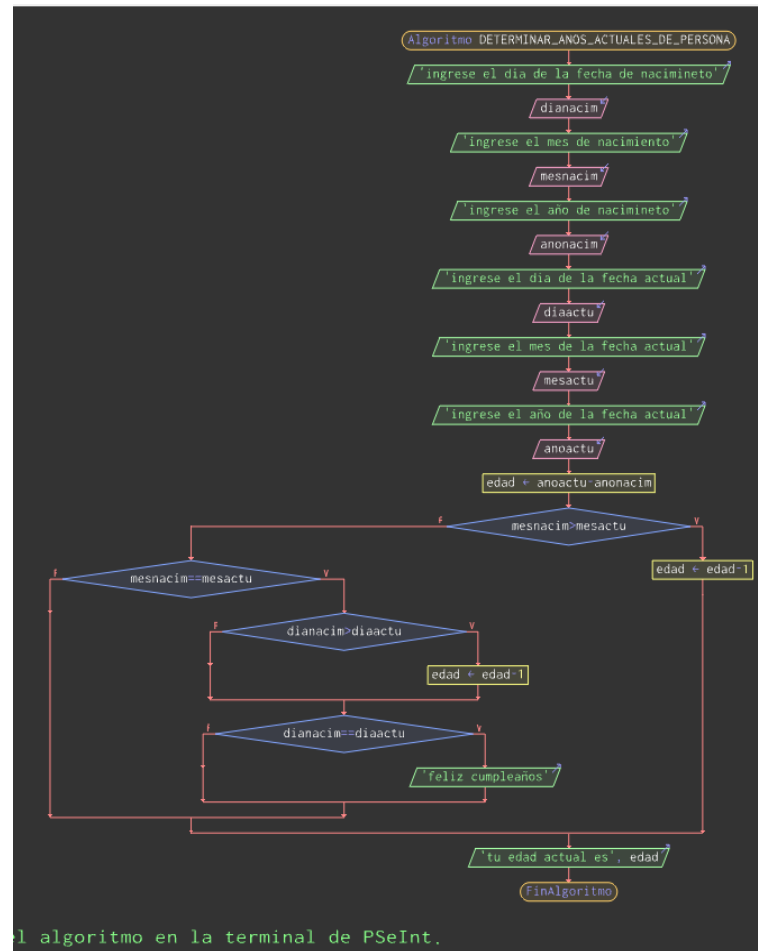
GA3-220501093-AA2-EV01 fundamentos de programación estructurada y estructuras cíclicas

Algoritmo que a partir de una fecha de nacimiento y una fecha actual determine la edad en años actual de una persona.

Op. Algebraicos	
+	(suma)
-	(resta)
*	(multiplicación)
/	(división)
↑	(potencia)
MOD	(resto/modulo)
Op. Lógicos	
Y	(conjunción)
O	(disyunción)
NO	(negación)
Op. Relacionales	
=	(igual)
≠	(distinto)
<	(menor)
≤	(menor o igual)
>	(mayor)
≥	(mayor o igual)
Func. Matemáticas	
abs	(valor absoluto)
trunc	(valor truncado)
redon	(valor redondeado)
raiz	(raíz cuadrada)
sen	(seno)
cos	(coseno)
tan	(tangente)
asen	(arcoseno)
acos	(arcocoseno)
atan	(arcotangente)
ln	(logaritmo natural)
exp	(func. exponencial)

```

1  Algoritmo Determinar_años_actuales_de_persona
2  escribir "ingrese el día de la fecha de nacimiento"
3  leer dianacim
4  Escribir "ingrese el mes de nacimiento"
5  leer mesnacim
6  Escribir "ingrese el año de nacimiento"
7  leer anonacim
8  Escribir "ingrese el día de la fecha actual"
9  leer diaactu
10 Escribir "ingrese el mes de la fecha actual"
11 leer mesactu
12 Escribir "ingrese el año de la fecha actual"
13 leer anoactu
14 edad ← anonacim - dianacim
15 si mesnacim > mesactu entonces
16     edad ← edad - 1
17 sino
18     si mesnacim == mesactu Entonces
19         si dianacim > diaactu Entonces
20             edad ← edad - 1
21         FinSi
22     si dianacim == diaactu Entonces
23         Escribir "feliz cumpleaños"
24     FinSi
25 FinSi
26 Fin Si
27 Escribir "tu edad actual es" edad
28
29
30 FinAlgoritmo
31
  
```



El algoritmo en la terminal de PSeInt.

```

PSeInt - Ejecutando proceso DETERMINAR_AÑOS_ACTUALES_DE_PE...

*** Ejecución Iniciada. ***
ingrese el día de la fecha de nacimiento
> 02
ingrese el mes de nacimiento
> 09
ingrese el año de nacimiento
> 1990
ingrese el día de la fecha actual
> 26
ingrese el mes de la fecha actual
> 10
ingrese el año de la fecha actual
> 2024
tu edad actual es34
*** Ejecución Finalizada. ***
  
```

Algoritmo que permita determinar si un año indicado es o no un año bisiesto.

Operadores y Funciones

Op. Algebraicos

+ (suma)

- (resta)

* (multiplicación)

/ (división)

↑ (potencia)

MOD (resto/modulo)

Op. Lógicos

Y (conjunción)

O (disyunción)

NO (negación)

Op. Relacionales

= (igual)

≠ (distinto)

< (menor)

≤ (menor o igual)

```

1  Algoritmo es_o_no_año_bisnieto
2  Escribir "ingresa"
3  leer año
4  si (año mod 4 = 0 Y año mod 100 ≠ 0) Entonces
5      escribir "el año" año "es bisnieto"
6  SiNo
7      si (año mod 400 = 0) Entonces
8          Escribir "el año" año "es bisnieto"
9      SiNo
10         Escribir "el año" año "no es bisnieto"
11     FinSi
12 Fin si
13 FinAlgoritmo
14

```

PSelnt - Ejecutando proceso ES_O_NO_ANO_BISNIETO

```

*** Ejecución Iniciada. ***
ingresa
> 2023
el año2023no es bisnieto
*** Ejecución Finalizada. ***

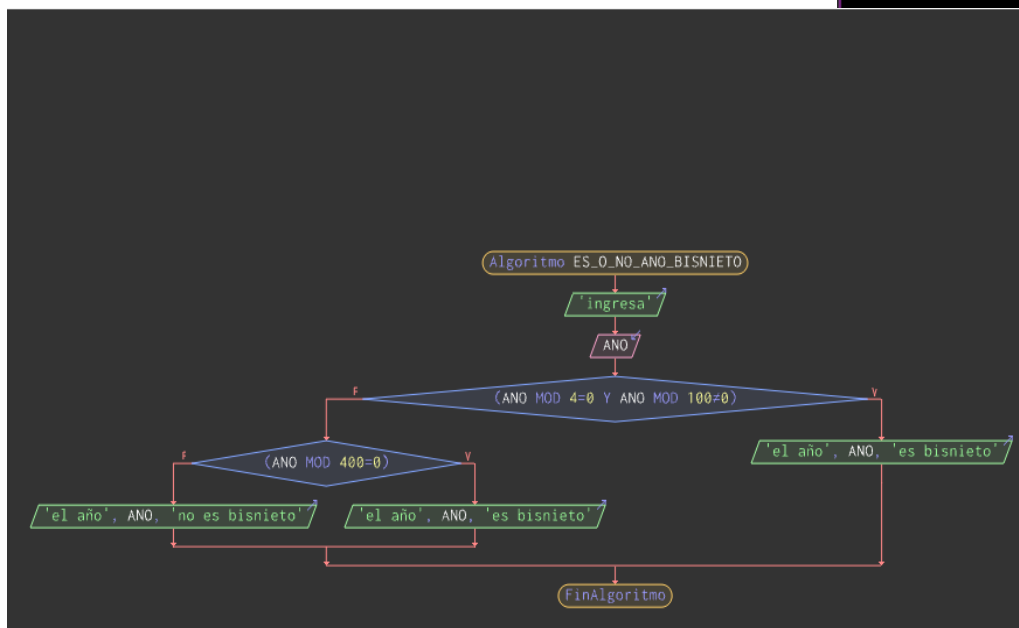
```

PSelnt - Ejecutando proceso ES_O_NO_ANO_BISNIETO

```

*** Ejecución Iniciada. ***
ingresa
> 2024
el año2024es bisnieto
*** Ejecución Finalizada. ***

```



GA3-220501093-AA2-EV02 foro temático: Estructuras de programación

✚ ¿Qué beneficios tiene una representación en comparación con la otra?

Diagrama de flujo	Pseudocódigo
Es una representación gráfica que utiliza símbolos establecidos de fácil comprensión.	Ocupa menos espacio en el desarrollo del problema.
Su estructura es fácil de organizar.	Permite tener una visión más amplia acerca de las operaciones del programa en cuestión.
Al igual que el pseudocódigo, el diagrama de flujo con fines de análisis de algoritmos de programación puede ser ejecutado en un ordenador.	Permite la resolución de cualquier problema que pueda tener el programa antes de ser programado.
Facilitan la comprensión de problemas complejos, al mostrarlo como un dibujo.	Resulta muy fácil pasar del pseudocódigo al lenguaje de programación.

✚ ¿Respecto a las herramientas existentes, cuál tiene más representación?

En mi opinión, dado los conocimientos adquiridos durante el proceso de diagramar, me parece que los diagramas de flujo tienen más representación, dado que estos son una representación gráfica que utiliza símbolos establecidos de fácil comprensión.

✚ ¿Cuál sería la más fácil para entender las estructuras algorítmicas?

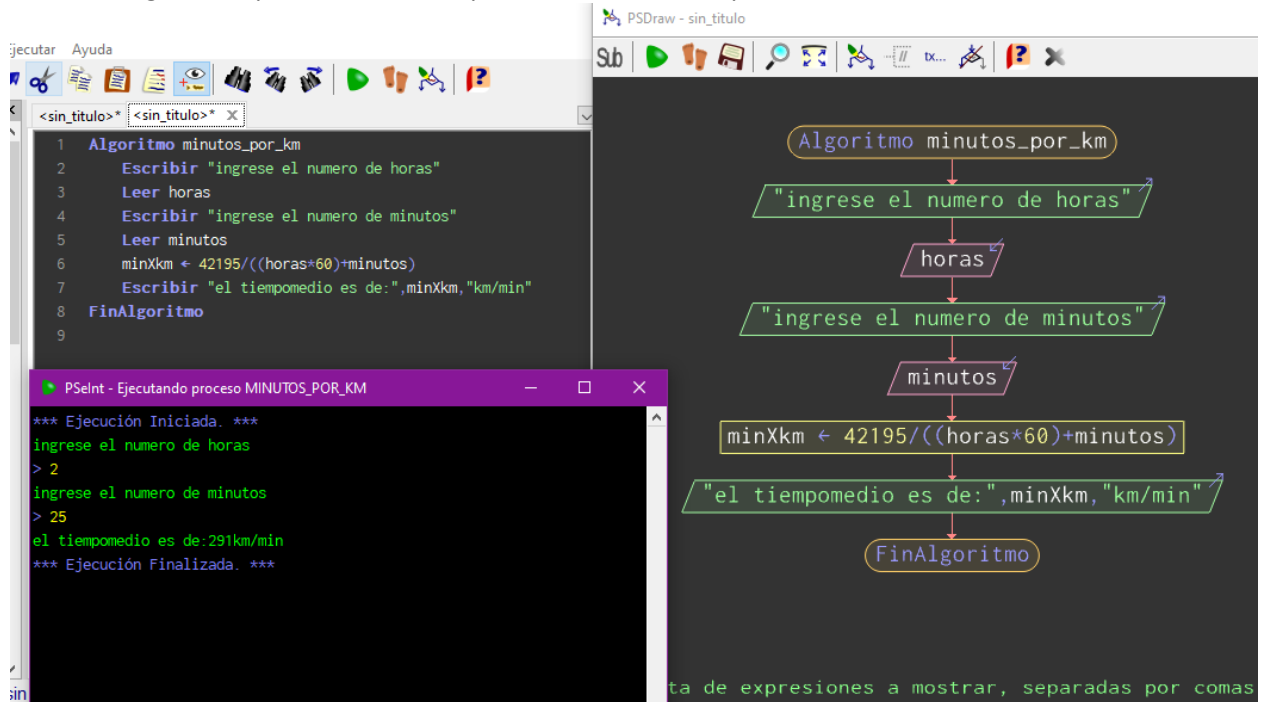
Mi elección sigue siendo el diagrama de flujo ya que dan una presentación mucho más ordenada y de mejor entendimiento.

✚ ¿Cuál permitirá una transición más fácil hacia la codificación?

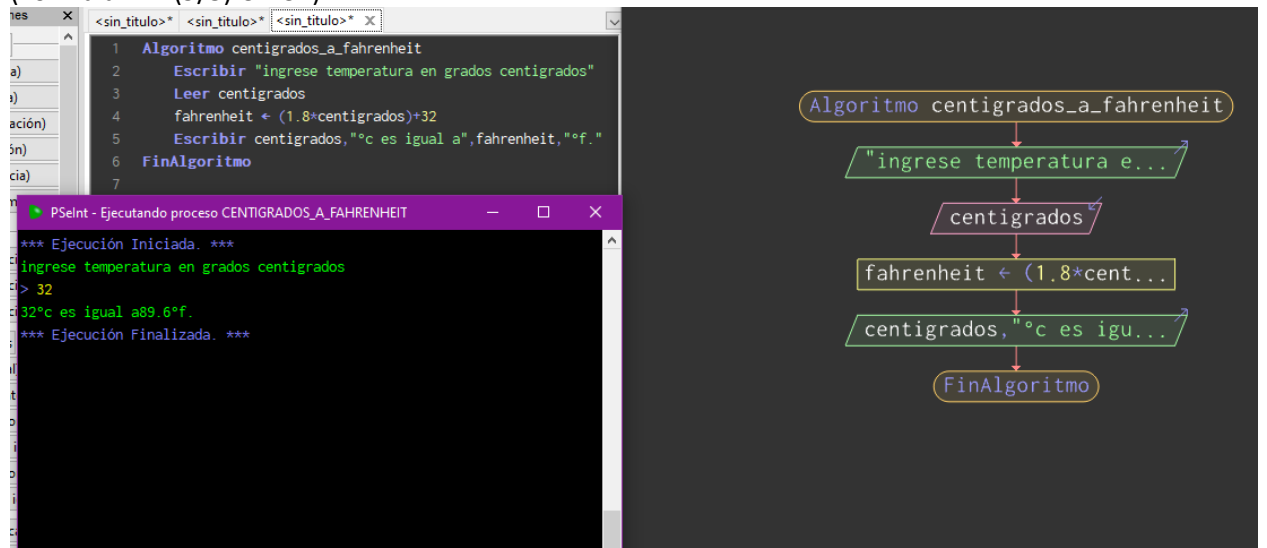
El pseudocódigo, ya que es una versión simplificada de los conceptos de algunos lenguajes de programación. Se escribe en algún lenguaje natural como español o inglés y funciona para que se represente de una manera entendible el algoritmo para un programa.

GA3-220501093-AA2-EV03 taller aplicando funciones y procedimientos en la solución de algoritmos

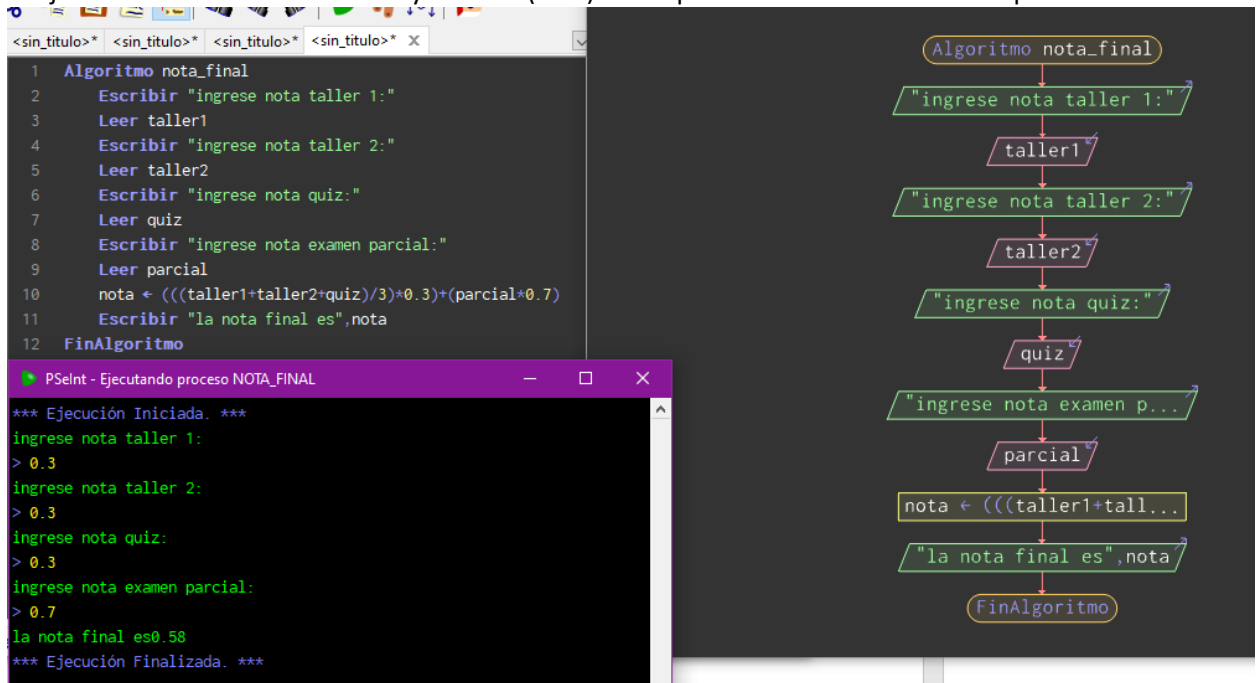
1. Un corredor de maratón (distancia 42,195 Km) ha recorrido la carrera en 2 horas 25 minutos. Se desea un algoritmo que calcule el tiempo medio en minutos por kilómetro.



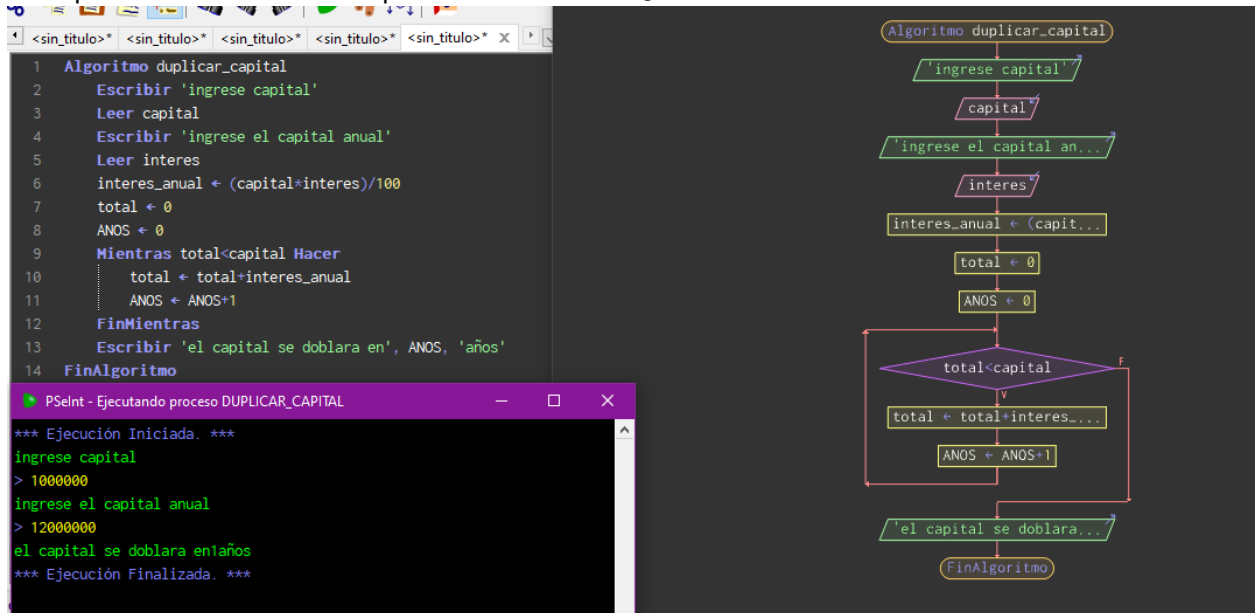
2. Realizar la conversión de una temperatura dada en grados Centígrados a grados Fahrenheit (Fórmula: $F = (9/5) C + 32$).



3. Escribir el algoritmo que permite calcular la nota correspondiente al primer parcial de “análisis” para un estudiante cualquiera. Se debe considerar que hay dos talleres y un quiz, que en conjunto valen un 30% de la nota y el resto (70%) corresponde a la nota del examen parcial.



4. Un capital C está situado a un tipo de interés R anual ¿al término de cuántos años se doblará?



5. Elaborar un algoritmo que permita ingresar 20 números y muestre todos los números menores e iguales a 25.

```

1  Algoritmo menores_iguales_25
2  contador ← 1
3  Dimensionar menores(20)
4  Para i←1 Hasta 20 Con Paso 1 Hacer
5      Escribir 'ingrese un numero:'
6      Leer ingresado
7      Si ingresados<25 Entonces
8          menores[contador] ← ingresado
9          contador ← contador+1
10     FinSi
11 FinPara
12 Escribir 'los numeros menores o iguales a 25 son:'
13 Para j←1 Hasta (20) Con Paso 1 Hacer
14     Si menores[j]>0 Entonces
15         Escribir menores[j]
16     FinSi
17 FinPara
18 FinAlgoritmo
    
```

PSelnt - Ejecutando proceso MENORES_IGUALES_25

los numeros menores o iguales a 25 son:

10

5

12

60

40

12

17

18

70

20


22

56

Algoritmo menores_iguales_25

```

graph TD
    Start([Inicio]) --> Contador[contador ← 1]
    Contador --> Dimensionar[Dimensionar menores(20)]
    Dimensionar --> Ingrese[ingrese un numero:]
    Ingrese --> Ingresado[/Ingresado/]
    Ingresado --> Condicion1{ingresados<25}
    Condicion1 -- V --> Asignar[menores[contador] ← in...]
    Asignar --> Incrementar[contador ← contador+1]
    Incrementar --> Condicion1
    Condicion1 -- F --> Condicion2{menores[j]>0}
    Condicion2 -- V --> Escribir[Escribir menores[j]]
    Escribir --> Condicion2
    Condicion2 -- F --> FinAlgoritmo([FinAlgoritmo])
    
```



6. Hacer un programa que sume 5 precios de camisas (en dólares) y que luego muestre el total de la venta en pesos.

```

1  Algoritmo total_dolares_a_pesos
2  total ← 0
3  Para i←1 Hasta 5 Con Paso 1 Hacer
4      Escribir 'ingrese el valor de la camisa', i
5      Leer camisa
6      total ← total+camisa
7  FinPara
8  Escribir 'el total de la venta es:', (total*4365), 'pesos colombianos'
9  FinAlgoritmo
    
```

PSelnt - Ejecutando proceso TOTAL_DOLARES_A_PESOS

*** Ejecución Iniciada. ***

ingrese el valor de la camisa1

> 4365

ingrese el valor de la camisa2

> 4365

ingrese el valor de la camisa3

> 4365

ingrese el valor de la camisa4

> 4365

ingrese el valor de la camisa5

> 4365

el total de la venta es:95266125pesos colombianos

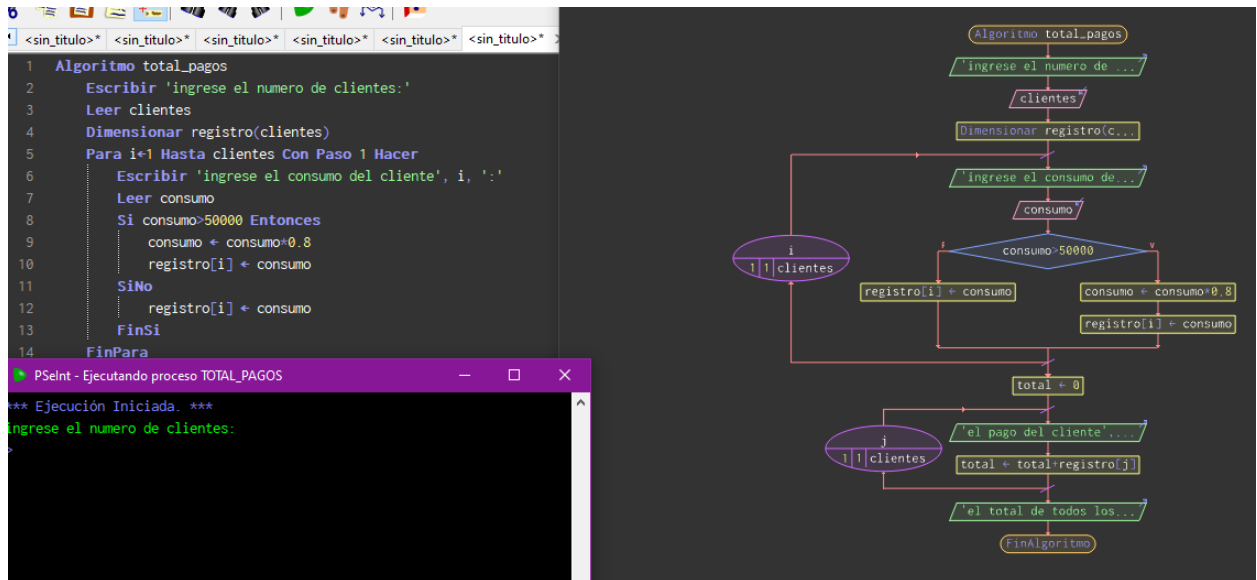
*** Ejecución Finalizada. ***

Algoritmo total_dolares_a_pesos

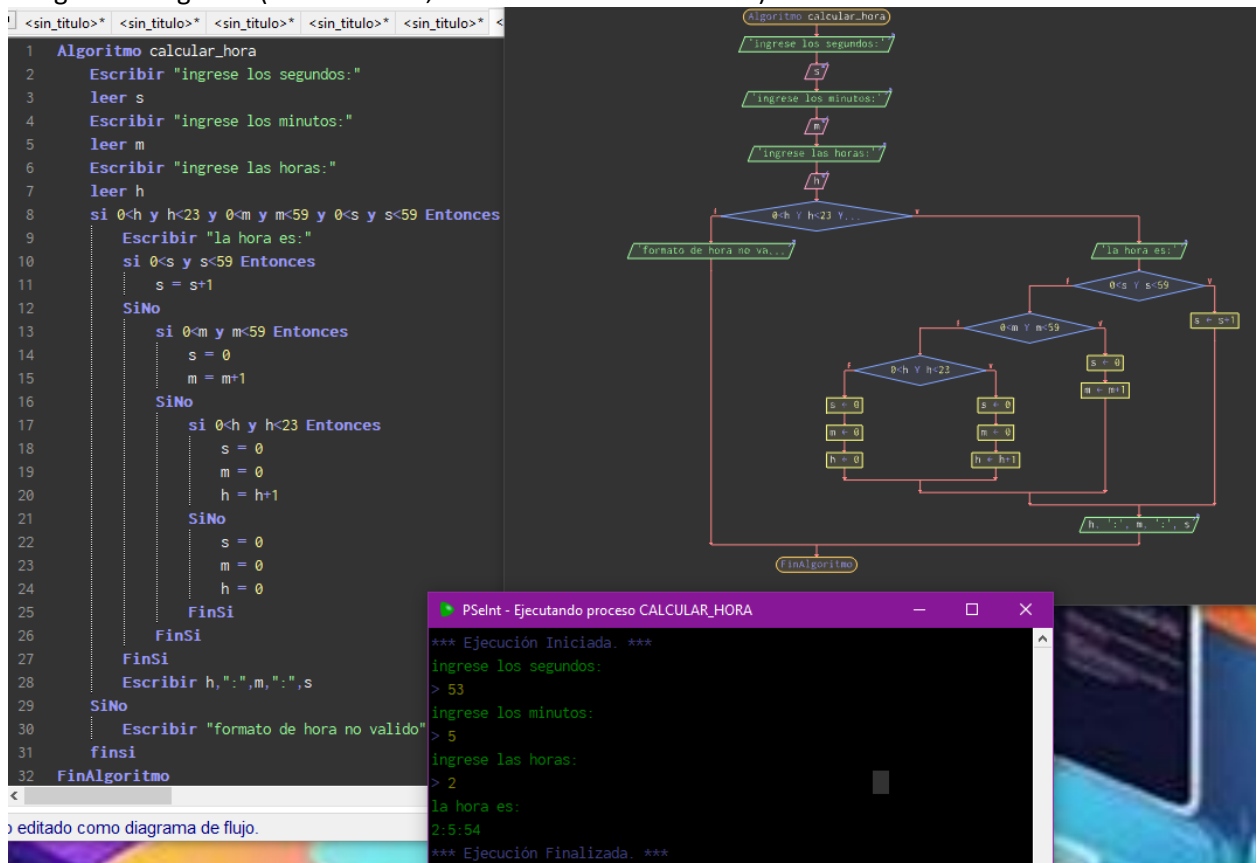
```

graph TD
    Start([Inicio]) --> Total[total ← 0]
    Total --> Ingrese[ingrese el valor de l...]
    Ingrese --> Camisa[/camisa/]
    Camisa --> Sumar[total ← total+camisa]
    Sumar --> Condicion{total<4365}
    Condicion -- V --> Ingrese
    Condicion -- F --> FinAlgoritmo([FinAlgoritmo])
    
```

7. Hacer un programa que registre el consumo realizado por los clientes de un restaurante, si el consumo de cada cliente excede 50000 se hará un descuento del 20%. Se debe mostrar el pago de cada cliente y el total de todos los pagos.



8. Diseñar un algoritmo que permita ingresar la hora, minutos y segundos, y que calcule la hora en el siguiente segundo ("0<= H <=23", "0<= M <=59" "0<= S<=59").



9. Dado N, escribir el producto desde 1 hasta N.

```

1  Algoritmo producto_hasta_n
2    Escribir 'ingrese un numero:'
3    Leer num
4    total ← 1
5    Para i←1 Hasta num Con Paso 1 Hacer
6      total ← total*i
7    FinPara
8    Escribir 'el producto desde 1 hasta', num, 'es de:', total
9    FinAlgoritmo

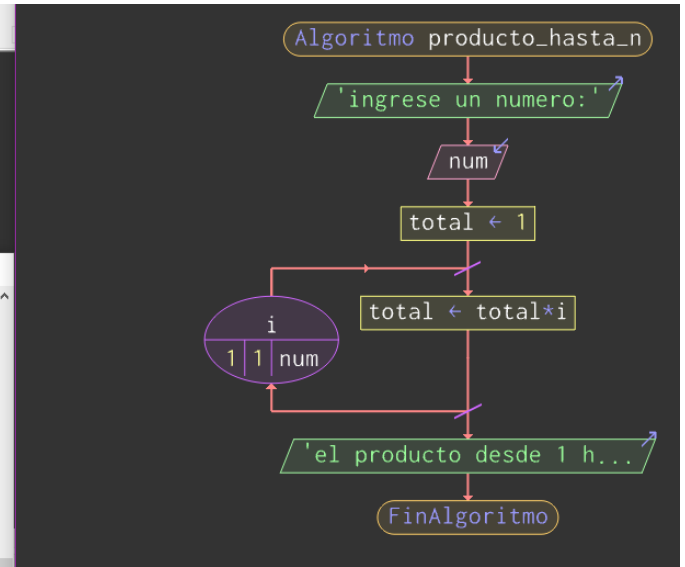
```

PSeInt - Ejecutando proceso PRODUCTO_HASTA_N

```

*** Ejecución Iniciada. ***
ingrese un numero:
> 10
el producto desde 1 hasta10es de:3628800
*** Ejecución Finalizada. ***

```



10. Realizar un algoritmo que muestre por pantalla la tabla de multiplicar decreciente de cualquier número, ingresado entre el 1 y el 10.

```

1  Algoritmo tabla_multiplicar
2    Escribir 'ingrese un numero del 1 al 10:'
3    Leer entrada
4    contador ← 10
5    Para i←1 Hasta 10 Con Paso 1 Hacer
6      Escribir entrada, ' * ', contador, ' = ', (entrada*contador)
7      contador ← contador-1
8    FinPara
9    FinAlgoritmo

```

PSeInt - Ejecutando proceso TABLA_MULTIPLICAR

```

ingrese un numero del 1 al 10:
> 8
8 * 10 = 80
8 * 9 = 72
8 * 8 = 64
8 * 7 = 56
8 * 6 = 48
8 * 5 = 40
8 * 4 = 32
8 * 3 = 24
8 * 2 = 16
8 * 1 = 8
*** Ejecución Finalizada. ***

```

