



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS
AMBIENTALES

Kit de Programación

iAprende Programación de Forma Interactiva!

Autores: Burbano España Johan Alexander

Cabrera Ruano Jefferson Alexander

Fecha: 27/06/2024

Índice

1. Introducción	3
2. Descripción.....	4
2.1 Caja Principal (Tablero)	4
2.2 Datos.....	4
2.3 Variables, Operadores y Funciones Matemáticas.....	4
2.4 Tipos de Datos.....	5
2.5 Pseudocódigo y Diagrama de Flujo	5
3. EJERCICIOS	9
3.1 Funciones Matemáticas	9
3.2 Operadores Lógicos	15
3.3 Operadores Relacionales.....	19
3.4 Operadores Algebraicos	25
3.5 Ejercicios con 'SI'	33
3.6 Ejercicios con 'SEGUN'	40
3.7 Ejercicios con 'REPETIR'	46
3.8 Ejercicios con 'MIENTRAS'	52
3.9 Ejercicios con 'PARA'.....	61

1. Introducción

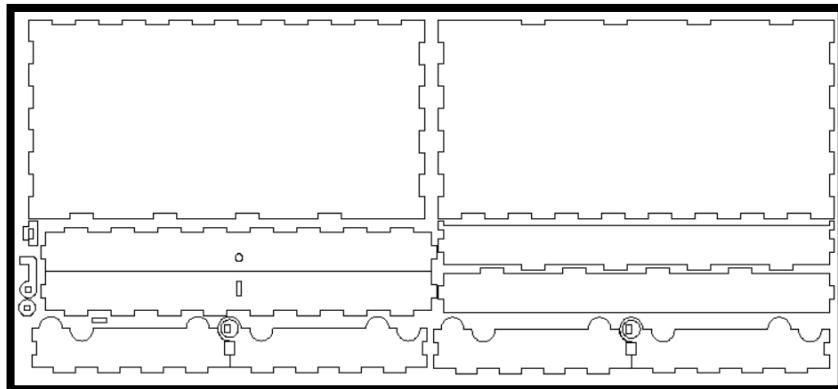
El fin de este curso es capacitar a los participantes en los fundamentos de la programación, proporcionándoles las habilidades y conocimientos necesarios para comprender y crear algoritmos simples, desarrollar programas básicos en pseudocódigo utilizando PSeInt, y resolver problemas de manera sistemática y lógica.

Al finalizar el curso, los participantes serán capaces de diseñar, codificar y ejecutar programas simples, así como de aplicar los conceptos aprendidos para abordar desafíos de programación en su entorno académico o profesional

2. Descripción

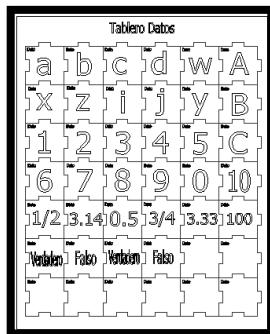
2.1 Caja Principal (Tablero)

Un tablero con forma de baúl en donde se almacenan las fichas del material didáctico del Kit de Programación.

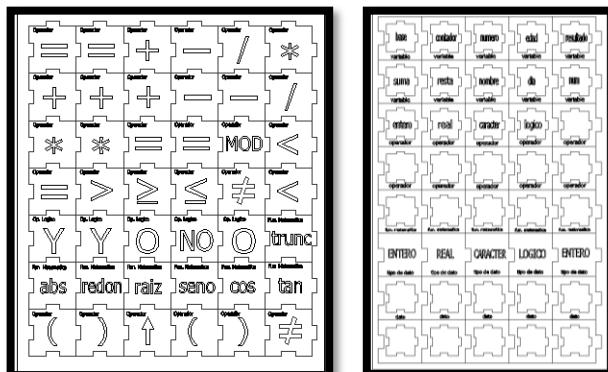


2.2 Datos

Estas fichas representan datos numéricos tanto de tipo entero como decimales y fraccionarios. También de expresar datos de tipo carácter como letras o palabras. Además de fichas en blanco para escribir el dato que quiera el usuario expresar.



2.3 Variables, Operadores y Funciones Matemáticas



2.3.1 Variables

Una variable es un elemento de datos con nombre cuyo valor puede cambiar durante el curso de la ejecución de un programa.

2.3.2 Operadores Algebraicos

Los operadores matemáticos definen las operaciones básicas que actúan sobre los números y otras construcciones matemáticas.

2.3.3 Operadores Lógicos

Los operadores lógicos se usan para comparar dos valores y devolver un resultado verdadero, falso o nulo.

2.3.4 Funciones Matemáticas

Una función matemática es una relación que se establece entre dos conjuntos, a través de la cual a cada elemento del primer conjunto se le asigna un único elemento del segundo conjunto o ninguno.

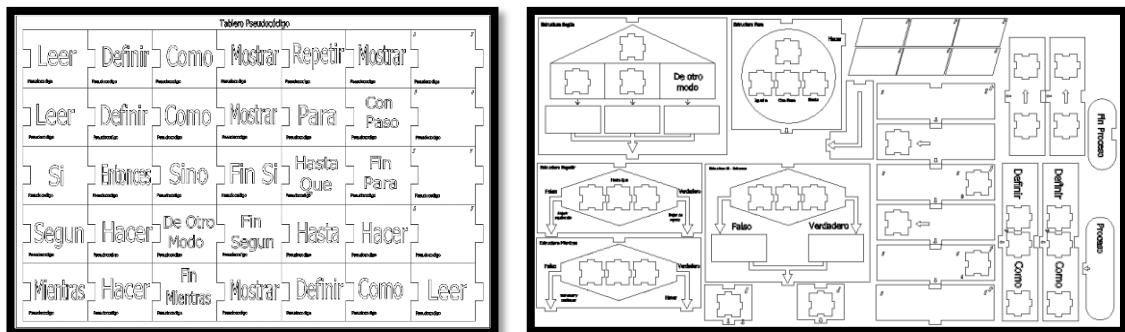
2.4 Tipos de Datos

Se conoce como tipo de datos a aquello que puede contener una variable y cómo esta almacena los datos.

2.5 Pseudocódigo y Diagrama de Flujo

El pseudocódigo es una forma de representar código, como algoritmos, funciones y otros procesos, utilizando una combinación de lenguaje natural y elementos similares al lenguaje de programación.

En el diagrama de flujo, utilizamos símbolos y flechas para representar cada paso del proceso.



2.5.1 Inicio y Final

Es la forma en la cual se indica el “inicio del proceso” y “final del proceso”. Su icono suele ser un rectángulo con las esquinas redondeadas.

Pseudocódigo	Diagrama de Flujo
Proceso 'I'	Un rectángulo redondeado que contiene la palabra "Proceso" y una flecha apuntando hacia abajo.
Fin Proceso	Un rectángulo redondeado que contiene la frase "Fin Proceso".

2.5.2 Asignación o Definir

Permite asignar un valor o definir una variable, donde se almacenan los datos que se necesitan para resolver el problema. Suele representarse con un rectángulo.

Pseudocódigo	Diagrama de Flujo
Un rectángulo dividido en dos secciones, ambas etiquetadas como "Pseudocódigo". La sección superior contiene la frase "Definir Como".	Un rectángulo que contiene la frase "Definir Como". Una flecha apunta de este rectángulo a un cuadro vacío debajo de él.

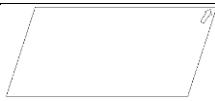
2.5.3 Datos de Entrada

Indica la recepción de datos en la entrada. Se representa con un recuadro con las esquinas inclinadas y una flecha hacia dentro.

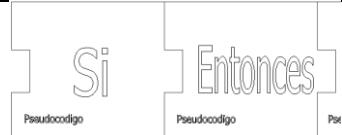
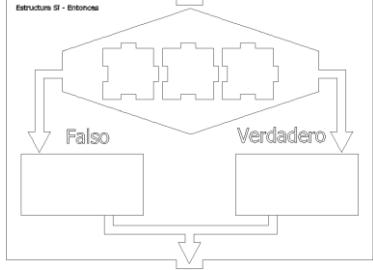
Pseudocódigo	Diagrama de Flujo
Un rectángulo dividido en dos secciones, ambas etiquetadas como "Pseudocódigo". La sección superior contiene la palabra "Leer".	Un cuadro con esquinas inclinadas que tiene una flecha apuntando hacia su interior.

2.5.4 Datos de Salida

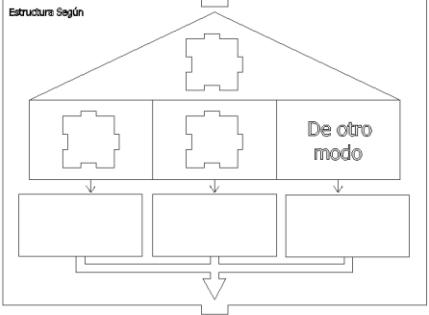
Indica la impresión de datos en la salida. Se representa con un recuadro con las esquinas inclinadas y una flecha hacia fuera.

Pseudocódigo	Diagrama de Flujo
	
	
	

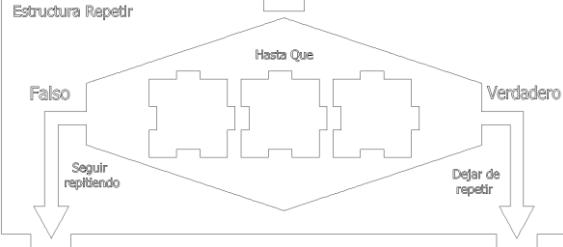
2.5.5 Estructura de Control - Si

Pseudocódigo	Diagrama de Flujo
 Pseudocódigo Pseudocódigo Pst	
 Pseudocódigo Pseudocódigo	

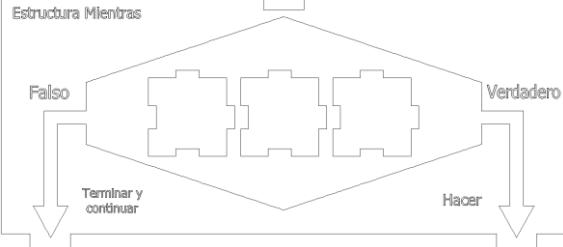
2.5.6 Estructura de Control - Segun

Pseudocódigo	Diagrama de Flujo
 Pseudocódigo Pseudocódigo	
 Pseudocódigo Pseudocódigo	

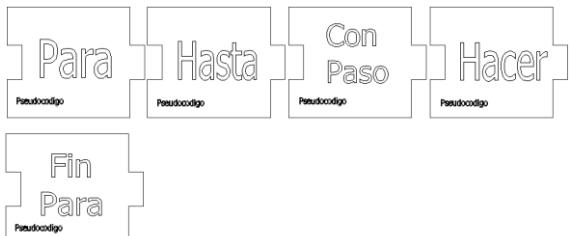
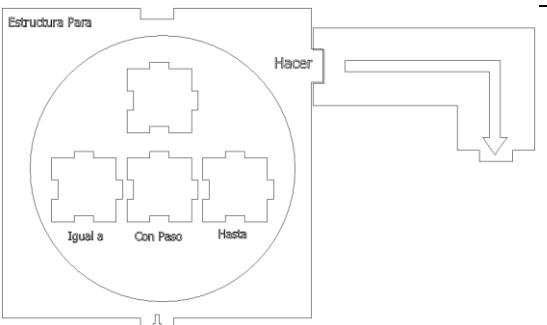
2.5.7 Estructura de Control - Repetir

Pseudocódigo	Diagrama de Flujo
 <pre> Repetir Pseudocódigo Hasta Que Pseudocódigo </pre>	<p>Estructura Repetir</p>  <pre> graph TD Start(()) --> Cond{ } Cond -- Falso --> Seguir[Seguir repitiendo] Seguir --> Body[] Body --> Cond Cond -- Verdadero --> Dejar[Dejar de repetir] Dejar --> End(()) </pre>

2.5.8 Estructura de Control - Mientras

Pseudocódigo	Diagrama de Flujo
 <pre> Mientras Hacer Pseudocódigo Pseudocódigo Fin Mientras Pseudocódigo </pre>	<p>Estructura Mientras</p>  <pre> graph TD Start(()) --> Cond{ } Cond -- Falso --> Terminar[Terminar y continuar] Terminar --> End(()) Cond -- Verdadero --> Hacer[Hacer] Hacer --> Body[] Body --> Cond </pre>

2.5.9 Estructura de Control - Para

Pseudocódigo	Diagrama de Flujo
 <pre> Para Hasta Con Paso Hacer Pseudocódigo Pseudocódigo Pseudocódigo Pseudocódigo Fin Para Pseudocódigo </pre>	<p>Estructura Para</p>  <pre> graph TD Start(()) --> Cond{ } Cond -- Igual a --> Body[] Body --> Cond Cond -- Con Paso --> Hacer[] Cond -- Hasta --> End(()) Hacer --> Body </pre>

3. EJERCICIOS

3.1 Funciones Matemáticas

3.1.1 Calcular la raíz cuadrada de un número

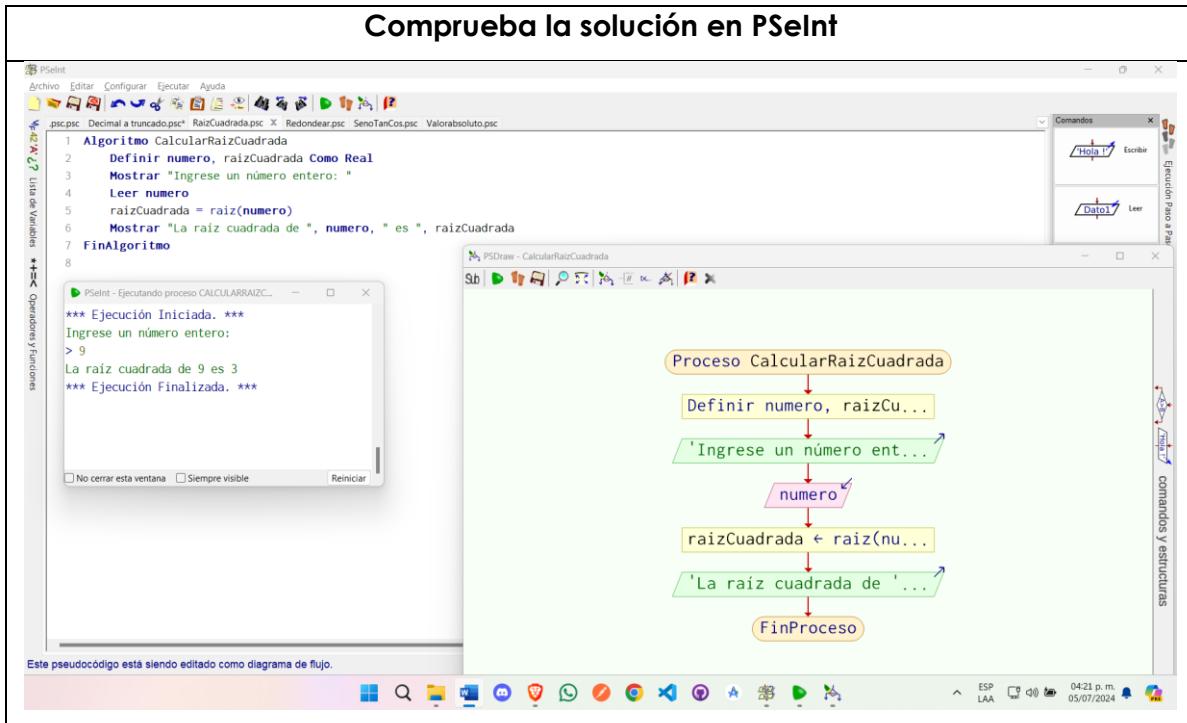
Escribe un programa que lea un número entero y calcule su raíz cuadrada utilizando la función 'raiz' de PSeInt.

Pasos a Seguir

1. Dar un nombre a este proceso: 'CalcularRaizCuadrada'
2. Defina dos tipos de variables de tipo 'Real'
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese un número entero: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso de un valor al usuario.
5. Asigne a la variable definida el resultado de la función matemática '**raiz**'.
6. Utilice 'Mostrar' para mostrar un mensaje con el número ingresado y el resultado de la función matemática.
7. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Proceso CalcularRaizCuadrada Definir num como Real Mostrar "Ingrese un numero entero" Leer num num = abs(num) num = raiz(num) Mostrar "El valor absoluto es: " + num Fin Proceso </pre>	<pre> Proceso CalcularRaizCuadrada Definir num como Real Entrada: "Ingrese un numero entero" num = abs(num) num = raiz(num) Salida: "El valor absoluto es: " + num Fin Proceso </pre>

Comprueba la solución en PSeInt



The screenshot shows the PSeInt environment. On the left, the pseudocode editor displays:

```

1 Algoritmo CalcularRaizCuadrada
2   Definir numero, raizCuadrada Como Real
3   Mostrar "Ingrese un número entero: "
4   Leer numero
5   raizCuadrada = raiz(numero)
6   Mostrar "La raíz cuadrada de ", numero, " es ", raizCuadrada
7 FinAlgoritmo
  
```

The execution window shows the output of running the pseudocode with input 9, displaying the result as 3.

On the right, the flowchart editor shows the corresponding flowchart:

```

graph TD
    Start([Proceso CalcularRaizCuadrada]) --> Def[Definir numero, raizCuadrada]
    Def --> Inp["'Ingrese un número entero:'"]
    Inp --> Num[/numero/]
    Num --> Raiz["raizCuadrada ← raiz(numero)"]
    Raiz --> Res["'La raíz cuadrada de ' numero ' es ' raizCuadrada"]
    Res --> End([FinProceso])
  
```

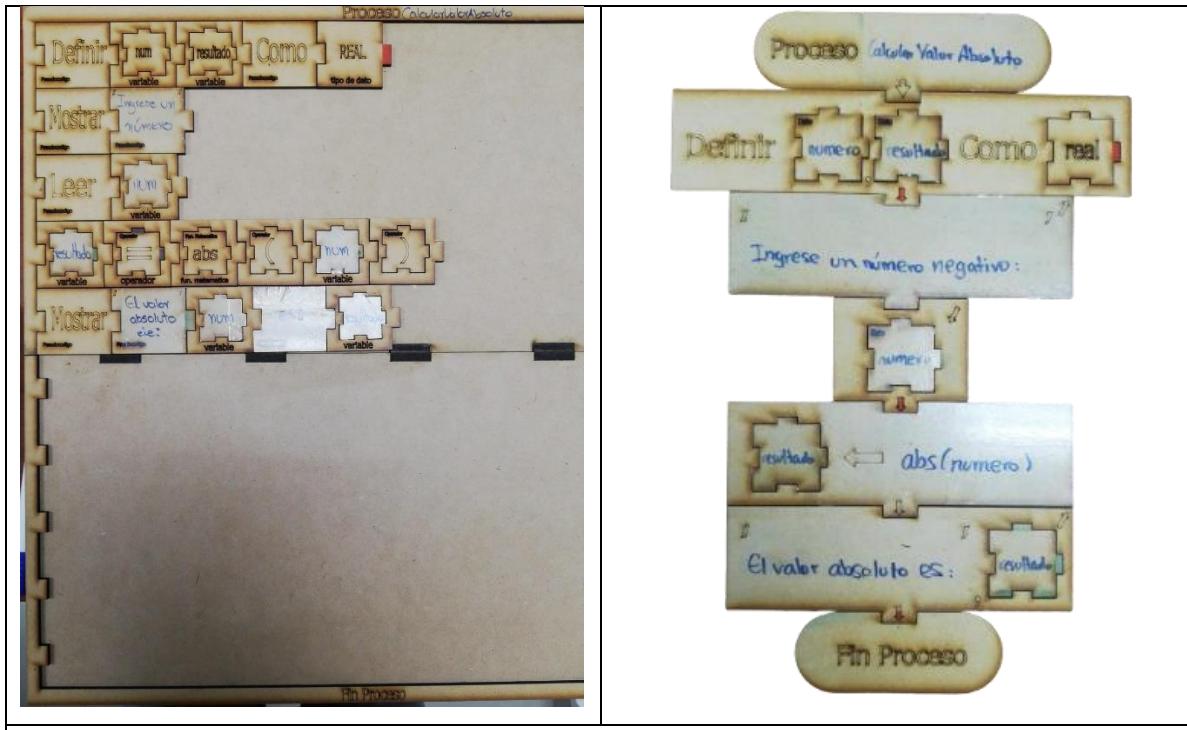
3.1.2 Cálculo del valor absoluto

Crea un programa que pida al usuario un número entero y muestre su valor absoluto usando la función 'abs'.

Pasos a Seguir

1. Dar un nombre a este proceso: 'CalcularValorAbsoluto'
2. Defina dos tipos de variables de tipo 'Real'
3. Utilice 'Mostrar para mostrar texto al usuario: 'Ingrese un número negativo: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso de un valor al usuario.
5. Asigne a la variable definida el resultado de la función matemática '**abs**'.
6. Utilice 'Mostrar' para mostrar un mensaje con el número ingresado y el resultado de la función matemática.
7. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
--------------	-------------------



Comprueba la solución en PSeint

The screenshot shows the PSeint interface with the following details:

- Execution Window:** Shows the pseudocode for 'CalcularValorAbsoluto' and its execution results. The output shows the program asking for a negative number, receiving '-89', calculating its absolute value as 89, and then exiting.
- Code Editor:** Displays the pseudocode for the algorithm:

```

1 Algoritmo CalcularValorAbsoluto
2   Definir numero, valorAbsoluto Como Real
3   Mostrar "Ingrese un número negativo: "
4   Leer numero
5   valorAbsoluto = abs(numero)
6   Mostrar "El valor absoluto de ", numero, " es ", valorAbsoluto
7 FinAlgoritmo
  
```

- PSDraw Window:** Shows the graphical representation of the process flowchart.

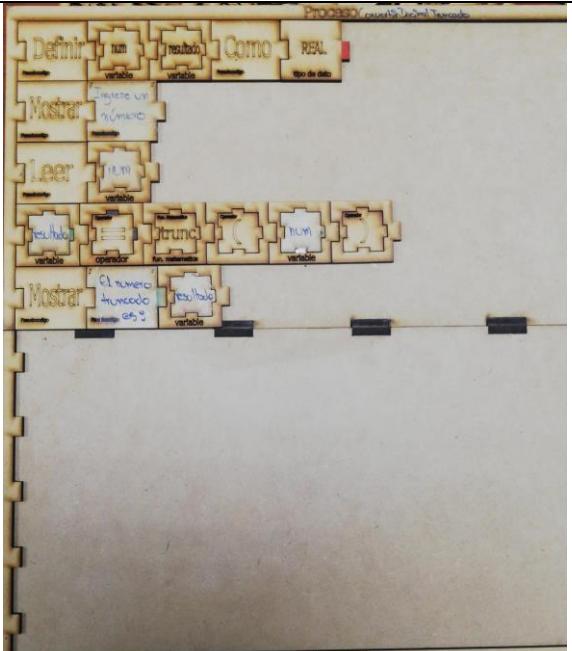
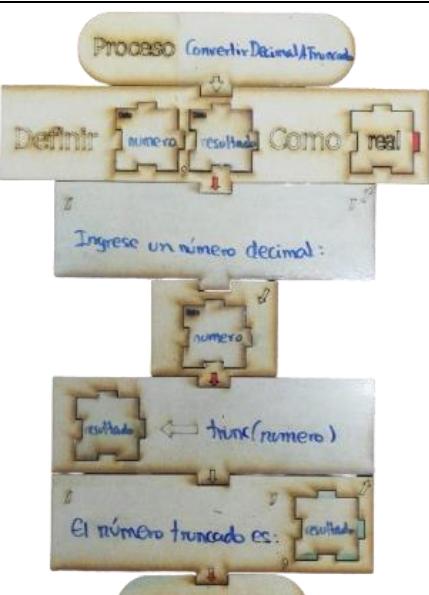
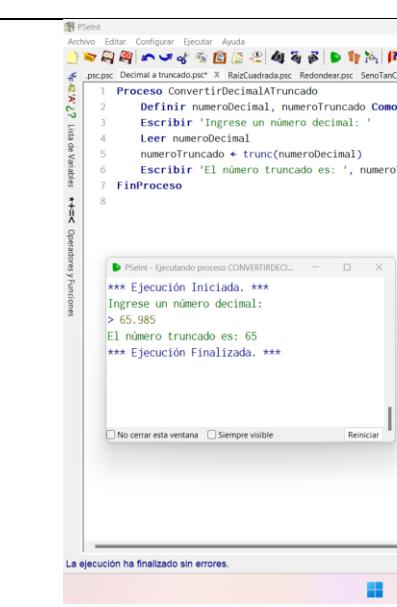
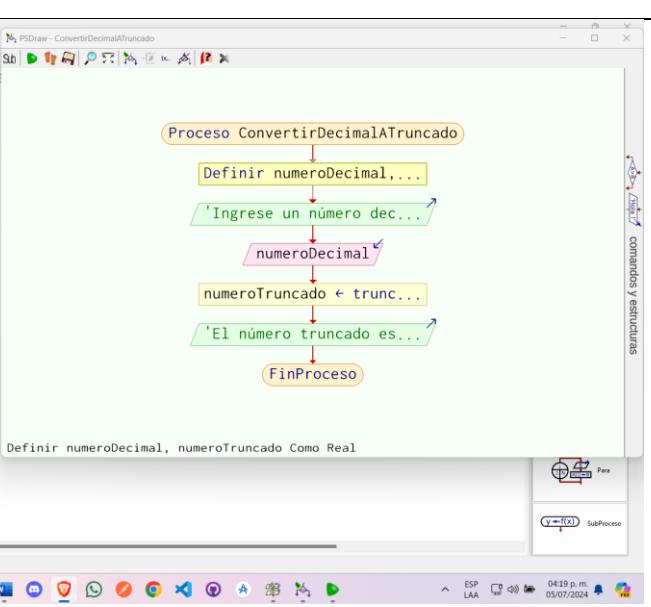
3.1.3 Convertir un número decimal a entero truncado

Diseña un programa que solicite un número decimal y lo convierta a un número entero truncado utilizando la función trunc.

Pasos a Seguir

1. Dar un nombre a este proceso: 'ConvertirDecimalTruncado'

2. Defina dos tipos de variables de tipo 'Real'
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese un número negativo: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso de un valor al usuario.
5. Asigne a la variable definida el resultado de la función matemática '**trun**'.
6. Utilice 'Mostrar' para mostrar un mensaje con el número ingresado y el resultado de la función matemática.
7. Finalice el Proceso

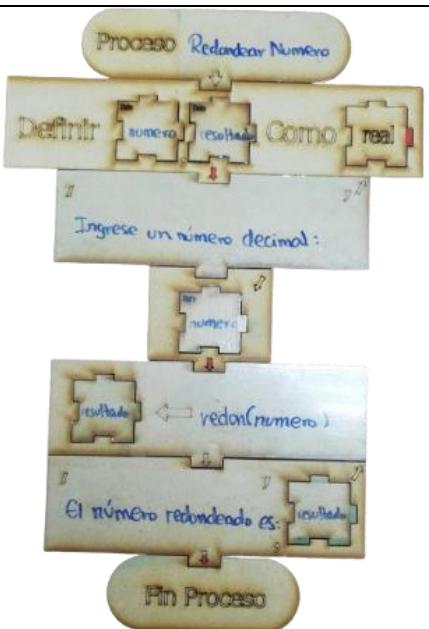
Pseudocódigo	Diagrama de Flujo
 <pre> 1 Proceso ConvertirDecimalATruncado 2 Definir numeroDecimal, numeroTruncado Como Real 3 Escribir 'Ingrese un número decimal: ' 4 Leer numeroDecimal 5 numeroTruncado ← trunc(numeroDecimal) 6 Escribir 'El número truncado es: ', numeroTruncado 7 FinProceso </pre>	 <pre> graph TD Start([Proceso ConvertirDecimalATruncado]) --> Def[Definir numero, resultado Como real] Def --> Ingresar[Ingrese un número decimal: numero] Ingresar --> Leer[Leer numero] Leer --> Truncate[resultado ← trunc(numero)] Truncate --> Resultado[El número truncado es: resultado] Resultado --> Fin([Fin Proceso]) </pre>
Comprueba la solución en PSelint	
 <pre> 1 Proceso ConvertirDecimalATruncado 2 Definir numeroDecimal, numeroTruncado Como Real 3 Escribir 'Ingrese un número decimal: ' 4 Leer numeroDecimal 5 numeroTruncado ← trunc(numeroDecimal) 6 Escribir 'El número truncado es: ', numeroTruncado 7 FinProceso </pre> <p>PSelint - Ejecutando proceso CONVERTIRDEC...</p> <p>*** Ejecución Iniciada. *** Ingrese un número decimal: > 65.985 El número truncado es: 65 *** Ejecución Finalizada. ***</p>	 <p>PSelint - Ejecutando proceso CONVERTIRDEC...</p> <p>Definir numeroDecimal, numeroTruncado Como Real</p> <pre> graph TD Start([Proceso ConvertirDecimalATruncado]) --> Def[Definir numeroDecimal,...] Def --> Ingresar[Ingrese un número dec...] Ingresar --> Leer[numeroDecimal] Leer --> Truncate[numeroTruncado ← trunc...] Truncate --> Resultado[El número truncado es...] Resultado --> Fin([Fin Proceso]) </pre> <p>La ejecución ha finalizado sin errores.</p>

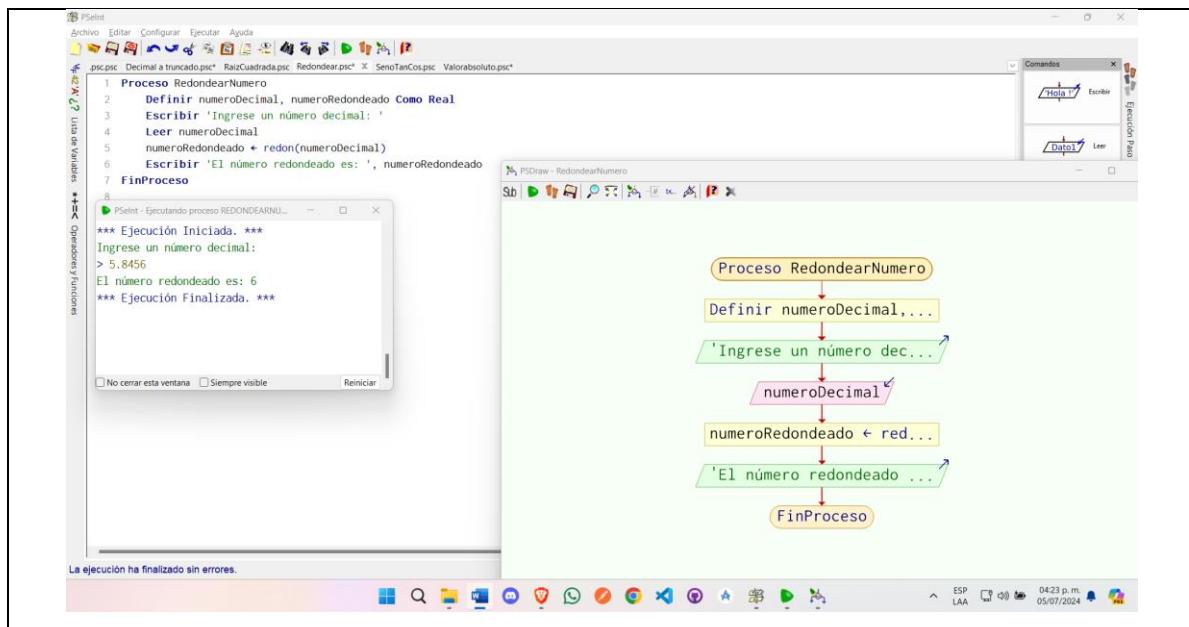
3.1.4 Redondear un número

Programa que lea un número decimal y lo redondee al entero más cercano usando la función redondear.

Pasos a Seguir

1. Dar un nombre a este proceso: 'RedondearNumero'
2. Defina dos tipos de variables de tipo 'Real'
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese un número decimal: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso de un valor al usuario.
5. Asigne a la variable definida el resultado de la función matemática '**redon**'.
6. Utilice 'Mostrar' para mostrar un mensaje con el número ingresado y el resultado de la función matemática.
7. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> PROCESO RedondearNumero Definir numero, resultado Como REAL Mostrar "Ingrese un numero decimal: " Leer numero resultado = redon(numero) Mostrar "El numero redondeado es: ", resultado Fin Proceso </pre>	
Comprueba la solución en PSeInt	



3.1.5 Cálculo del seno de un ángulo en grados

Escribe un programa que lea un ángulo en grados y calcule su seno utilizando la función seno de PSeInt.

Pasos a Seguir

1. Dar un nombre a este proceso: 'SenoAngulo'.
2. Defina dos tipos de variables de tipo 'Real'
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el grado del angulo: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso de un valor al usuario.
5. Asigne a la variable 'senoAngulo' el resultado de la función `sen(variable)`.
6. Utilice 'Mostrar' para mostrar un mensaje con el resultado de la función `seno`.
7. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
--------------	-------------------

```

    PROCESO CalcularSeno
    Definir num, resultado Como REAL tipo dato
    Mostrar "Ingrese un ángulo en grados"
    Leer num
    resultado := sen(num)
    Escribir "El seno del número es: ", resultado
    FinProceso
  
```

Proceso CalcularSenos

Definir num, resultado Como real

Ingrese un ángulo en grados:

num

resultado ← sen(num)

El seno del número es: resultado

Fin Proceso

Comprueba la solución en PSelint

PSelint

Variables Operadores y Funciones

```

1 Proceso CalcularSenos
2 Definir num, resultado Como Real
3 Escribir 'Ingrese un ángulo en grados: '
4 Leer num
5 resultado := sen(num)
6 Escribir 'El seno del número es: ', resultado
7 FinProceso
  
```

La ejecución ha finalizado sin errores.

PSelint - Ejecutando proceso CALCULARSENOS

*** Ejecución Iniciada. ***

Ingrese un ángulo en grados:

> 45

El seno de 45 grados es: 0.8509035245

*** Ejecución Finalizada. ***

Proceso CalcularSenos

Definir num, resultado...

'Ingrese un ángulo en ...'

num

resultado ← sen(num)

'El seno de ', num, ' ...'

FinProceso

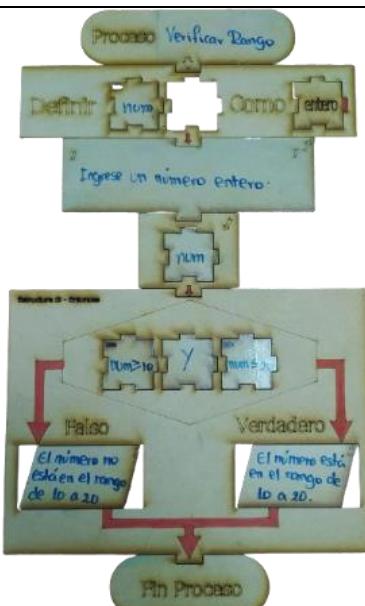
3.2 Operadores Lógicos

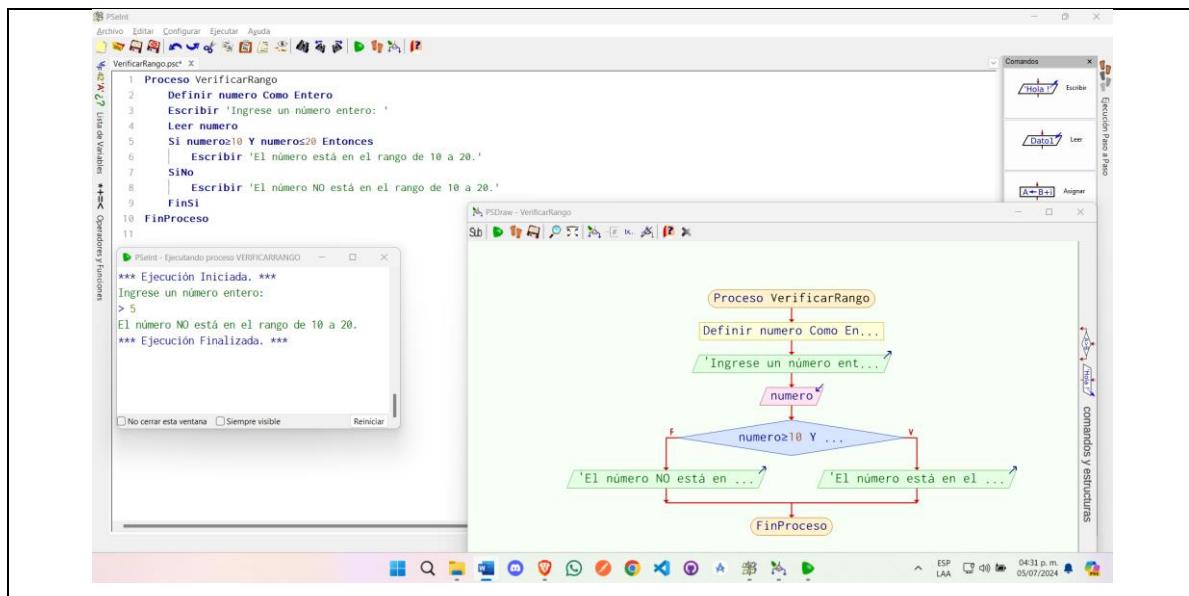
3.2.1 Verificar si un número está en un rango

Crea un programa que lea un número entero y verifique si está entre 10 y 20 (inclusive) usando operadores lógicos.

Pasos a Seguir

1. Dar un nombre a este proceso: 'VerificarRango'.
2. Defina una variable de tipo 'Entero'
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese un número entero: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso de un valor al usuario.
5. Use la estructura de control 'SI'
 - a. Si 'variable'>= 10 Y 'variable'<= 20 Entonces.
 - b. Mostrar 'El numero ingresado esta en el rango 10 y 20.'
 - c. Sino Mostrar 'El numero NO esta en el rango de 10 y 20'.
 - d. Fin Si
6. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Proceso VerificarRango Definir num Como Entero Mostrar "Ingrese un numero entero: " Leer num Si num >= 10 Y num <= 20 Entonces Mostrar "El numero ingresado esta en el rango 10 a 20." Sino Mostrar "El numero NO esta en el rango de 10 a 20." Fin Si End Proceso </pre>	
Comprueba la solución en PSeInt	



3.2.2 Validar si al menos una condición es verdadera

Escribe un programa que lea tres números y verifique si al menos uno de ellos es mayor que 50.

Pasos a Seguir

1. Dar un nombre a este proceso: 'VerificarCondicion'
2. Defina dos variables de tipo 'Entero'
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el primer número: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso de un valor al usuario.
5. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el segundo número: '
6. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso de un valor al usuario.
7. Use la estructura de control 'SI'
 - a. Si 'variable1' > 50 Y 'variable2' > 50 Entonces.
 - b. Mostrar 'Al menos uno de los números ingresados es mayor a 50'.
 - c. Sino Mostrar 'Ninguno de los números es mayor a 50'.
 - d. Fin Si
8. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
--------------	-------------------

```

    graph TD
        Start([Proceso VerificarAlMenosUnaCondicion]) --> Definir1[Definir numero1, numero2 Como Entero]
        Definir1 --> Leer1[Ler numero1]
        Leer1 --> Ingresar1[Ingrese el primer número: ]
        Ingresar1 --> Leer2[Ler numero2]
        Leer2 --> Ingresar2[Ingrese el segundo número: ]
        Ingresar2 --> Cond1{numero1 >= 50}
        Cond1 --> S1[Si]
        S1 --> Mostrar1[Mostrar Ninguno de los números es mayor que 50]
        S1 --> Cond2{numero2 >= 50}
        Cond2 --> S2[Sino]
        S2 --> Mostrar2[Mostrar Al menos uno de los números es mayor que 50]
        S2 --> FinS[Fin Si]
        FinS --> FinP[Fin Proceso]
    
```

Comprueba la solución en PSeInt

```

1 Proceso VerificarAlMenosUnaCondicion
2     Definir numero1, numero2 Como Entero
3     Escribir 'Ingrese el primer número: '
4     Leer numero1
5     Escribir 'Ingrese el segundo número: '
6     Leer numero2
7     Si numero1>=50 O numero2>=50 Entonces
8         Escribir 'Al menos uno de los números es mayor que 50.'
9     SiNo
10        Escribir 'Ninguno de los números es mayor que 50.'
11    FinSi
12 FinProceso
    
```

*** Ejecución Iniciada. ***
 Ingrese el primer número:
 > 50
 Ingrese el segundo número:
 > 3
 Ninguno de los números es mayor que 50.
 *** Ejecución Finalizada. ***

```

graph TD
    Start([Proceso VerificarAlMenosUnaCondicion]) --> Definir1[Definir numero1, numero...]
    Definir1 --> Leer1['Ingrese el primer númer...']
    Leer1 --> numero1[numero1]
    numero1 --> Leer2['Ingrese el segundo númer...']
    Leer2 --> numero2[numero2]
    numero2 --> Cond{numero1>=50 O ...}
    Cond --> F{Falso}
    F --> Ninguno[Ninguno de los númer...]
    Ninguno --> FinP[FinProceso]
    Cond --> V{Verdadero}
    V --> Almenos[Al menos uno de los númer...]
    Almenos --> FinP
    
```

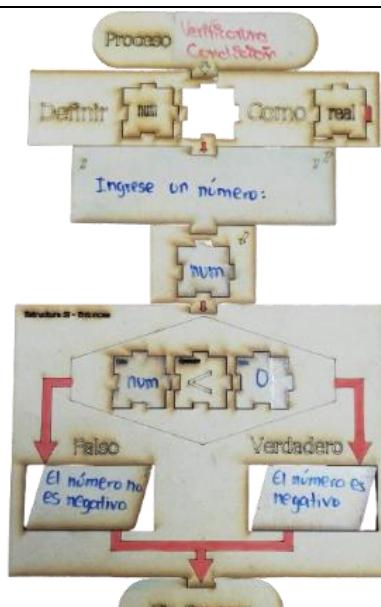
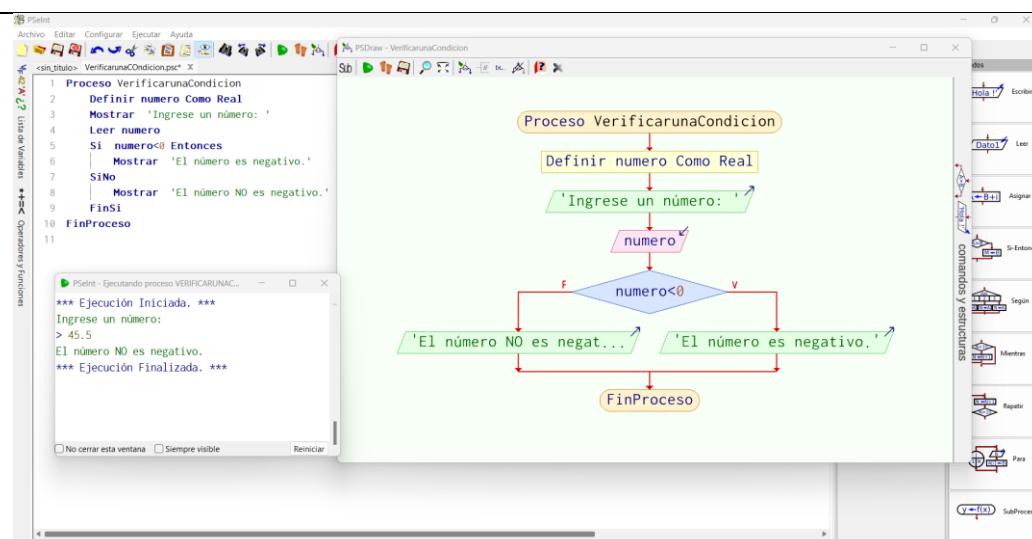
3.2.3 Verificar una condición

Crea un programa que lea un número y verifique si NO es negativo.

Pasos a Seguir

1. Dar un nombre a este proceso: 'VerificarunaCondicion'.
2. Defina una variable de tipo 'Rela
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese un número: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso de un valor al usuario.

5. Use la estructura de control 'SI'
 - a. Si 'variable' < 0 Entonces.
 - b. Mostrar 'El numero ingresado es negativo'.
 - c. Sino Mostrar 'El numero ingresado NO es negativo.'
 - d. Fin Si
6. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Proceso VerificarunaCondicion 1 Definir numero Como Real 2 Mostrar 'Ingrese un número:' 3 Leer numero 4 Si numero<0 Entonces Mostrar 'El número es negativo.' 8 Sino Mostrar 'El número NO es negativo.' 9 FinSi 10 FinProceso </pre>	
Comprueba la solución en PSeInt	
	

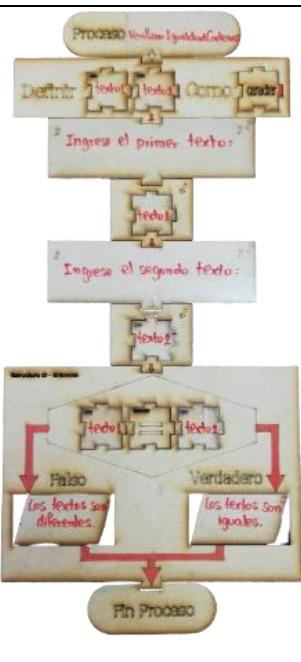
3.3 Operadores Relacionales

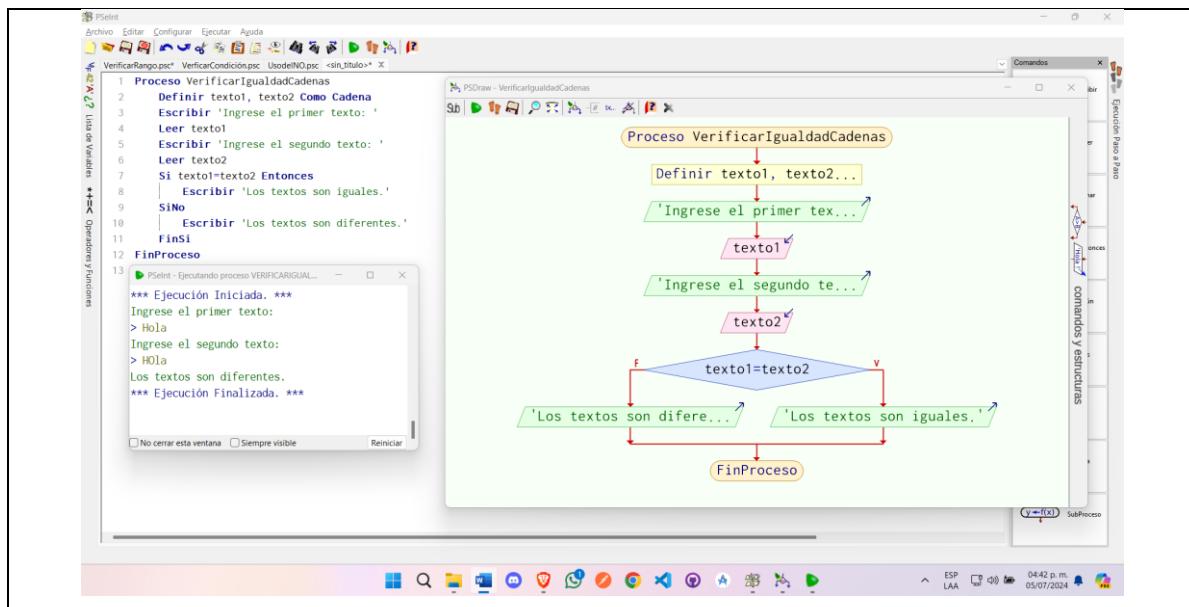
3.3.1 Verificar igualdad de datos 'CARACTER'

Diseña un programa que lea dos cadenas de texto y determine si son iguales.

Pasos a Seguir

1. Dar un nombre a este proceso: 'TextoSimilar'.
2. Defina dos variables de tipo 'Caracter'
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el primer texto: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
5. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el segundo texto: '
6. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
7. Use la estructura de control 'SI'
 - a. Si 'variable1' = 'variable2' Entonces.
 - b. Mostrar 'Los textos son iguales'
 - c. Sino Mostrar 'Los textos no son iguales'.
 - d. Fin Si
8. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> PROCESO "Verificar Igualdad de Cadena" Definir Texto1 Texto2 Como CARACTER Mostrar "Ingrese el primer texto:" Texto1 Leer Texto1 Mostrar "Ingrese el segundo texto:" Texto2 Leer Texto2 Si Texto1 = Texto2 Entonces Mostrar "Los textos son iguales." Sino Mostrar "Los textos no son iguales." Fin Si Fin Proceso </pre>	
Comprueba la solución en PSeInt	



3.3.2 Comprobar si un número es mayor o igual a otro

Crea un programa que lea dos números y determine si el primero es mayor o igual al segundo.

Pasos a Seguir

1. Dar un nombre a este proceso: 'ComprobarMayorigual'.
2. Defina dos variables de tipo 'Entero'
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el primer numero: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
5. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el segundo numero: '
6. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
7. Use la estructura de control 'SI'
 - a. Si 'variable1' >= 'variable2' Entonces.
 - b. Mostrar 'El primer numero es mayor o igual al segundo'.
 - c. Sino Mostrar 'El primer numero es menor que el segundo'.
 - d. Fin Si
8. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
--------------	-------------------

```

PROCESO [Comprobar Mayor Igual]
Definir num1, num2 Como Entero
Mostrar "Ingresé el primer número"
Ler num1
Mostrar "Ingresé el segundo número"
Ler num2
Si num1 > num2 Entonces
    Mostrar "El primer número es mayor o igual al segundo."
Sino
    Mostrar "El primer número es menor que el segundo."
FinSí
Fin Proceso

```

```

PROCESO [Comprobar Mayor Igual]
Definir num1, num2 Como Entero
Mostrar "Ingresé el primer número"
Ler num1
Mostrar "Ingresé el segundo número"
Ler num2
num1 > num2?
    Si No Entonces
        Mostrar "El primer número es menor que el segundo."
    Sino
        Mostrar "El primer número es mayor o igual al segundo."
    FinSí
Fin Proceso

```

Comprueba la solución en PSelint

```

1 Proceso ComprobarMayorIgual
2 Definir numero1, numero2 Como Entero
3 Escribir "Ingresé el primer número: "
4 Leer numero1
5 Escribir "Ingresé el segundo número: "
6 Leer numero2
7 Si numero1>numero2 Entonces
8     Escribir "El primer número es mayor o igual al segundo."
9 SiNo
10    Escribir "El primer número es menor que el segundo."
11 FinSí
12 FinProceso
13

```

PSelint - Ejecutando proceso COMPROBARMA...
*** Ejecución Iniciada. ***
Ingresé el primer número:
> 15
Ingresé el segundo número:
> 5
El primer número es mayor o igual al segundo.
*** Ejecución Finalizada. ***

PSelint - Ejecutando proceso COMPROBARMA...

Asignación/Dimensión/Definición

```

Proceso ComprobarMayorIgual
Definir numero1, numero...
'Ingresé el primer númer...
numero1
'Ingresé el segundo númer...
numero2
numero1>numero2?
    Si No Entonces
        'El primer número es m...
    Sino
        'El primer número es m...
    FinSí
FinProceso

```

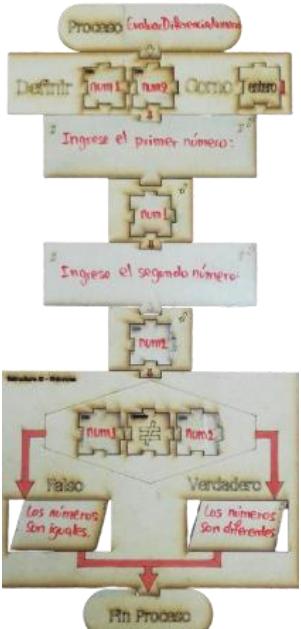
3.3.3 Evaluuar la diferencia entre dos números

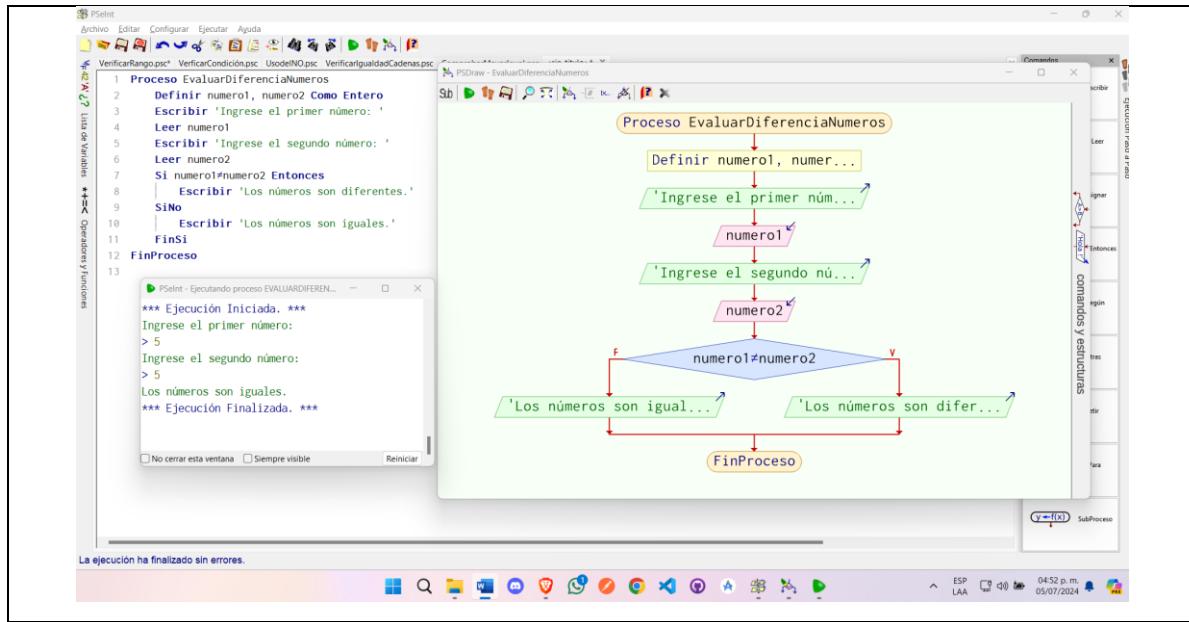
Escribe un programa que lea dos números y verifique si son diferentes.

Pasos a Seguir

1. Dar un nombre a este proceso: 'EvaluuarDiferencia'.
2. Defina dos variables de tipo 'Entero'
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingresé el primer numero: '

4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
5. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el segundo numero: '
6. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
7. Use la estructura de control 'SI'
 - a. Si 'variable1' <> 'variable2' Entonces.
 - b. Mostrar 'Los números son diferentes'.
 - c. Sino Mostrar 'Los numero son iguales'.
 - d. Fin Si
8. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Definir num1, num2 Como ENTERO Mostrar "Ingrese el primer número" Leer num1 Mostrar "Ingrese el segundo número" Leer num2 Si num1 # num2 Entonces Mostrar "Los números son diferentes" Sino Mostrar "Los numero son iguales" Fin Si </pre>	
Comprueba la solución en PSeInt	



3.3.4 Determinar si un número es menor que otro

Programa que lea dos números y determine si el primero es menor que el segundo.

Pasos a Seguir

1. Dar un nombre a este proceso: 'DeterminarMenor'.
2. Defina dos variables de tipo 'Entero'
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el primer numero: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
5. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el segundo numero: '
6. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
7. Use la estructura de control 'SI'
 - a. Si 'variable1' < 'variable2' Entonces.
 - b. Mostrar 'El primer numero es menor al segundo'.
 - c. Sino Mostrar 'El primer numero no es menor que el segundo'.
 - d. Fin Si
8. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Proceso DeterminarMenor 1. Definir num1, num2 Como Entero 2. Leer num1 3. Leer num2 4. Si num1 < num2 Entonces 5. Mostrar "El primer número es menor que el segundo." 6. Sino 7. Mostrar "El primer número no es menor que el segundo." 8. FinSi Fin Proceso </pre>	
Comprueba la solución en PSeInt	

3.4 Operadores Algebraicos

3.4.1 Suma de dos números

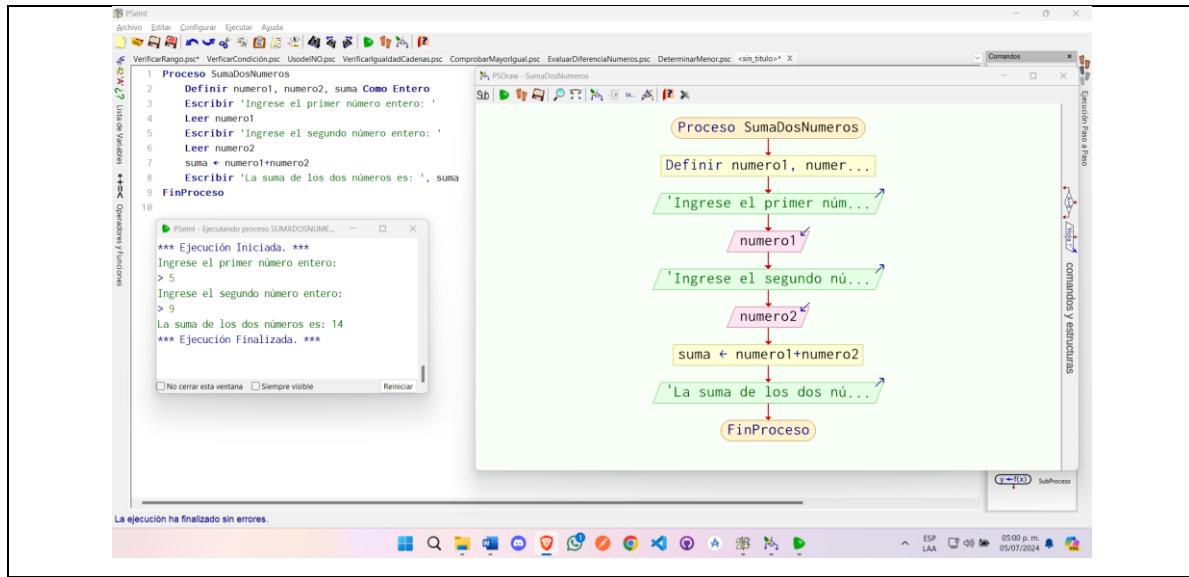
Escribe un programa que lea dos números enteros y muestre su suma.

Pasos a Seguir

1. Dar un nombre a este proceso: 'Sumadosnumeros'.
2. Defina tres variables de tipo 'Entero'.

3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el primer numero: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
5. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el segundo numero: '
6. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
7. Asigne en la variable 'resultado' la operación 'variable1' + 'variable2'.
8. Utilice 'Mostrar' con el siguiente mensaje 'El resultado de la suma es: ', resultado.
9. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> PROCESO SumaDosNumeros Definir Num1 tipo de dato Definir Num2 tipo de dato Definir Suma tipo de dato Mostrar "Ingrese el primer numero" Leer Num1 Mostrar "Ingrese el segundo numero" Leer Num2 Suma := Num1 + Num2 Mostrar "La suma de los dos numeros es: " Mostrar Suma Fin Proceso </pre>	
Comprueba la solución en PSeInt	



3.4.2 Resta de dos números

Crea un programa que lea dos números decimales y muestre la diferencia entre ellos.

Pasos a Seguir

1. Dar un nombre a este proceso: 'Restadosnumeros'.
2. Defina tres variables de tipo 'Entero'
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el primer numero: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
5. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el segundo numero: '
6. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
7. Asigne en la variable 'resultado' la operación 'variable1' - 'variable2'.
8. Utilice 'Mostrar' con el siguiente mensaje 'El resultado de la resta es: ', resultado.
9. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
--------------	-------------------

```

    Definir num1, num2, resta Como Real
    Ingrese el primer número
    Leer num1
    Ingrese el segundo número
    Leer num2
    resta ← num1 - num2
    Mostrar La resta de los dos números es: resta
  
```

```

    Proceso RestaDosNumeros
    ↓
    Definir numero1, numero2 Como Real
    ↓
    Ingrese el primer número:
    numero1 ←
    ↓
    Ingrese el segundo número:
    numero2 ←
    ↓
    diferencia ← numero1 - numero2
    ↓
    La diferencia entre los dos números es: diferencia
    ↓
    FinProceso
  
```

Comprueba la solución en PSeInt

```

1 Proceso RestaDosNumeros
2   Definir numero1, numero2, diferencia Como Real
3   Escribir 'Ingrese el primer número decimal: '
4   Leer numero1
5   Escribir 'Ingrese el segundo número decimal: '
6   Leer numero2
7   diferencia ← numero1 - numero2
8   Escribir 'La diferencia entre los dos números es: ', diferencia
9 FinProceso
  
```

PSelint - Ejecutando proceso RESTADOSNUMEROS...
*** Ejecución Iniciada. ***
Ingresé el primer número decimal:
> 9
Ingresé el segundo número decimal:
> 745
La diferencia entre los dos números es: -736
*** Ejecución Finalizada. ***

PSelint - RestaDosNumeros

```

    Proceso RestaDosNumeros
    ↓
    Definir numero1, numero...
    ↓
    'Ingrese el primer núm...'
    numero1 ←
    ↓
    'Ingrese el segundo núm...'
    numero2 ←
    ↓
    diferencia ← numero1 - n...
    ↓
    'La diferencia entre 1...
    ↓
    FinProceso
  
```

Asignación/Dimensión/Definición

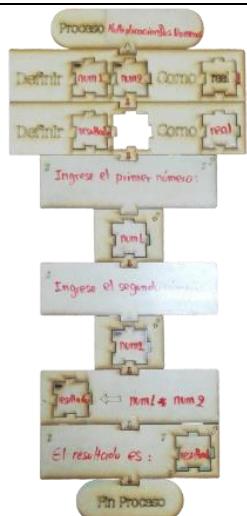
3.4.3 Multiplicación de dos números

Diseña un programa que lea tres números y muestre su producto.

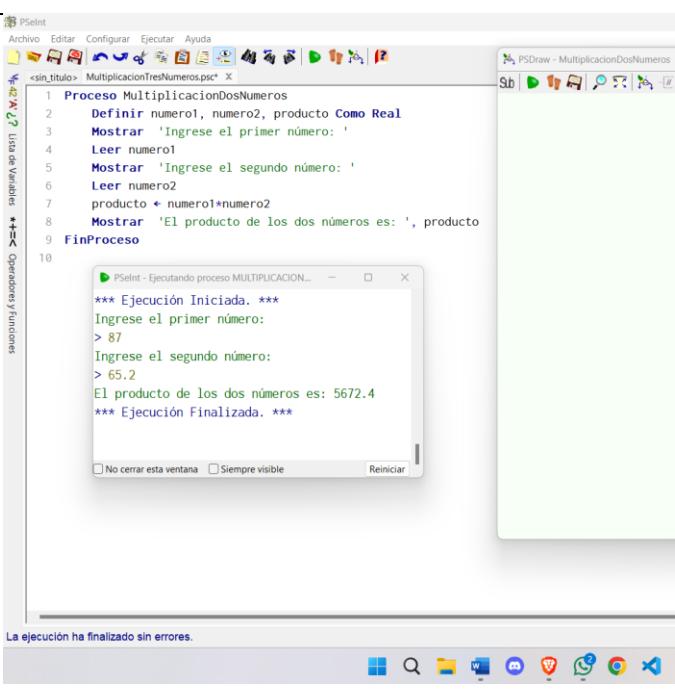
Pasos a Seguir

1. Dar un nombre a este proceso: 'Productodosnumeros'.
2. Defina tres variables de tipo 'Real'.
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el primer numero: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.

5. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el segundo numero: '
6. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
7. Asigne en la variable 'resultado' la operación 'variable1 * variable2'.
8. Utilice 'Mostrar' con el siguiente mensaje 'El resultado de la multiplicacion es: ', resultado.
9. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Proceso MultiplicacionDosNumeros 1 Definir numero1, numero2, producto Como REAL 2 Mostrar "Ingrese el primer número: " 3 Leer numero1 4 Mostrar "Ingrese el segundo número: " 5 Leer numero2 6 producto ← numero1*numero2 7 Mostrar "El producto de los dos números es: ", producto 8 FinProceso </pre>	

Comprueba la solución en PSelnt

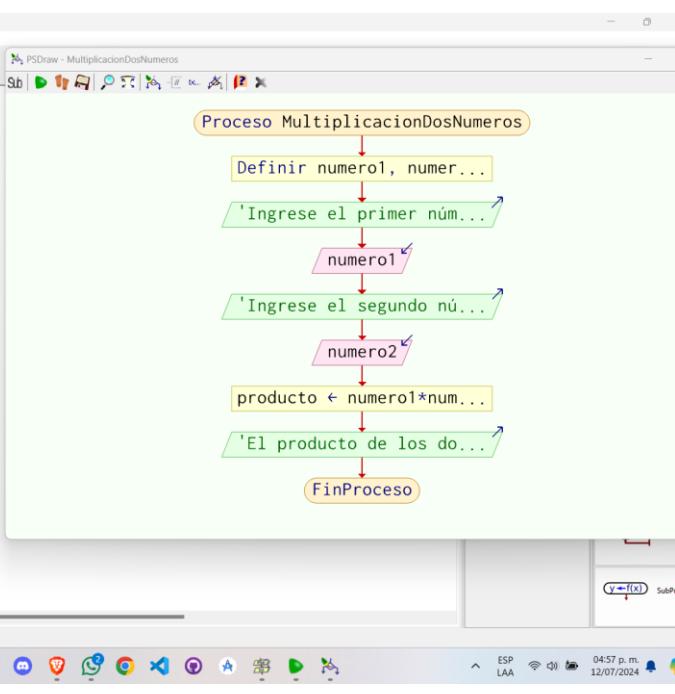


The terminal window shows:

```

*** Ejecución Iniciada. ***
Ingrese el primer número:
> 87
Ingrese el segundo número:
> 65.2
El producto de los dos números es: 5672.4
*** Ejecución Finalizada. ***

```



The flowchart steps are annotated as follows:

- Step 1: 'Definir numero1, numero...'
- Step 2: 'Ingrese el primer núm...'
- Step 3: 'numero1'
- Step 4: 'Ingrese el segundo núm...'
- Step 5: 'numero2'
- Step 6: 'producto ← numero1*num...'
- Step 7: 'El producto de los do...'
- Step 8: 'FinProceso'

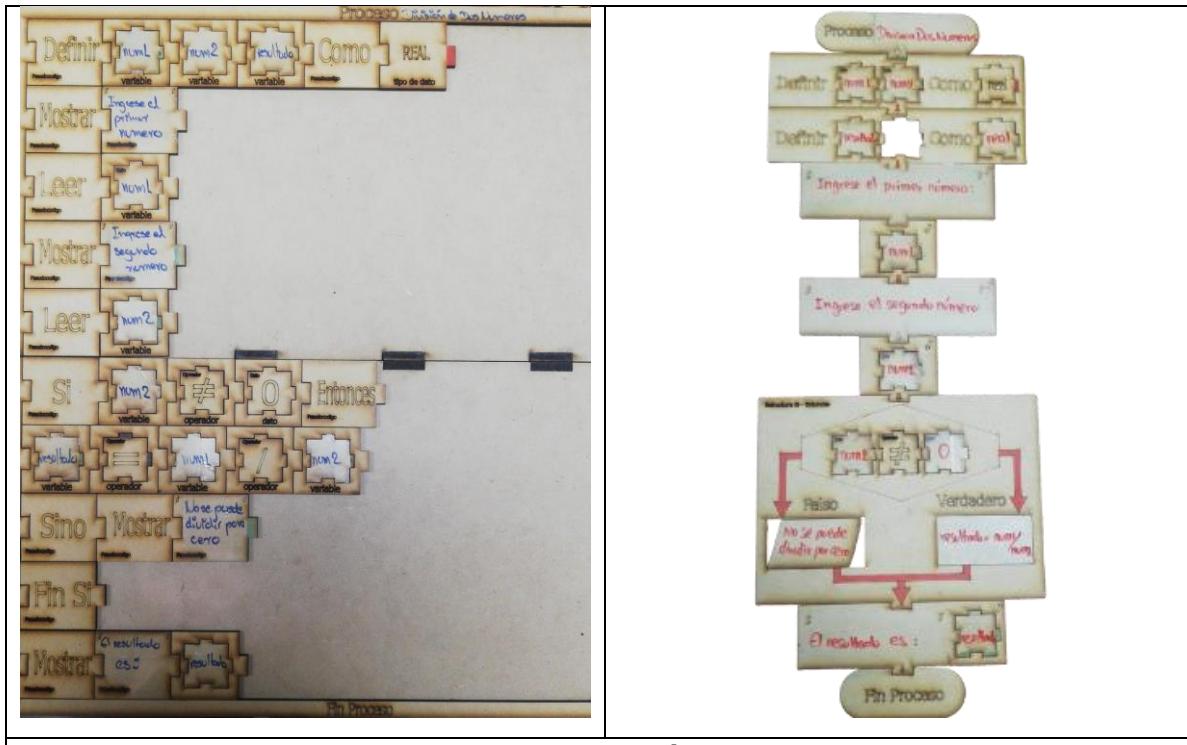
3.4.4 División de dos números

Programa que lea dos números y muestre el cociente y el residuo de la división del primero entre el segundo.

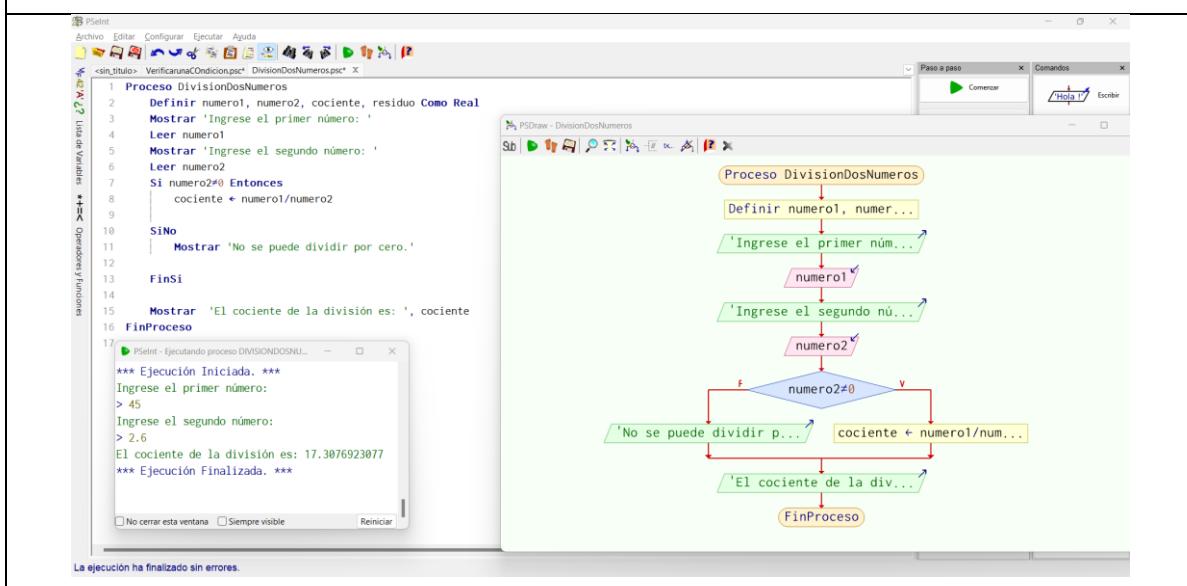
Pasos a Seguir

1. Dar un nombre a este proceso: 'Dividirdosnumeros'.
2. Defina tres variables de tipo 'Real'.
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el primer numero: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
5. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el segundo numero: '
6. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
7. Use la estructura de control 'SI'
 - a. Si 'variable2' <> 0 Entonces.
 - b. Asigne en la variable 'resultado' la operación 'variable1' * 'variable2'.
 - c. Sino Mostrar 'No se puede dividir para cero'.
 - d. Fin Si
9. Utilice 'Mostrar' con el siguiente mensaje 'El resultado de la division es: ', resultado.
10. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
--------------	-------------------



Comprueba la solución en PSeInt



3.4.5 Cálculo del área de un triángulo

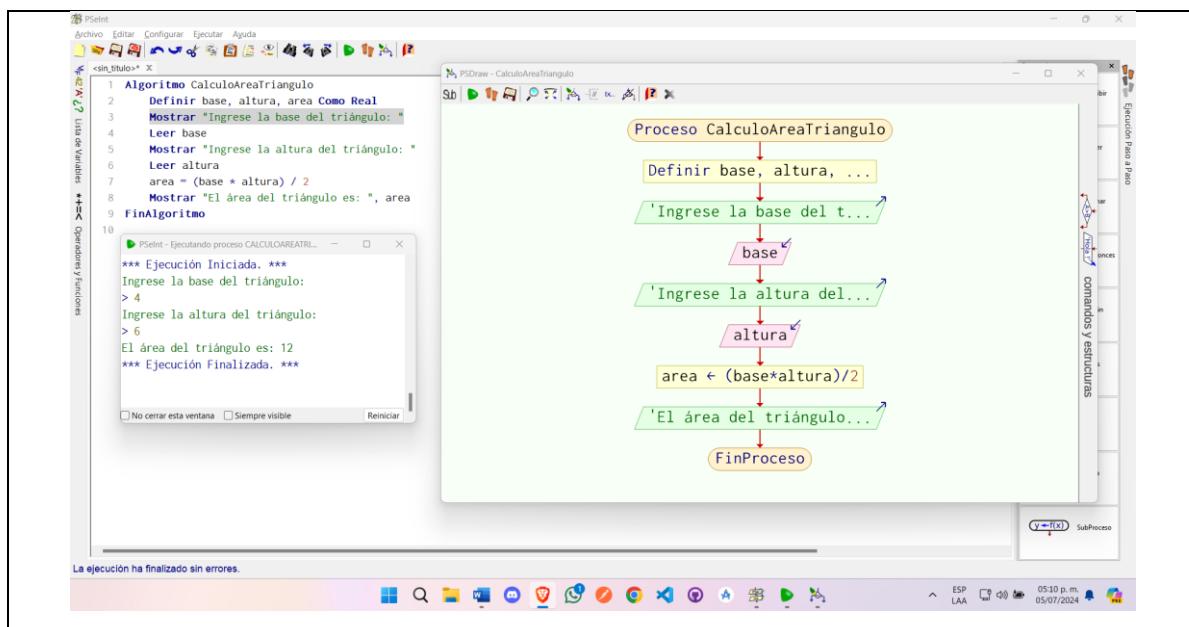
Escribe un programa que lea la base y la altura de un triángulo y calcule su área utilizando la fórmula $\text{Área} = (\text{base} * \text{altura}) / 2$.

Pasos a Seguir

1. Dar un nombre a este proceso: 'CalcularAreaTriangulo'.
2. Defina tres variables de tipo 'Entero'.

3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el valor de la base del triangulo: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
5. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el valor de la altura del triangulo: '
6. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del texto al usuario.
7. Asigne en la variable 'area' la operación (base * altura) /2.
8. Utilice 'Mostrar' con el siguiente mensaje 'El área del triangulo es: ', area.
9. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Definir base como real Mostrar "Ingrese la base del triangulo" Leer base Mostrar "Ingrese la altura del triangulo" Leer altura Mostrar "El area del triangulo es: " Mostrar area Fin Proceso </pre>	
Comprueba la solución en PSeInt	



3.5 Ejercicios con 'SI'

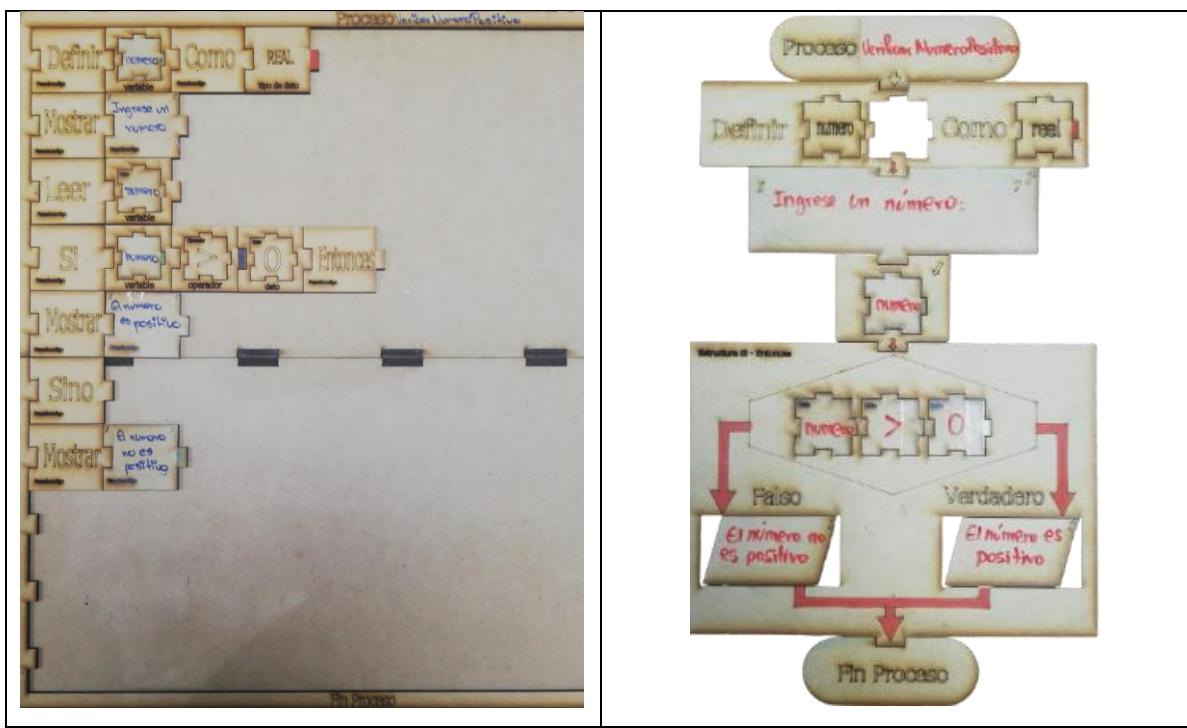
3.5.1 Verificar si un número es positivo

Escribe un programa que lea un número y determine si es positivo. Si lo es, debe mostrar el mensaje "El número es positivo".

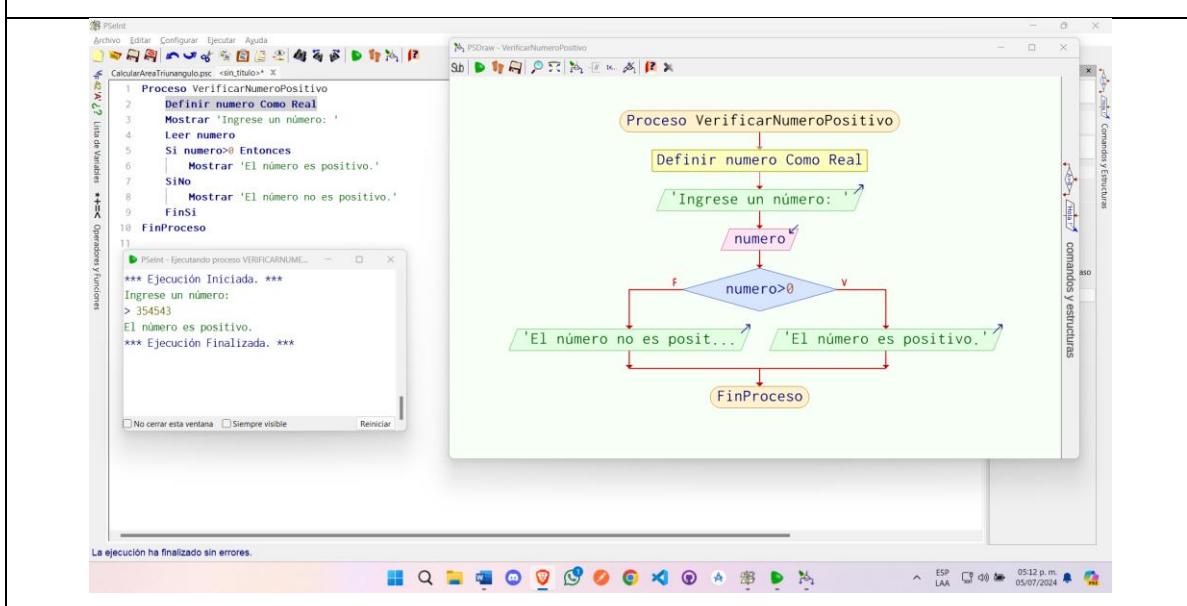
Pasos a Seguir

1. Dar un nombre a este proceso: 'VerificarNumPositivo'.
2. Defina una variable de tipo 'Real'.
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese un numero: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del valor al usuario.
5. Use la estructura de control 'SI'
 - a. Si 'variable1' > 0 Entonces.
 - b. Mostrar 'El numero ingresado es positivo'
 - c. Sino Mostrar 'El numero no es positivo'.
 - d. Fin Si
6. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
--------------	-------------------



Comprueba la solución en PSeInt



3.5.2 Determinar si un número es par o impar

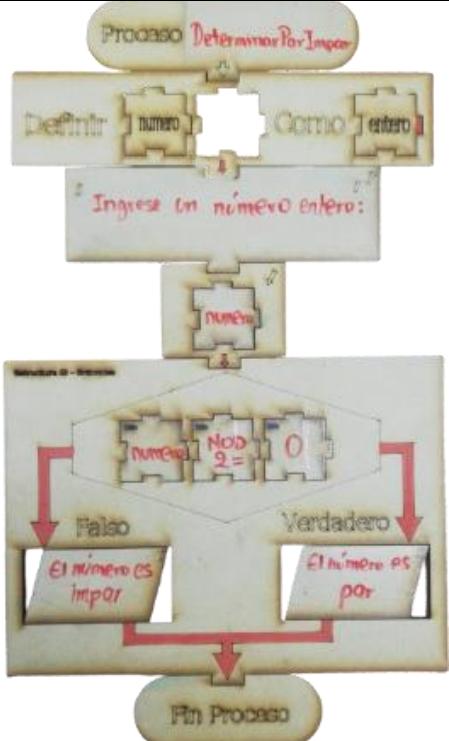
Crea un programa que lea un número entero y determine si es par o impar.

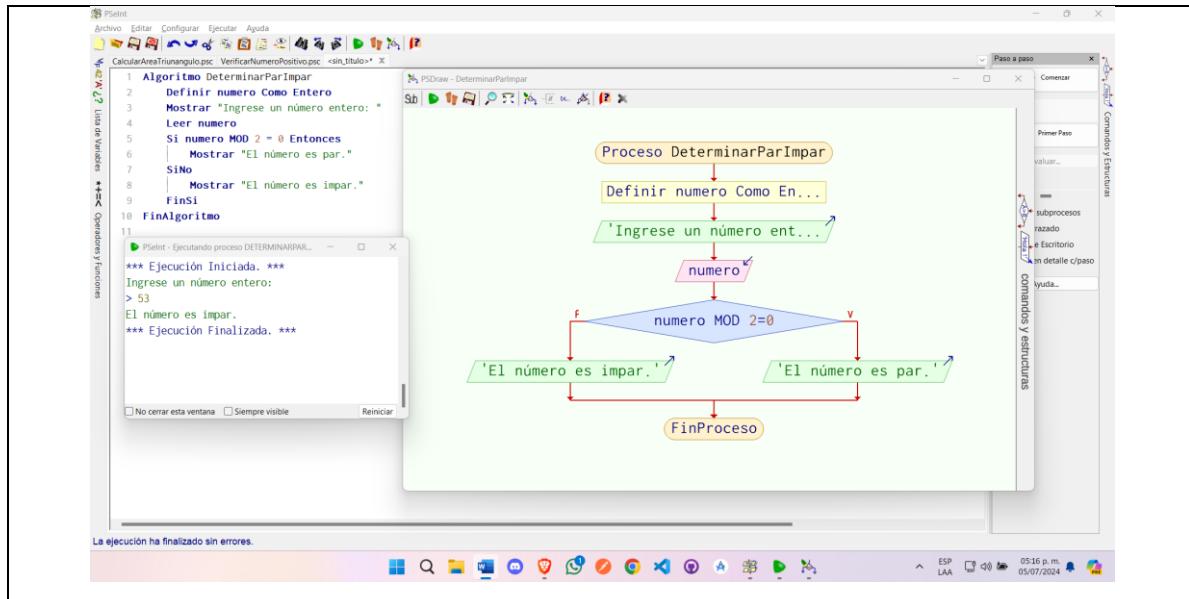
Muestra un mensaje acorde a cada caso.

Pasos a Seguir

1. Dar un nombre a este proceso: 'VerificarNumImpar'.
2. Defina una variable de tipo 'Real'.
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese un numero: '

4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del valor al usuario.
5. Use la estructura de control 'SI'
 - a. Si 'variable1' MOD 2 = 0 Entonces.
 - b. Mostrar 'El numero es par'
 - c. Sino Mostrar 'El numero es impar'.
 - d. Fin Si
6. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Proceso DeterminarParImpar Definir Número Como ENTERO Mostrar "Ingresar un número" Leer Número Si Número MOD 2 = 0 Entonces Mostrar "El número es par" Sino Mostrar "El número es impar" Fin Si Fin Proceso </pre>	
Comprueba la solución en PSeInt	



3.5.3 Calcular descuento por compra

Diseña un programa que lea el valor de una compra y aplique un descuento del 10% si el monto es mayor a 100. Muestra el total a pagar después del descuento.

Pasos a Seguir

1. Dar un nombre a este proceso: 'DescuentoCompra'.
2. Defina dos variables de tipo 'Real'.
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el valor total de la compra: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del valor al usuario.
5. Use la estructura de control 'SI'
 - a. Si 'variable1' > 100 Entonces.
 - b. Asigne el resultado 'variable1' * 0.90 a la variable 'totalapagar'.
 - c. Sino Asigne el resultado 'variable1' a la variable 'totalapagar'.
 - d. Fin Si
6. Utilice 'Mostrar' para mostrar texto al usuario: 'El valor total a pagar es: '
7. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
--------------	-------------------

```

    PROCESO CalcularDescuentoCompra
    1 Definir valorCompra, totalPagar Como Real
    2 Leer valorCompra
    3 Si valorCompra > 100 Entonces
    4   totalPagar = valorCompra * 0.90
    5 Sino
    6   totalPagar = valorCompra
    7 FinSi
    8 Mostrar "El total a pagar despues del descuento es: ", totalPagar
    9 FinAlgoritmo
  
```

The diagram shows the execution of the algorithm. It starts with defining variables, reading a value, and then branching based on a condition. If the value is greater than 100, it calculates a 10% discount. Otherwise, it keeps the original value. Finally, it prints the result.

Comprueba la solución en PSeInt

The screenshot shows the algorithm code in PSeInt. The execution window displays the input value (556), the calculated result (506.4), and the completion message.

```

    Algoritmo CalcularDescuentoCompra
    1 Definir valorCompra, totalPagar Como Real
    2 Leer valorCompra
    3 Si valorCompra > 100 Entonces
    4   totalPagar = valorCompra * 0.90
    5 Sino
    6   totalPagar = valorCompra
    7 FinSi
    8 Mostrar "El total a pagar despues del descuento es: ", totalPagar
    9 FinAlgoritmo
  
```

*** Ejecución Iniciada. ***
Ingrese el valor de la compra:
> 556
El total a pagar despues del descuento es: 506.4
*** Ejecución Finalizada. ***

This screenshot shows the flowchart in PSeInt. The execution window displays the input value (556), the calculated result (506.4), and the completion message.

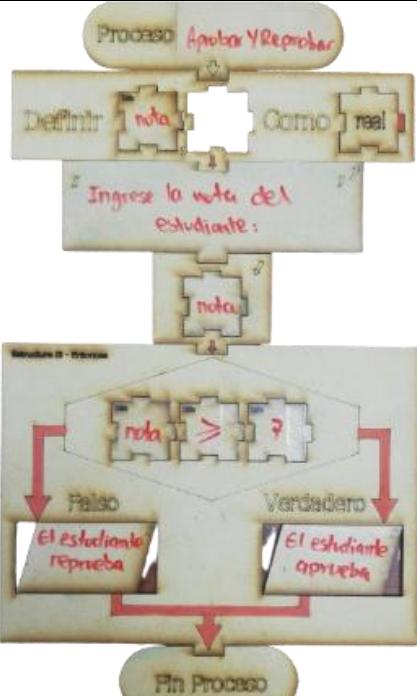
3.5.4 Evaluar si un estudiante aprueba o repreeba

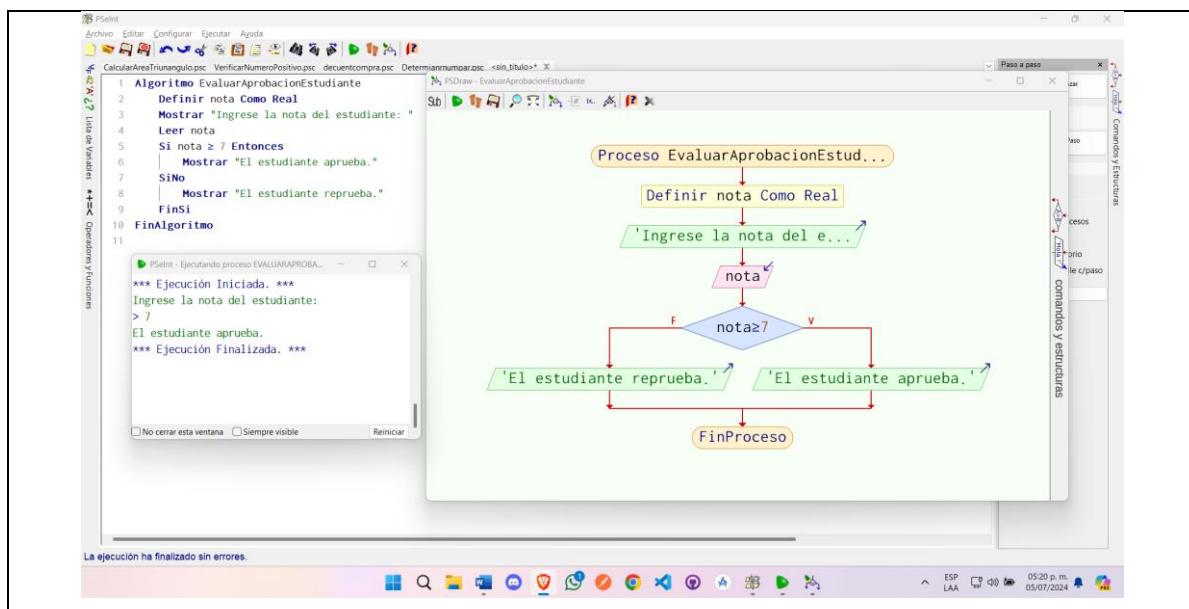
Programa que lea la nota de un estudiante y determine si aprueba (nota mayor o igual a 70) o repreeba. Muestra un mensaje con el resultado.

Pasos a Seguir

1. Dar un nombre a este proceso: 'NotasEstudiantes'.
2. Defina una variable de tipo 'Real'.
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese la nota del estudiante:

4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del valor al usuario.
5. Use la estructura de control 'SI'
 - a. Si 'variable1' >= 7 Entonces.
 - b. Mostrar 'El estudiante aprueba'
 - c. Sino Mostrar 'El estudiante no aprueba'.
 - d. Fin Si
6. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Proceso Nota a Recreto Definir nota Como REAL Mostrar "Ingrese la nota del estudiante" Leer nota Si nota >= 7 Entonces Mostrar "El estudiante aprueba" Sino Mostrar "El estudiante reprende" Fin Si Fin Proceso </pre>	
Comprueba la solución en PSeInt	



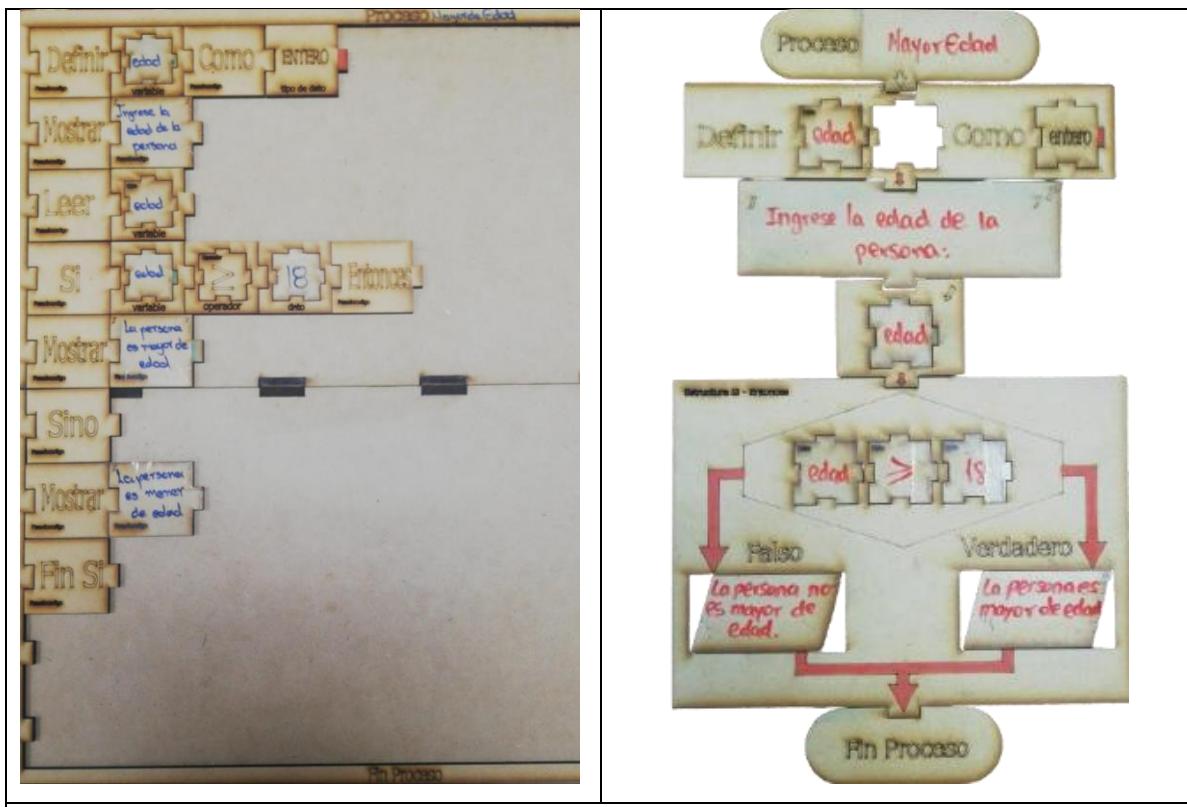
3.5.5 Comprobar si una persona es mayor de edad

Escribe un programa que lea la edad de una persona y determine si es mayor de edad (18 años o más). Muestra un mensaje correspondiente.

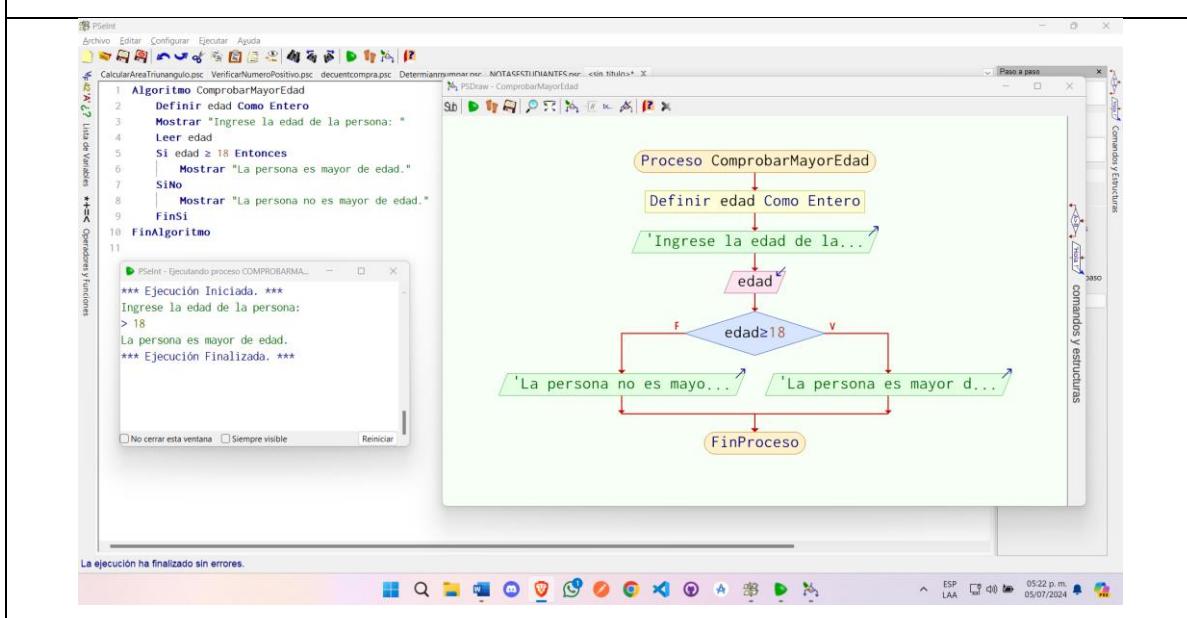
Pasos a Seguir

1. Dar un nombre a este proceso: 'EdadPersona'.
2. Defina una variable de tipo 'Entero'.
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese la edad de la persona: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del valor al usuario.
5. Use la estructura de control 'SI'
 - a. Si 'variable1' >= 18 Entonces.
 - b. Mostrar 'La persona es mayor de edad'
 - c. Sino Mostrar 'La persona es menor de edad'.
 - d. Fin Si
6. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
--------------	-------------------



Comprueba la solución en PSelint



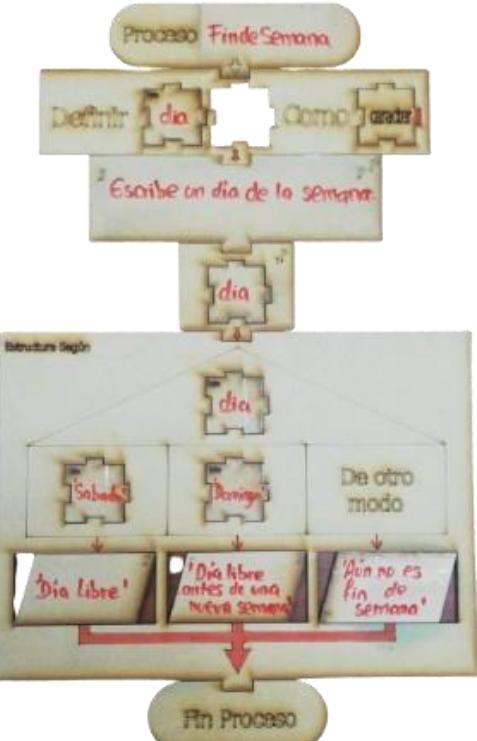
3.6 Ejercicios con 'SEGUN'

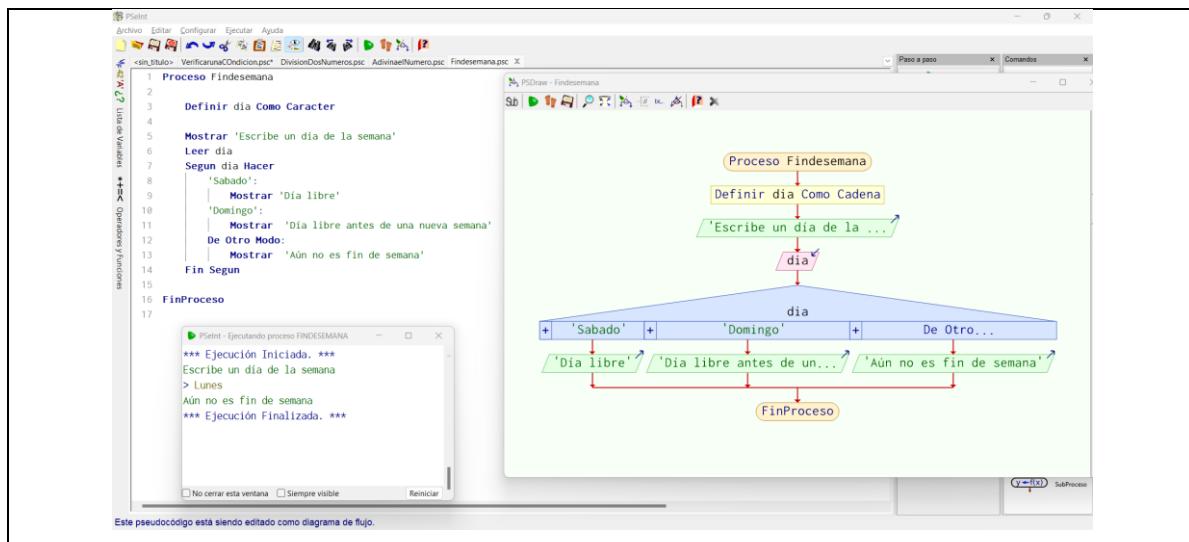
3.6.1 Mostrar día de la semana

Crea un programa que lea el texto que ingresa el usuario solicitando un día de la semana. Sabado = 'Día libre', Domingo = 'Día libre antes de una nueva semana' y De otro modo = Aún no es fin de semana'.

Pasos a Seguir

1. Dar un nombre a este proceso: 'FindeSemana'.
2. Defina una variable de tipo 'Carácter'.
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Escriba un día de la semana'
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del valor al usuario.
5. Use la estructura de control 'Según' la 'variable' Hacer
 - a. Caso 'Sabado'
 - b. Mostrar 'Dia libre'.
 - c. Caso 'Domingo'
 - d. Mostrar 'Dia libre antes de una nueva semana'.
 - e. De otro Modo
 - f. Mostrar 'Aun no es fin de semana'.
 - g. Fin Segun
6. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Proceso FindeSemana Definir dia Como CHARACTER Mostrar "Escribe un dia de la semana" Leer dia Segun dia Caso "Sabado" Mostrar "Dia Libre" Caso "Domingo" Mostrar "Dia Libre antes de una nueva semana" De Otro Modo Mostrar "Aun no es fin de semana" Fin Segun Fin Proceso </pre>	
Comprueba la solución en PSeInt	



3.6.2 Menú Funciones Matemáticas

Diseña un programa que lea un valor del usuario y muestra el menú para convertir este número con la función de redondeo o de valor absoluto.

Pasos a Seguir

1. Dar un nombre a este proceso: 'MenuFunMate'.
2. Defina tres variables de tipo 'Real'.
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese cualquier numero:'
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del valor al usuario.
5. Use la estructura de control 'Según' la 'variable' Hacer
 - a. Caso 1
 - b. Asigne el resultado del proceso redon(variable) a la variable 'resultado'.
 - c. Caso 2
 - d. Asigne el resultado del proceso abs(variable) a la variable 'resultado'.
 - e. De otro Modo
 - f. Mostrar 'Opcion no valida, intente de nuevo'.
 - g. Fin Según
6. Mostrar 'El resultado de la variable es: 'resultado''.
7. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Proceso MenuFuncMate 1 Definir X, elige, resultado Como Real 2 Mostrar 'Ingrese cualquier número' 3 Leer X 4 Mostrar 'Elija una opción: 1 = Redondear o 2 = Valor Absoluto' 5 Segun elige Hacer 6 1: resultado = redon(X) 7 2: resultado = abs(X) 8 De Otro Modo: 9 Mostrar 'Digito no valido, intente de nuevo' 10 Fin Segun 11 12 Mostrar 'El resultado de X es: ', resultado 13 FinProceso </pre>	
Comprueba la solución en PSeInt	

3.6.3 Adivina el número

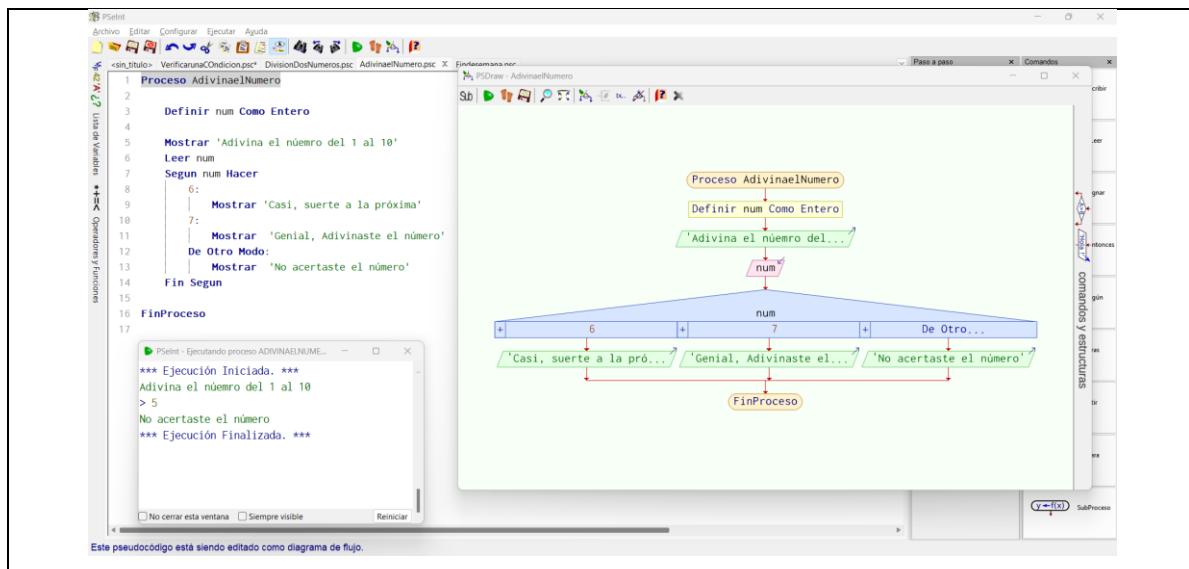
Programa que lea un número del 1 al 10 y muestre los siguientes mensajes 6 = 'Casi, suerte a la próxima', 7 = 'Genial, Adivinaste el número' , De otro modo = 'No acertaste el número'.

Pasos a Seguir

- Dar un nombre a este proceso: 'AdivinaelNúmero'.

9. Defina una variable de tipo 'Carácter'.
10. Utilice 'Mostrar' para mostrar texto al usuario: 'Adivina el numero del 1 al 10 '
11. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del valor al usuario.
12. Use la estructura de control 'Según' la 'variable' Hacer
 - a. Caso: 6
 - b. Mostrar 'Casi, suerte a la proxima'.
 - c. Caso 7
 - d. Mostrar 'Genial, adivinaste el numero'.
 - e. De otro Modo
 - f. Mostrar 'No acertaste el numero'.
 - g. Fin Segun
13. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Proceso AdivinaelNúmero Definir num Como ENTERO Mostrar "Adivina el numero del 1 al 10" Leer num Segun num Hacer Caso 6 Mostrar "Casi, suerte a la proxima" Caso 7 Mostrar "Genial, adivinaste el numero" De Otro Modo Mostrar "No acertaste el numero" Fin Segun Fin Proceso </pre>	<pre> graph TD Start([Proceso AdivinaelNúmero]) --> Def[Definir num Como ENTERO] Def --> Mostrar1[Mostrar "Adivina el numero del 1 al 10"] Mostrar1 --> Leer[Leer num] Leer --> Segun[Segun num Hacer] Segun --> Caso6[Caso 6 Mostrar "Casi, suerte a la proxima"] Segun --> Caso7[Caso 7 Mostrar "Genial, adivinaste el numero"] Segun --> DeOtroModo[De otro modo Mostrar "No acertaste el numero"] Caso6 --> FinProceso([Fin Proceso]) Caso7 --> FinProceso DeOtroModo --> FinProceso </pre>
Comprueba la solución en PSeInt	



3.6.4 Menú de operaciones aritméticas

Escribe un programa que lea una opción (1 = Sumar, 2 = Restar) y realice la operación correspondiente entre dos números definidos por el usuario.

Pasos a Seguir

1. Dar un nombre a este proceso: 'MenuOpeAritmeticas'.
2. Defina tres variables de tipo 'Real'.
3. Asigne un valor numérico a la 'variable1'.
4. Asigne un valor numérico a la 'variable2'.
5. Utilice 'Mostrar' para mostrar texto al usuario: 'Seleccione una opción 1 (Suma) o 2 (Resta)'.
6. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del valor al usuario.
7. Use la estructura de control 'Según' la 'variable' Hacer
 - a. Caso 1
 - b. Asigne el resultado del proceso 'variable1' + 'variable2' a la variable 'resultado'.
 - c. Caso 2
 - d. Asigne el resultado del proceso 'variable1' – 'variable2' a la variable 'resultado'.
 - e. De otro Modo
 - f. Mostrar 'Opcion no valida, intente de nuevo'.

- g. Fin Según
8. Mostrar 'El resultado de la operación es: ' resultado.
 9. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
<pre> Proceso OperacionesAritmeticas 1 Definir opcion, numero1, numero2, resultado Como REAL 2 numero1 = 50 3 numero2 = 10 4 Mostrar 'Seleccione una opción 1 (Suma) y 2 (Resta)' 5 Leer opcion 6 Según opcion Hacer 7 1: resultado ← numero1+numero2 8 2: resultado ← numero1-numero2 9 De Otro Modo: 10 Mostrar 'Opción no válida.' 11 FinSegún 12 13 Mostrar 'El resultado es: ', resultado 14 FinProceso </pre>	
Comprueba la solución en PSeInt	

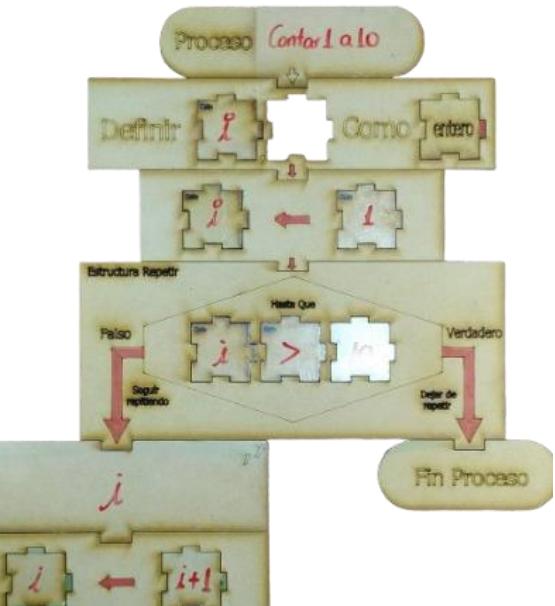
3.7 Ejercicios con 'REPETIR'

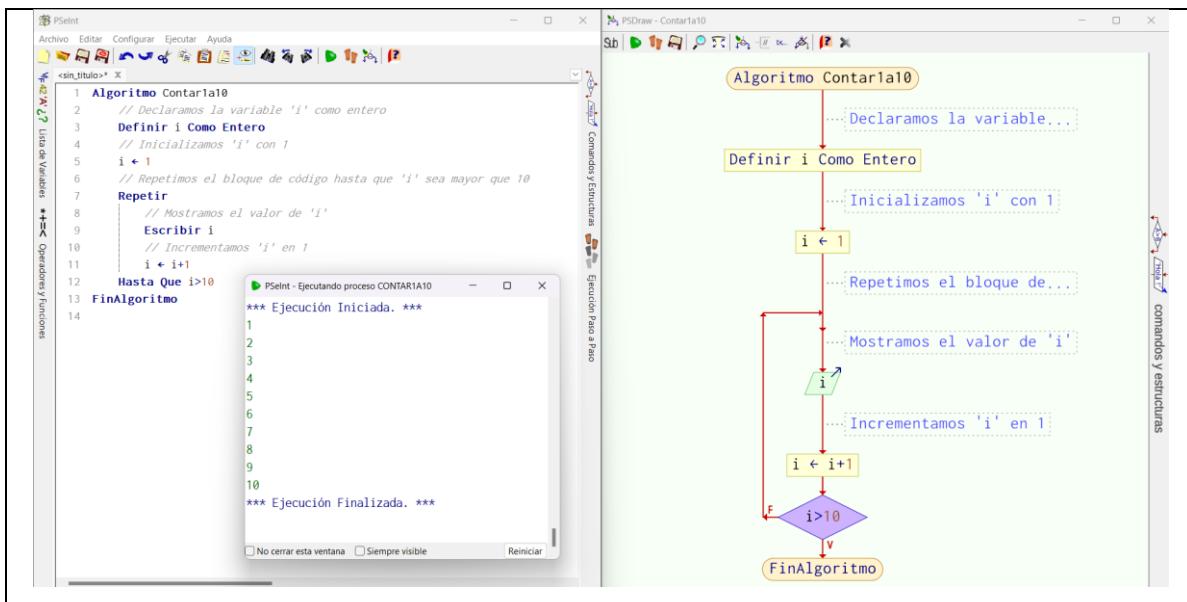
3.7.1 Contar del 1 al 10

Crea un programa que cuente del 1 al 10 utilizando una estructura repetitiva.

Pasos a Seguir:

1. Dar un nombre a este proceso: 'Contar1a10'.
2. Defina una variable de tipo 'Entero' llamada i.
3. Inicialice 'i' con 1.
4. Utilice la estructura de control 'Repetir' hasta que i sea mayor que 10.
 - a. Mostrar el valor de 'i'.
 - b. Incrementar 'i' en 1.
5. Finalice el proceso.

Pseudocódigo	Diagrama de Flujo
<pre> PROCESO Contar1a10 Definir i Como Entero i = 1 Repetir Mostrar i i = i + 1 Hasta Que i > 10 Fin Proceso </pre>	
Comprueba la solución en PSeInt	



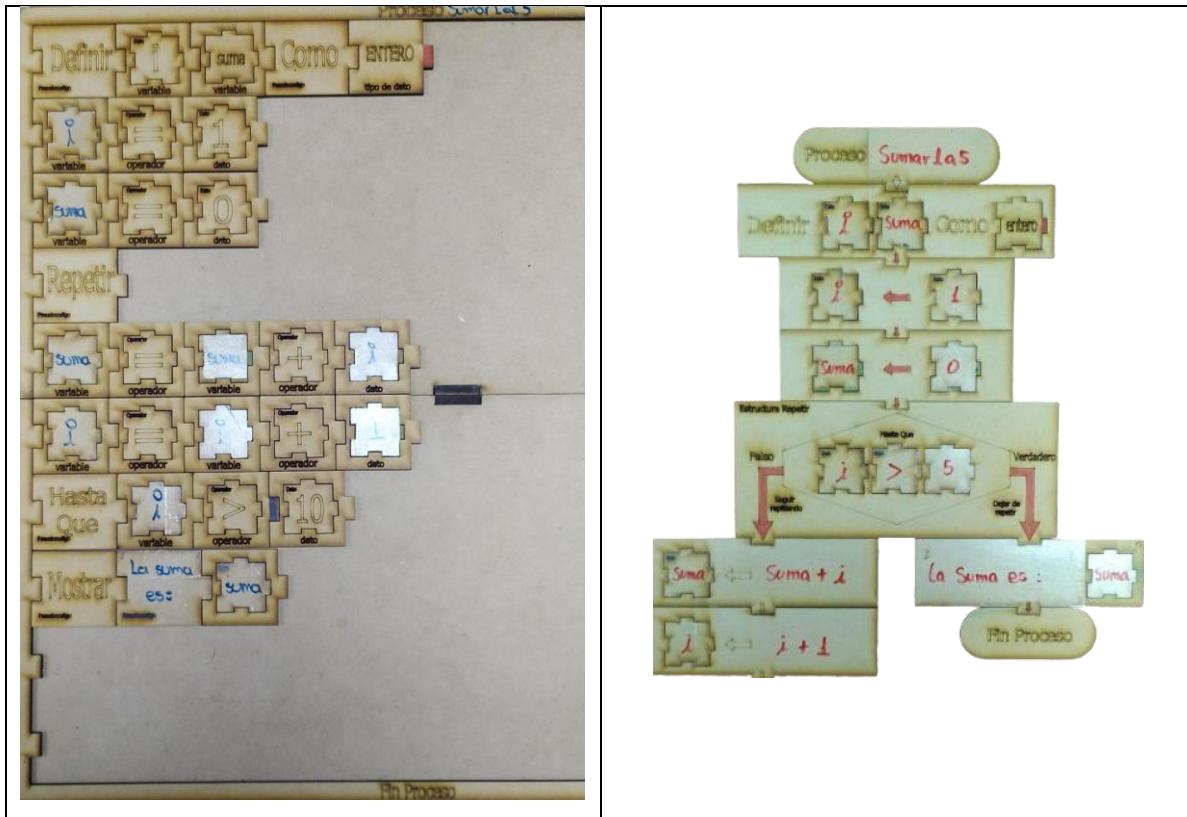
3.7.2 Sumar del 1 al 5

Crea un programa que sume los números del 1 al 5 utilizando una estructura repetitiva.

Pasos a Seguir:

1. Dar un nombre a este proceso: 'Sumar1a5'.
2. Defina dos variables de tipo 'Entero' llamadas 'i' y 'suma'.
3. Inicialice 'i' con 1.
4. Inicialice 'suma' con 0.
5. Utilice la estructura de control 'Repetir' hasta que 'i' sea mayor que 5.
 - a. Sumar el valor de 'i' a 'suma'.
 - b. Incrementar 'i' en 1.
6. Mostrar el resultado de la suma con el mensaje 'La suma es:'.
7. Finalice el proceso.

Pseudocódigo	Diagrama de Flujo
--------------	-------------------



Comprueba la solución en PSeInt

The screenshot shows the PSeInt interface with the following details:

- Left Panel (Code):**

```

1 Algoritmo Sumar1a5
2 // Declaramos las variables 'i' y 'suma' como enteros
3 Definir i, suma Como Entero
4 // Inicializamos 'i' con 1
5 i ← 1
6 // Inicializamos 'suma' con 0
7 suma ← 0
8 // Repetimos el bloque de código hasta que 'i' sea mayor que 5
9 Repetir
10   // Sumamos el valor de 'i' a 'suma'
11   suma ← suma+i
12   // Incrementamos 'i' en 1
13   i ← i+1
14 Hasta Que i>5
15 // Mostramos el resultado de la suma
16 Escribir ['La suma es: ', suma]
17 FinAlgoritmo
  
```
- Middle Panel (Execution Results):**

PSelint - Ejecutando proceso SUMAR1A5

*** Ejecución Iniciada. ***
La suma es: 15
*** Ejecución Finalizada. ***
- Right Panel (Flowchart):**

Algoritmo Sumar1a5

```

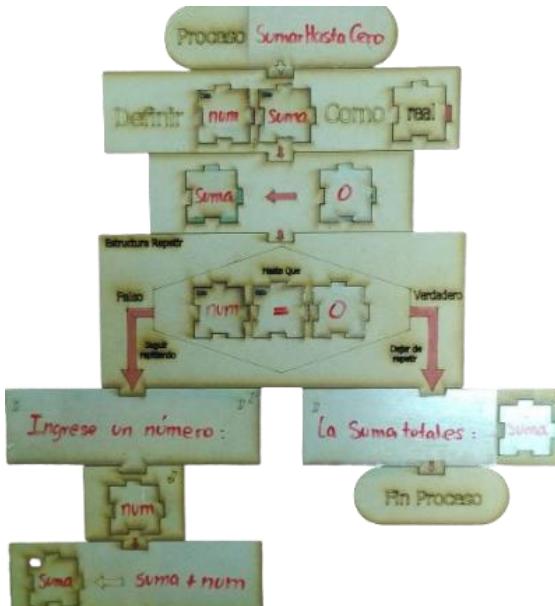
    Start
    ↓ Declaramos las variables...
    ↓ Definir i, suma Como E...
    ↓ i ← 1
    ↓ suma ← 0
    ↓ Repetimos el bloque de...
    ↓ i > 5?
    ↓ F: suma ← suma+i
    ↓ i ← i+1
    ↓ V: Mostramos el resultado...
    ↓ FinAlgoritmo
  
```

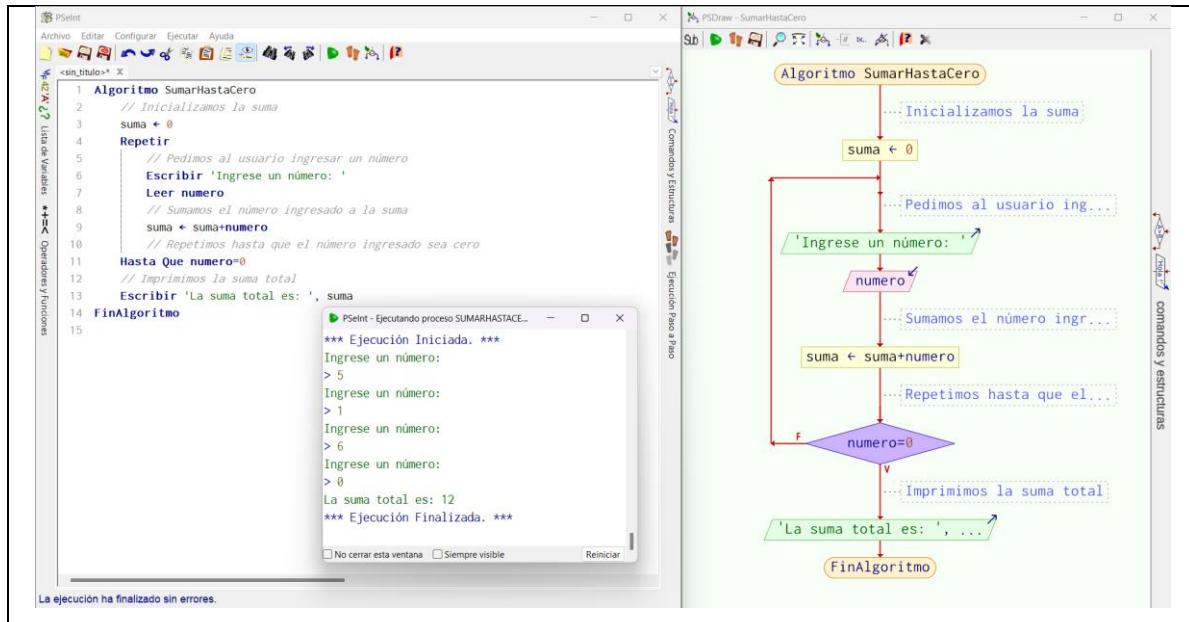
3.7.3 Sumar hasta ingresar cero

Crea un programa que sume números ingresados por el usuario hasta que se ingrese el número cero.

Pasos a Seguir:

1. Dar un nombre a este proceso: 'SumarHastaCero'.
2. Inicialice una variable 'suma' con 0.
3. Utilice la estructura de control 'Repetir' hasta que el número ingresado sea igual a 0.
 - a. Mostrar el mensaje 'Ingrese un número:' para pedir al usuario un número.
 - b. Leer el número ingresado por el usuario y guardarlo en una variable llamada 'numero'.
 - c. Sumar el valor de 'numero' a 'suma'.
8. Mostrar el resultado de la suma con el mensaje 'La suma total es:' .
9. Finalice el proceso.

Pseudocódigo	Diagrama de Flujo
<pre> PROCESO SumarHastaCero Definir num, suma Como REAL suma ← 0 Repetir Mostrar "Ingrese un número" Leer num suma ← suma + num Hasta Que num = 0 Mostrar "La suma total es: " suma Fin Repetir Fin Proceso </pre>	
Comprueba la solución en PSeInt	



3.7.4 Verificar contraseña

Crea un programa que pida al usuario ingresar una contraseña hasta que sea correcta.

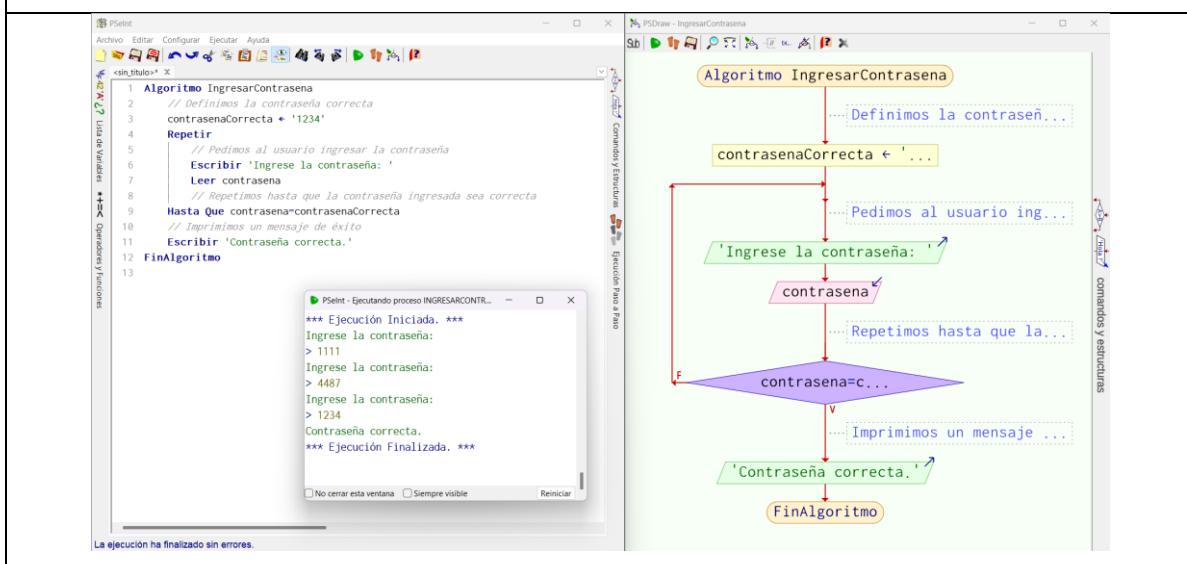
Pasos a Seguir:

1. Dar un nombre a este proceso: 'IngresarContrasena'.
2. Defina una variable 'contraseñaCorrecta' con el valor '1234'.
3. Utilice la estructura de control 'Repetir' hasta que la contraseña ingresada sea igual a 'contraseñaCorrecta'.
 - a. Mostrar el mensaje 'Ingrese la contraseña:' para pedir al usuario una contraseña.
 - b. Leer la contraseña ingresada por el usuario y guardarla en una variable llamada 'contraseña'.
10. Mostrar el mensaje 'Contraseña correcta.' cuando la contraseña ingresada sea correcta.
11. Finalice el proceso.

Pseudocódigo	Diagrama de Flujo
---------------------	--------------------------



Comprueba la solución en PSelint



3.8 Ejercicios con 'MIENTRAS'

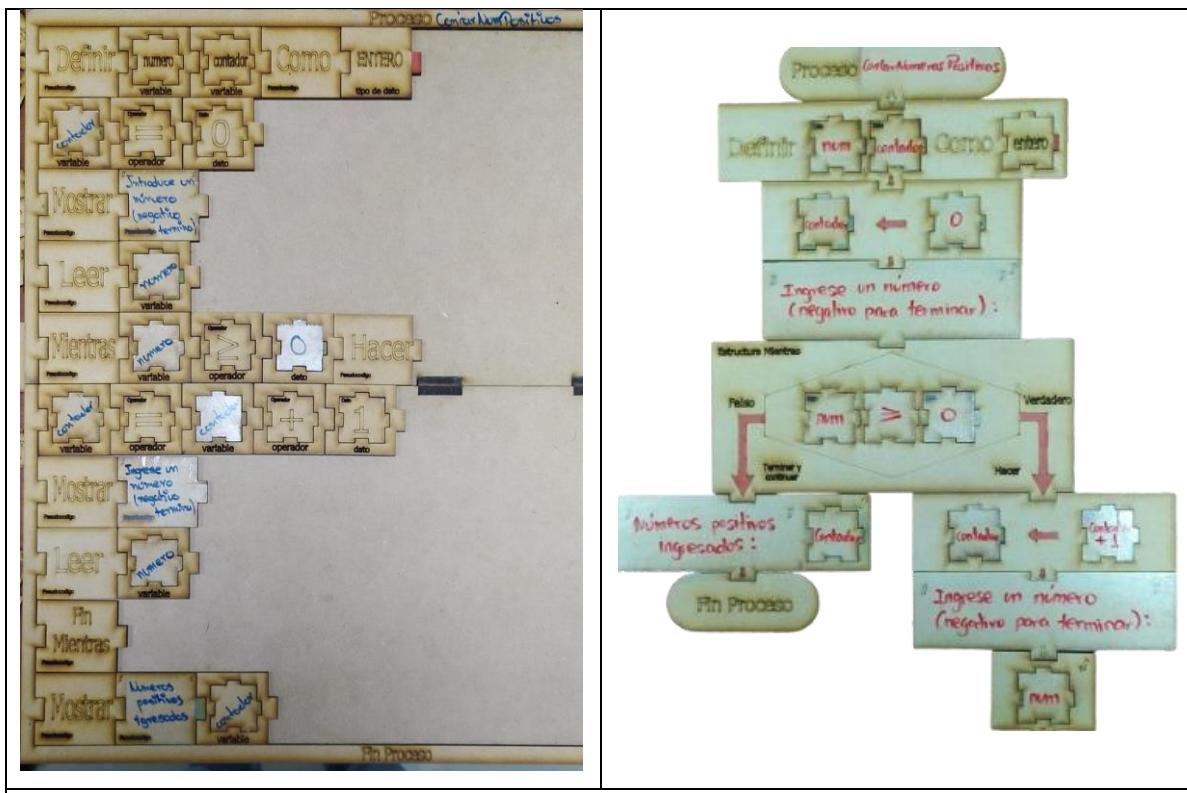
3.8.1 Contar números positivos

Crea un programa que cuente la cantidad de números positivos ingresados por el usuario hasta que se ingrese un número negativo.

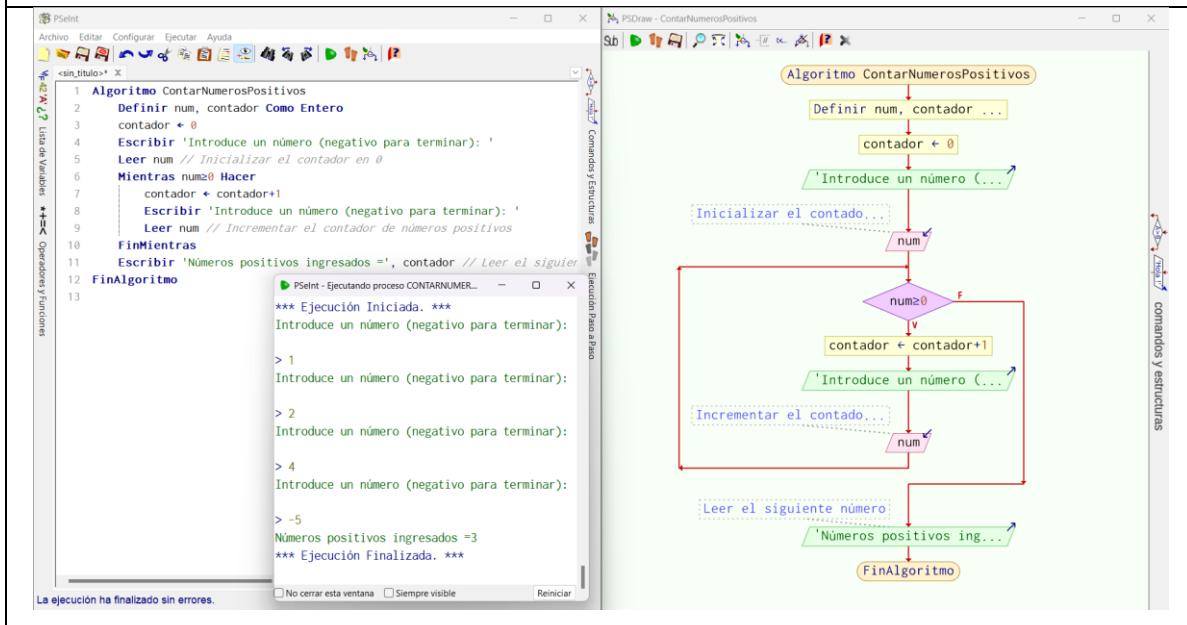
Pasos a Seguir:

1. Dar un nombre a este proceso: 'ContarNumerosPositivos'.
2. Defina dos variables de tipo 'Entero' llamadas 'num' y 'contador'.
3. Inicialice 'contador' con 0.
4. Muestre el mensaje 'Introduce un número (negativo para terminar):' para pedir al usuario un número.
5. Lea el número ingresado por el usuario y guárdelo en la variable 'num'.
6. Utilice la estructura de control 'Mientras' para repetir el siguiente bloque de código mientras 'num' sea mayor o igual a 0.
 - a. Incremente 'contador' en 1.
 - b. Muestre el mensaje 'Introduce un número (negativo para terminar):' para pedir otro número al usuario.
 - c. Lea el nuevo número ingresado por el usuario y guárdelo en la variable 'num'.
12. Muestre el resultado con el mensaje 'Números positivos ingresados =' seguido del valor de 'contador'.
13. Finalice el proceso.

Pseudocódigo	Diagrama de Flujo
--------------	-------------------



Comprueba la solución en PSeInt



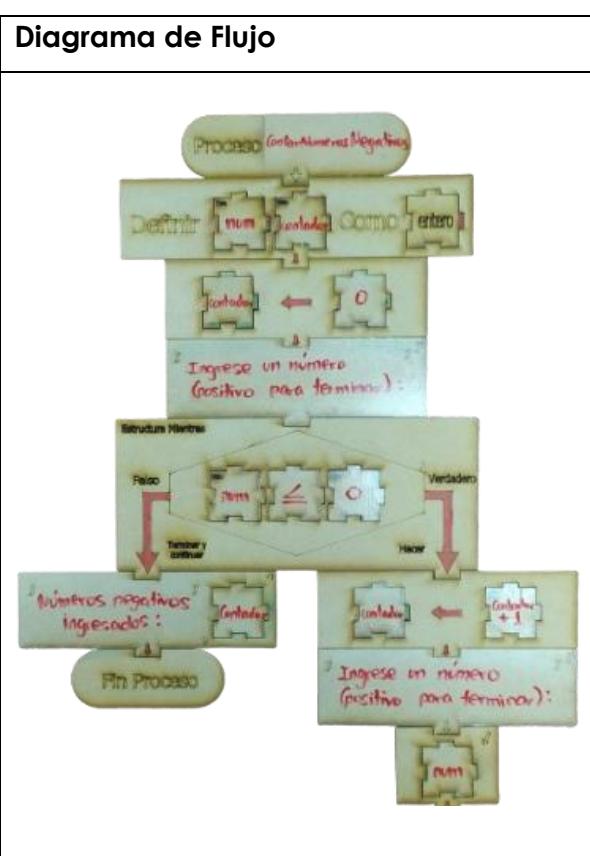
Contar números negativos

Crea un programa que cuente la cantidad de números negativos ingresados por el usuario hasta que se ingrese un número positivo.

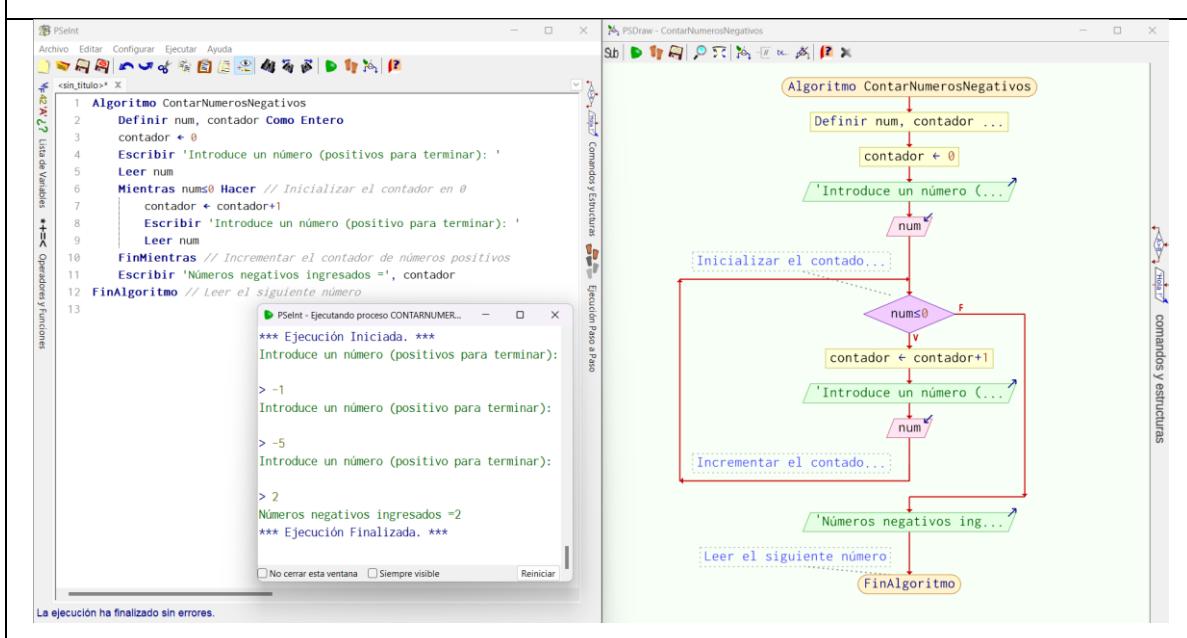
Pasos a Seguir:

1. Dar un nombre a este proceso: 'ContarNumerosNegativos' .

2. Defina dos variables de tipo 'Entero' llamadas 'num' y 'contador'.
3. Inicialice 'contador' con 0.
4. Muestre el mensaje 'Introduce un número (positivo para terminar):' para pedir al usuario un número.
5. Lea el número ingresado por el usuario y guárdelo en la variable 'num'.
6. Utilice la estructura de control 'Mientras's para repetir el siguiente bloque de código mientras 'num' sea menor o igual a 0.
 - a. Incremente 'contador' en 1.
 - b. Muestre el mensaje 'Introduce un número (positivo para terminar):' para pedir otro número al usuario.
 - c. Lea el nuevo número ingresado por el usuario y guárdelo en la variable 'num'.
14. Muestre el resultado con el mensaje 'Números negativos ingresados =' seguido del valor de 'contador'.
15. Finalice el proceso.

Pseudocódigo	Diagrama de Flujo
<pre> Proceso ContarNumerosNegativos Definir num, contador Como ENTERO num = 0 Mostrar "Introduce un numero (positivo para terminar):" Leer num Mientras num <= 0 Hacer contador = contador + 1 Mostrar "Introduce un numero (positivo para terminar):" Leer num Fin Mientras Mostrar "Numeros negativos ingresados: " & contador </pre>	

Comprueba la solución en PSelnt



The screenshot shows the PSelnt environment with three main windows:

- Pseudocode Window:** Displays the pseudocode for the algorithm "ContarNumerosNegativos".
- Flowchart Window:** Shows the flowchart corresponding to the pseudocode.
- Execution Window:** Shows the output of the program execution, including user input and system responses.

```

1 Algoritmo ContarNumerosNegativos
2   Definir num, contador Como Entero
3   contador ← 0
4   Escribir 'Introduce un número (positivos para terminar): '
5   Leer num
6   Mientras num<=0 Hacer // Inicializar el contador en 0
7     contador ← contador+1
8     Escribir 'Introduce un número (positivo para terminar): '
9     Leer num
10    FinMientras // Incrementar el contador de números positivos
11   Escribir 'Números negativos ingresados =', contador
12  FinAlgoritmo // Leer el siguiente número
13

```

Execution Window Output:

```

*** Ejecución Iniciada. ***
Introduce un número (positivos para terminar):
> -1
Introduce un número (positivo para terminar):
> -5
Introduce un número (positivo para terminar):
> 2
Números negativos ingresados =2
*** Ejecución Finalizada. ***

```

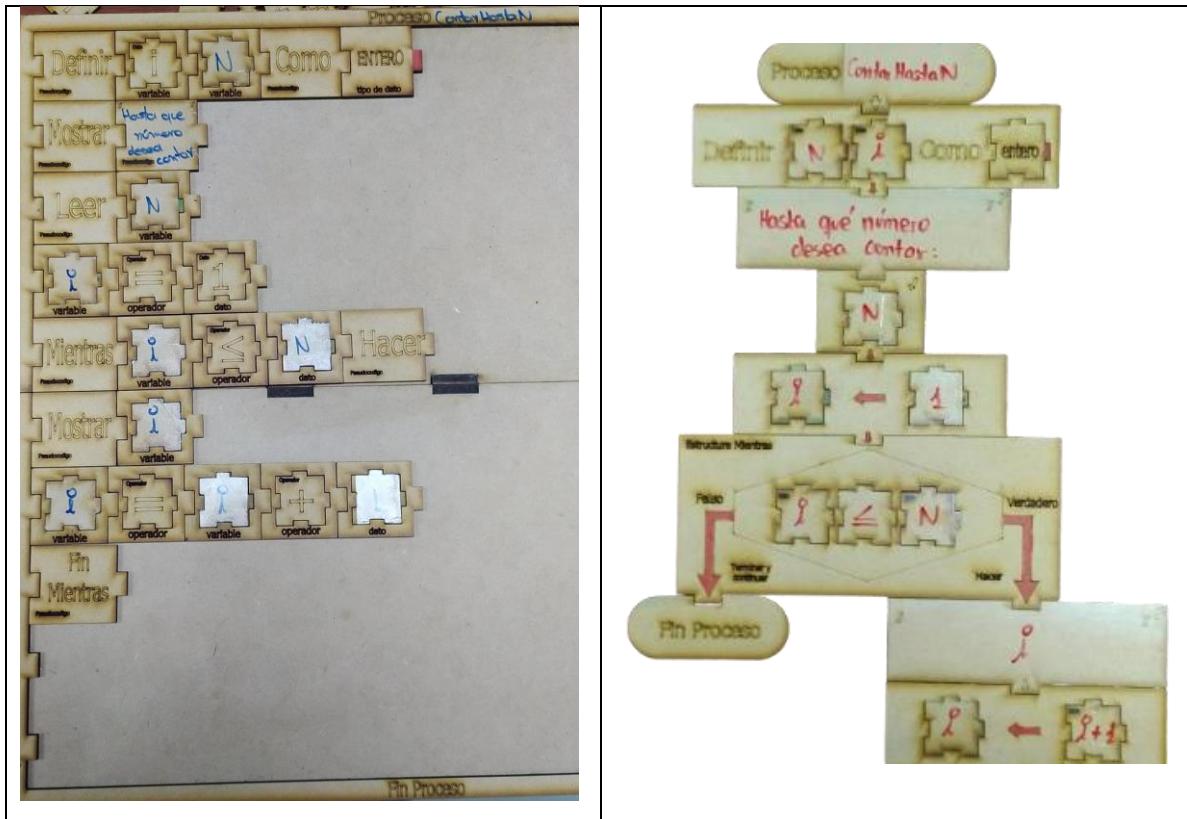
3.8.3 Contar hasta N

Crea un programa que cuente desde 1 hasta un número positivo N ingresado por el usuario.

Pasos a Seguir:

1. Dar un nombre a este proceso: 'ContarHastaN'.
2. Defina dos variables de tipo 'Entero' llamadas 'i' y 'N'.
3. Muestre el mensaje 'Introduce un número positivo N:' para pedir al usuario un número positivo.
4. Lea el número ingresado por el usuario y guárdelo en la variable 'N'.
5. Inicialice 'i' con 1.
6. Utilice la estructura de control 'Mientras' para repetir el siguiente bloque de código mientras 'i' sea menor o igual a 'N'.
 - a. Mostrar el valor de 'i'.
 - b. Incrementar 'i' en 1.
16. Finalice el proceso.

Pseudocódigo	Diagrama de Flujo
--------------	-------------------



Comprueba la solución en PSeInt

The left window shows the PSeInt code for the "ContarHastaN" algorithm:

```

1 Algoritmo ContarHastaN
2   Definir i, N Como Entero
3   Escribir 'Introduce un número positivo N: '
4   Leer N
5   i ← 1 // Leer el número N
6   Mientras i≤N Hacer // Inicializar el contador en 1
7     Escribir i
8     i ← i+1 // Mostrar el valor de i
9   FinMientras // Incrementar el contador
10 FinAlgoritmo
11

```

The right window shows the execution of the algorithm. The flowchart is displayed with arrows indicating the execution path. The execution window shows the output: "Introduce un número positivo N: > 5" followed by the numbers 1, 2, 3, 4, 5. The status bar at the bottom says "La ejecución ha finalizado sin errores."

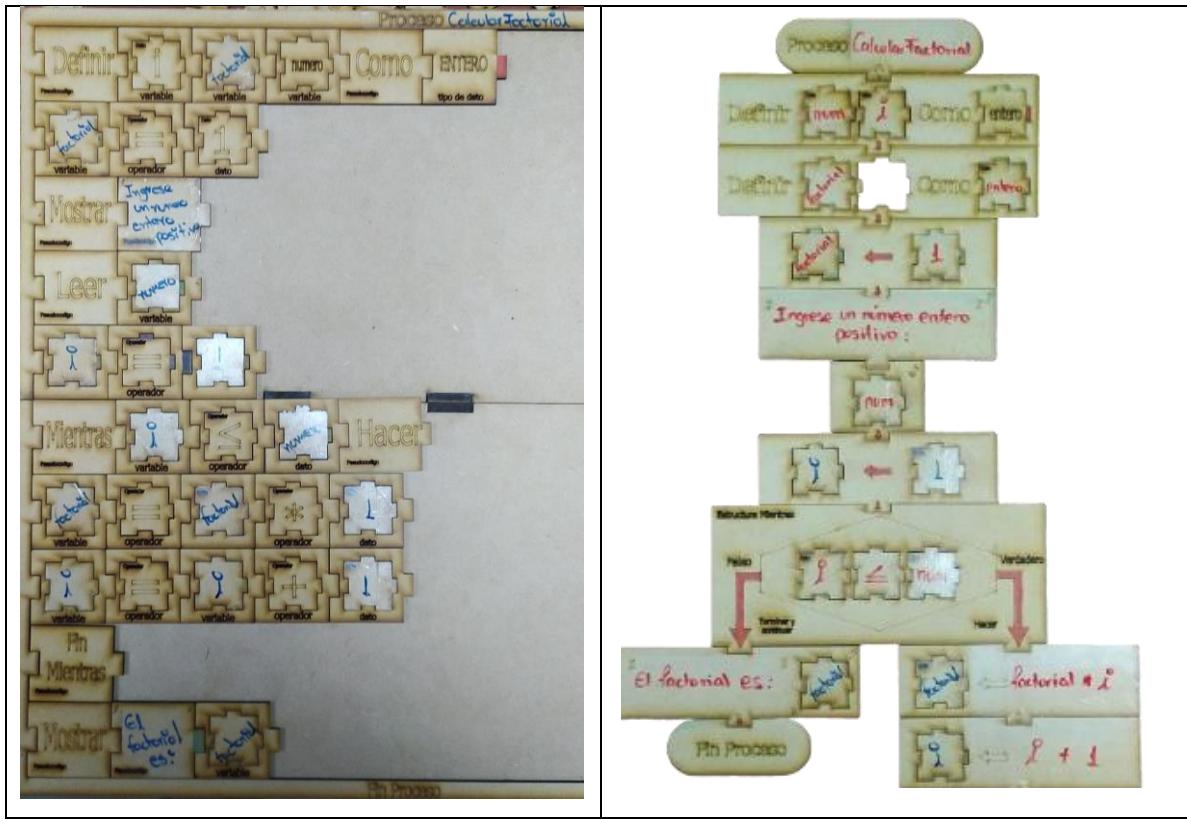
3.8.4 Calcular factorial

Crea un programa que calcule la factorial de un número entero positivo ingresado por el usuario.

Pasos a Seguir:

1. Dar un nombre a este proceso: 'CalcularFactorial'.
2. Defina tres variables de tipo 'Entero' llamadas 'num', 'i' y 'factorial'.
3. Inicialice 'factorial' con 1.
4. Muestre el mensaje 'Introduce un número entero positivo:' para pedir al usuario un número.
5. Lea el número ingresado por el usuario y guárdelo en la variable 'numero'.
6. Inicialice 'i' con 1.
7. Utilice la estructura de control 'Mientras' para repetir el siguiente bloque de código mientras 'i' sea menor o igual a 'numero'.
 - a. Multiplicar 'factorial' por 'i' y guardar el resultado en 'factorial'.
 - b. Incrementar 'i' en 1.
17. Mostrar el resultado del factorial con el mensaje 'El factorial es:' seguido del valor de 'factorial'.
18. Finalice el proceso.

Pseudocódigo	Diagrama de Flujo
--------------	-------------------



Comprueba la solución en PSelint

Algoritmo CalcularFactorial

```

1 // Declarar variables
2 Definir num, i, factorial Como Entero
3 factorial ← 1
4 // Pedir el número
5 Escribir 'Introduce un número entero positivo: '
6 Leer numero // Inicializar el factorial en 1
7 i ← 1
8 // Bucle MIENTRAS para calcular el factorial
9 Mientras isnumero Hacer
10   factorial ← factorial*i // Inicializar el contador en 1
11   i ← i+1
12 FinMientras // Multiplicar el factorial por el contador
13 // Mostrar el factorial del número
14 Escribir 'El factorial es: ', factorial // Incrementar el contador
15
16 FinAlgoritmo
    
```

PSelint - Ejecutando proceso CALCULARFACTO... *** Ejecución Iniciada. *** Introduce un número entero positivo: > 5 El factorial es: 120 *** Ejecución Finalizada. ***

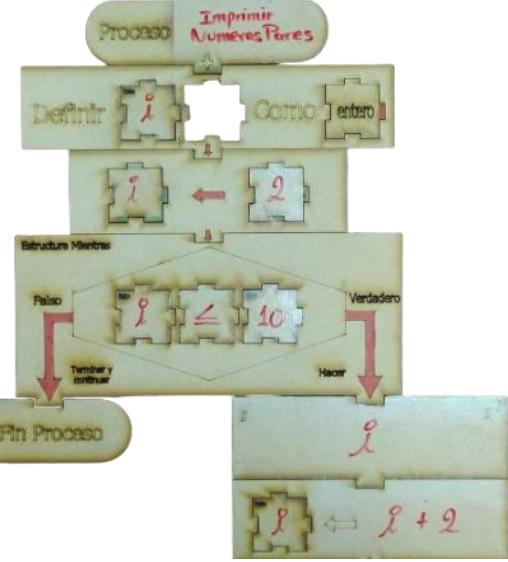
3.8.5 Imprimir números pares

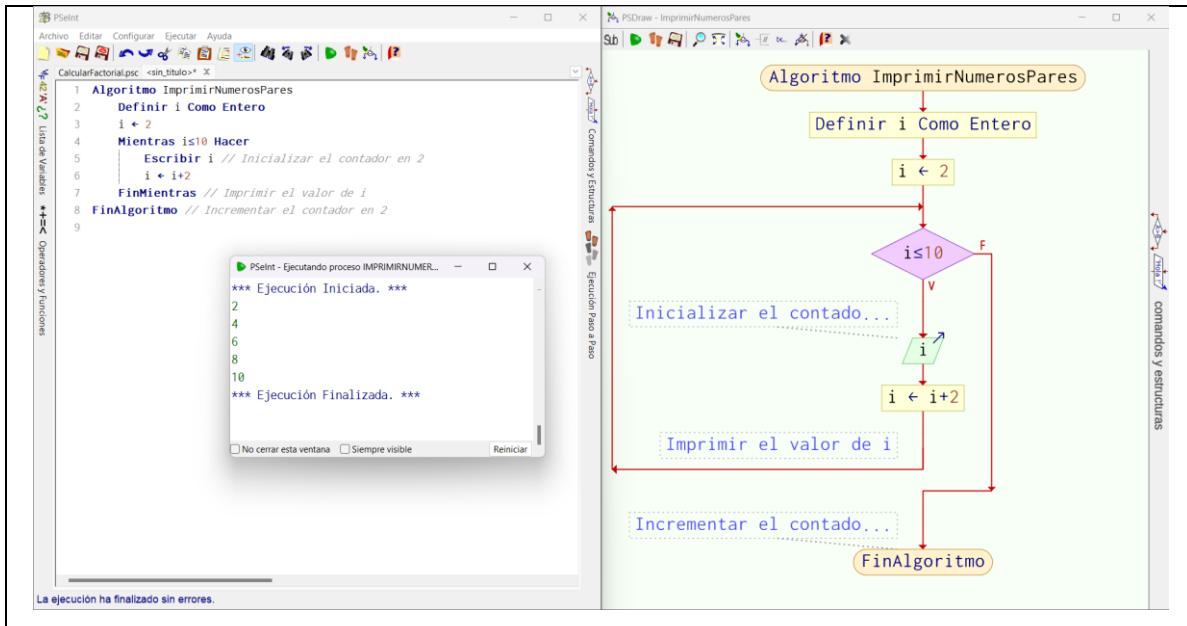
Crea un programa que imprima los números pares del 2 al 10.

Pasos a Seguir:

1. Dar un nombre a este proceso: 'ImprimirNumerosPares'.

2. Defina una variable de tipo 'Entero' llamada 'i'.
 3. Inicialice 'i' con 2.
 4. Utilice la estructura de control 'Mientras' para repetir el siguiente bloque de código mientras 'i' sea menor o igual a 10.
 - a. Mostrar el valor de 'i'.
 - b. Incrementar 'i' en 2.
19. Finalice el proceso.

Pseudocódigo	Diagrama de Flujo
 <pre> Proceso ImprimirNumerosPares Definir i Como entero i = 2 Mientras i <= 10 Hacer Mostrar i i = i + 2 Fin Mientras </pre>	
Comprueba la solución en PSeInt	



3.9 Ejercicios con 'PARA'

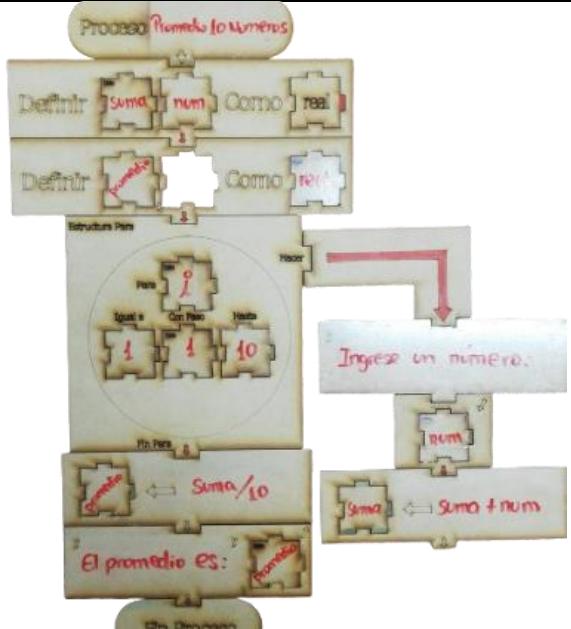
3.9.1 Calcular promedio de 10 números

Crea un programa que calcule el promedio de 10 números ingresados por el usuario.

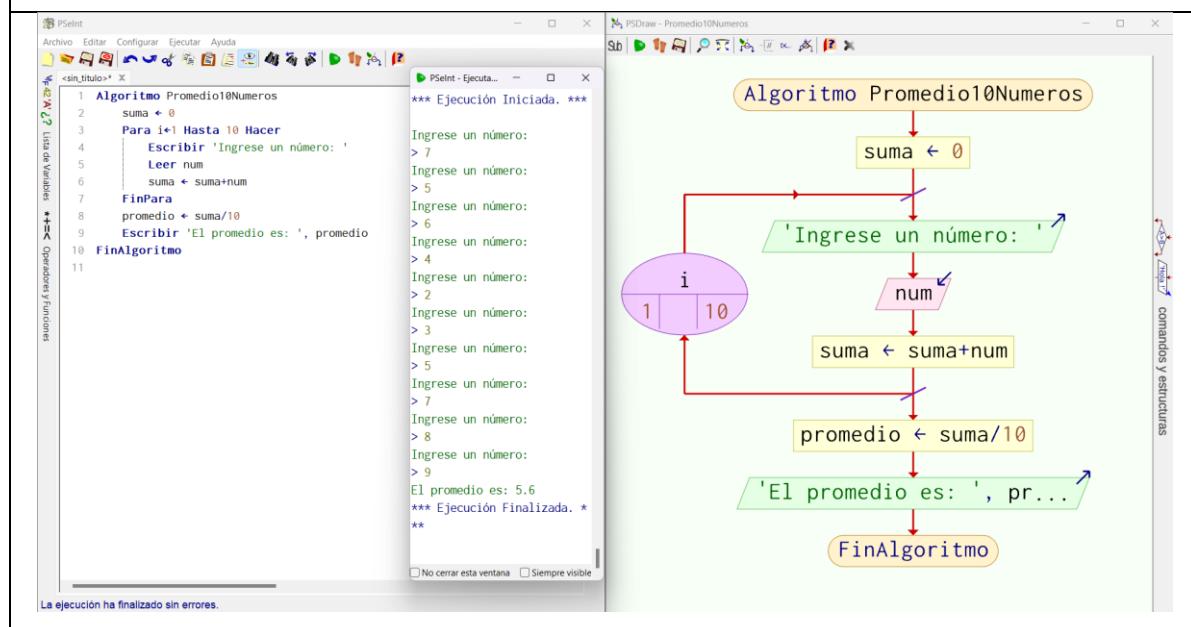
Pasos a Seguir:

1. Dar un nombre a este proceso: 'Promedio10Numeros'.
2. Inicializa la variable 'suma' en 0 para almacenar la suma de los números.
3. Utiliza un bucle 'Para' desde i igual a 1 hasta 10.
 - Muestra el mensaje 'Ingrese un número:' para pedir al usuario que ingrese un número.
 - Lee el número ingresado por el usuario y guárdalo en la variable 'num'.
 - Suma 'num' a la variable 'suma'.
4. Dentro del bucle:
 - Muestra el mensaje 'Ingrese un número:' para pedir al usuario que ingrese un número.
 - Lee el número ingresado por el usuario y guárdalo en la variable 'num'.
 - Suma 'num' a la variable 'suma'.
5. Después del bucle, calcula el promedio dividiendo 'suma' entre 10 y guárdalo en la variable 'promedio'.

6. Muestra el resultado del promedio con el mensaje 'El promedio es:' seguido del valor de 'promedio'.
7. Finaliza el proceso.

Pseudocódigo	Diagrama de Flujo
<pre> Proceso Definir suma Como REAL Definir i Como ENTERO Para i=1 Hasta 10 Hacer Leer num suma ← suma+num i=i+1 FinPara Promedio ← suma/10 Mostrar 'El promedio es:', promedio Fin Proceso </pre>	

Comprueba la solución en PSelint



```

1 Algoritmo Promedio10Numeros
2     suma ← 0
3     Para i=1 Hasta 10 Hacer
4         Escribir 'Ingrese un número: '
5         Leer num
6         suma ← suma+num
7     FinPara
8     promedio ← suma/10
9     Escribir 'El promedio es: ', promedio
10 FinAlgoritmo
11

```

** Ejecución Iniciada. **

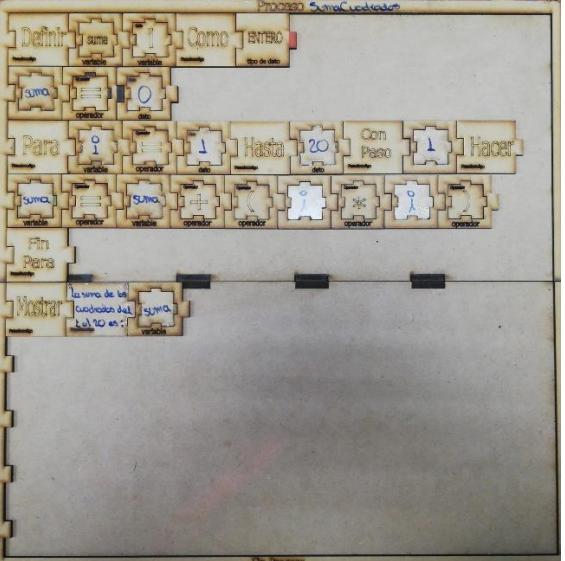
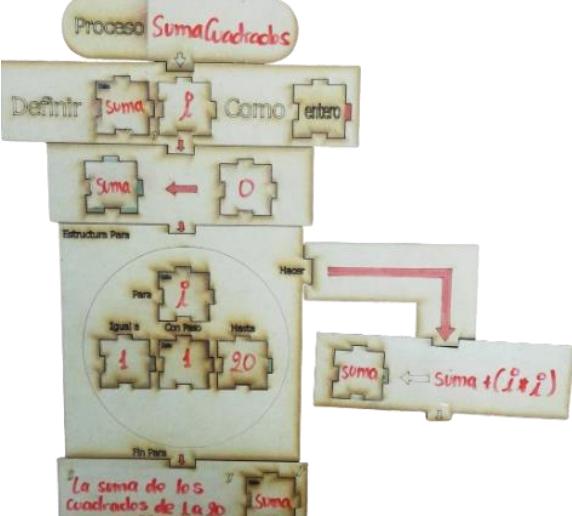
Ingrese un número:
> 7
Ingresar un número:
> 5
Ingresar un número:
> 6
Ingresar un número:
> 4
Ingresar un número:
> 2
Ingresar un número:
> 3
Ingresar un número:
> 5
Ingresar un número:
> 7
Ingresar un número:
> 8
Ingresar un número:
> 9
El promedio es: 5.6
** Ejecución Finalizada. **

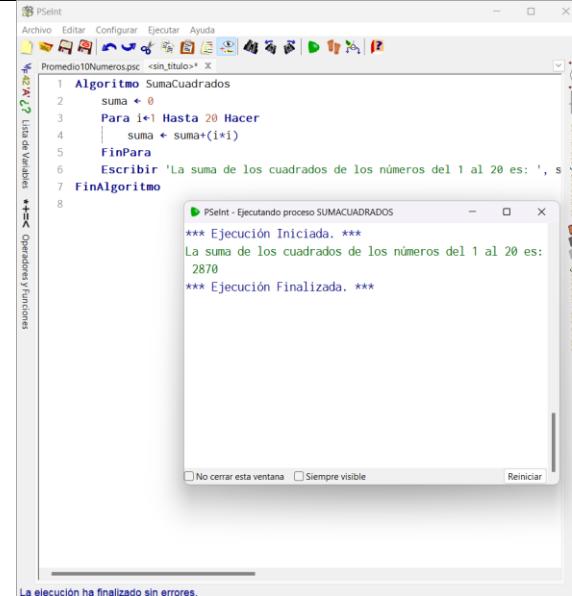
3.9.2 Suma de cuadrados del 1 al 20

Crea un programa que calcule la suma de los cuadrados de los números del 1 al 20.

Pasos a Seguir:

1. Dar un nombre a este proceso: 'SumaCuadrados'.
2. Inicializa la variable 'suma' en 0 para almacenar la suma de los cuadrados.
3. Utiliza un bucle 'Para' desde 'i' igual a 1 hasta 20.
4. Dentro del bucle:
 - a. Calcula el cuadrado de 'i' usando la expresión ' $i * i$ '.
 - b. Suma el resultado al valor actual de 'suma'.
5. Después del bucle, muestra el resultado de la suma de los cuadrados con el mensaje 'La suma de los cuadrados de los números del 1 al 20 es:' seguido del valor de 'suma'.
6. Finaliza el proceso.

Pseudocódigo	Diagrama de Flujo
 <pre> process "SumaCuadrados" define suma := 0 como entero para i = 1 hasta 20 con paso 1 hacer suma := suma + (i * i) fin para mostrar "La suma de los cuadrados de los números del 1 al 20 es:" & suma end </pre>	 <pre> Proceso SumaCuadrados Definir suma := 0 Como entero Para i = 1 hasta 20 Con Paso 1 Hacer suma := suma + (i * i) Fin Para Mostrar "La suma de los cuadrados de los números del 1 al 20 es:" & suma </pre>
Comprueba la solución en PSeInt	



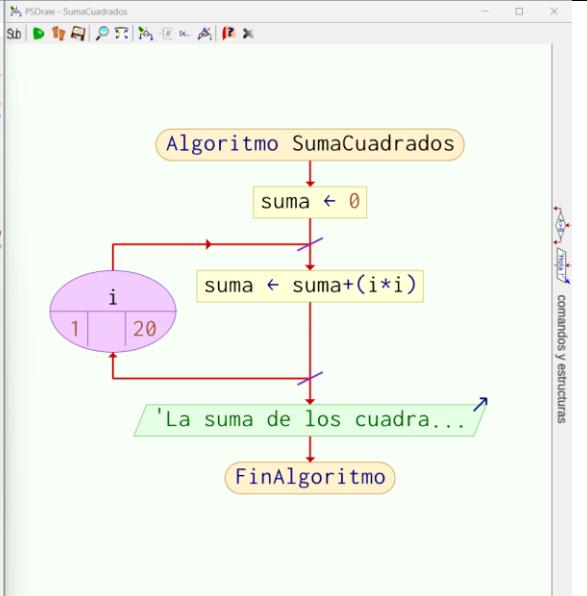
```

1 Algoritmo SumaCuadrados
2 suma ← 0
3 Para i←1 Hasta 20 Hacer
4   suma ← suma+(i*i)
5 FinPara
6 Escribir 'La suma de los cuadrados de los números del 1 al 20 es: ', s
7 FinAlgoritmo
  
```

PSeInt - Ejecutando proceso SUMACUADRADOS

*** Ejecución Iniciada. ***
La suma de los cuadrados de los números del 1 al 20 es:
2870
*** Ejecución Finalizada. ***

No cerrar esta ventana Siempre visible Reiniciar



```

graph TD
    Start([Algoritmo SumaCuadrados]) --> Suma0[suma ← 0]
    Suma0 --> Loop((i))
    Loop --> Cond{i | 1 | 20}
    Cond --> SumaSuma[suma ← suma+(i*i)]
    SumaSuma --> Output["'La suma de los cuadra..."]
    Output --> Fin[FinAlgoritmo]
    Fin --> End([comandos y estructuras])
  
```

La ejecución ha finalizado sin errores.