



FACULTAD DE INDUSTRIAS AGROPECUARIAS Y CIENCIAS
AMBIENTALES

Kit de Programación

¡Aprende 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

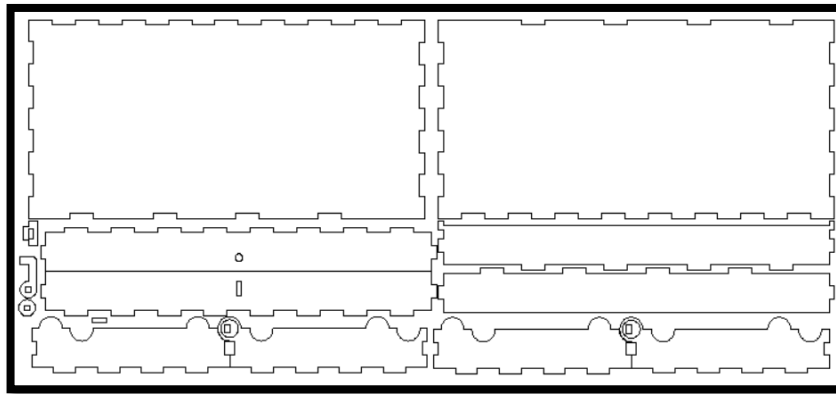
El fin de este curso es capacitar a los participantes en los fundamentos de la programación, proporcionándoles las habilidades y conocimientos necesarios para comprender y crear algoritmos simples, desarrollar programas básicos en pseudocódigo utilizando PSeInt, y resolver problemas de manera sistemática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

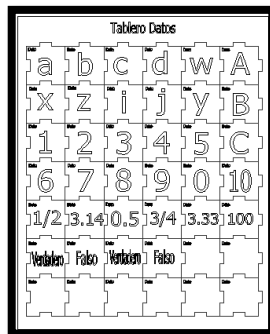
2.1 Caja Principal (Tablero)

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

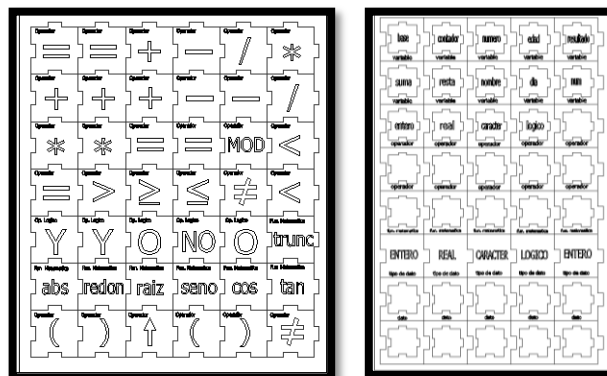


2.2 Datos

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.

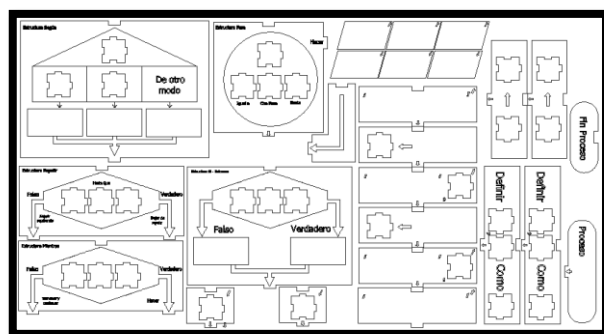
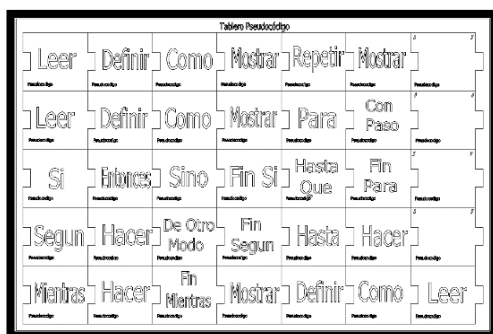
2.4 Tipos de Datos

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.



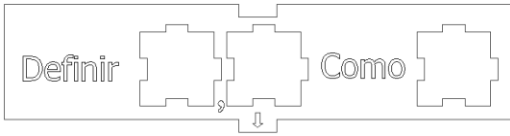
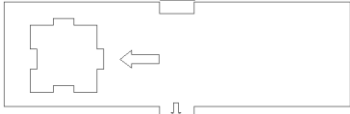
2.5.1 Inicio y Final

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 ↑	
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
Definir Como	
	



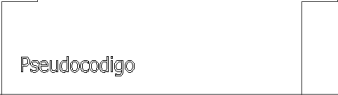



2.5.3 Datos de Entrada

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


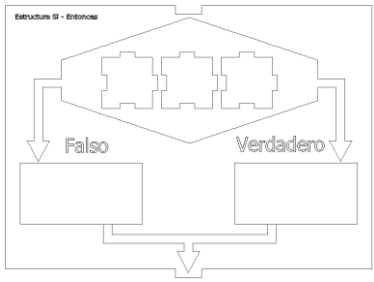

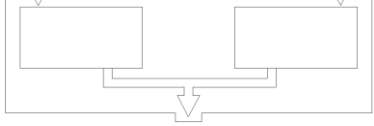
Pseudocódigo	Diagrama de Flujo
Leer	
	
	

2.5.4 Datos de Salida


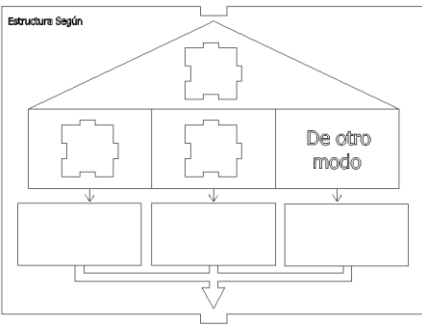

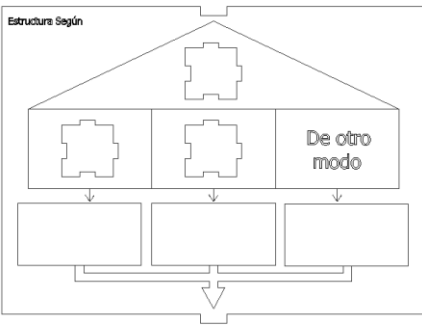
Indica la impresión de datos en la salida. Se representa con un recuadro con las esquinas inclinadas y una flecha hacia fuera.

Pseudocódigo	Diagrama de Flujo
	
	
	

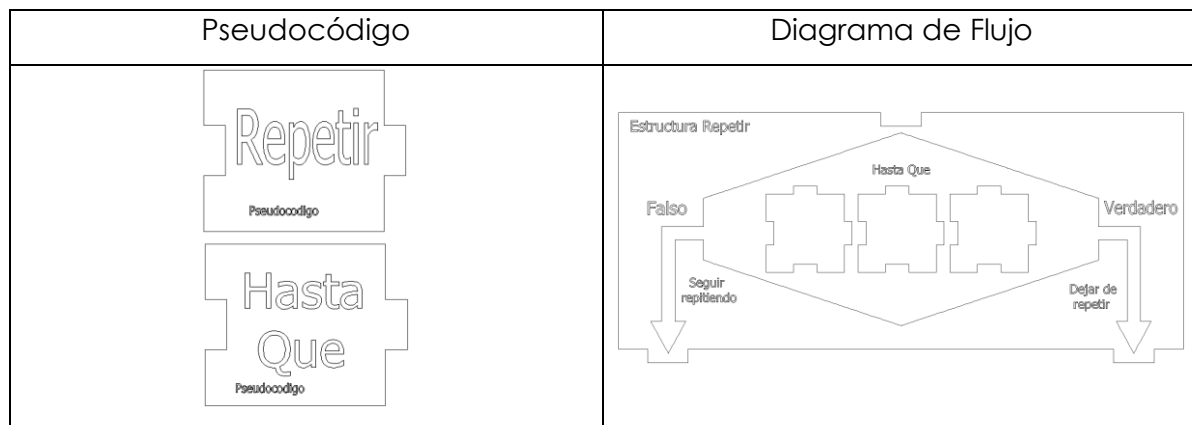
2.5.5 Estructura de Control - Si

Pseudocódigo	Diagrama de Flujo
	
	

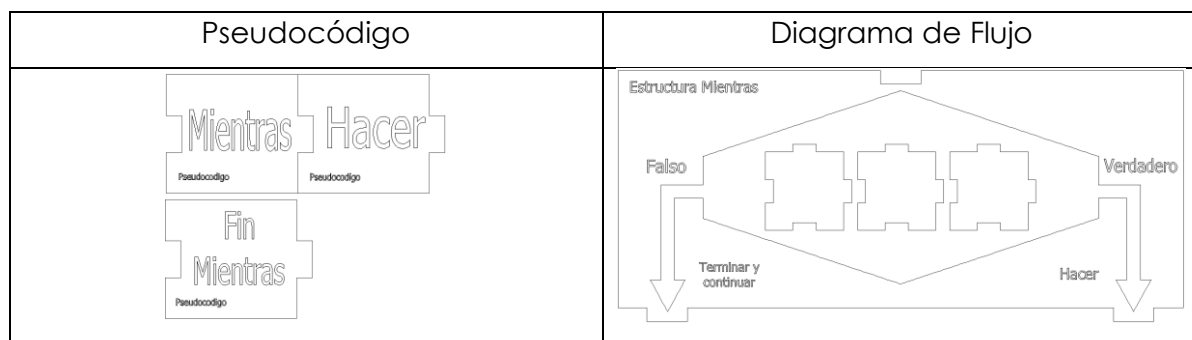
2.5.6 Estructura de Control - Segun

Pseudocódigo	Diagrama de Flujo
	
	

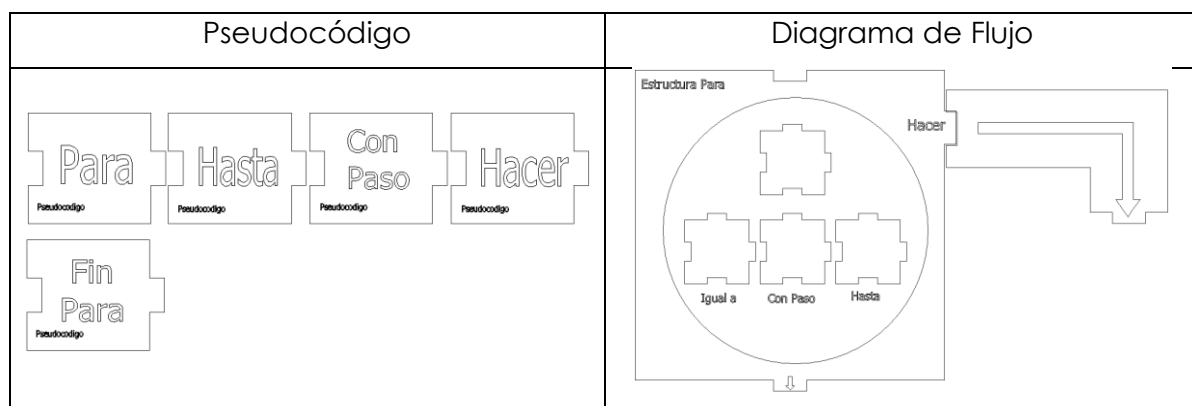
2.5.7 Estructura de Control - Repetir



2.5.8 Estructura de Control - Mientras



2.5.9 Estructura de Control - Para



3. EJERCICIOS

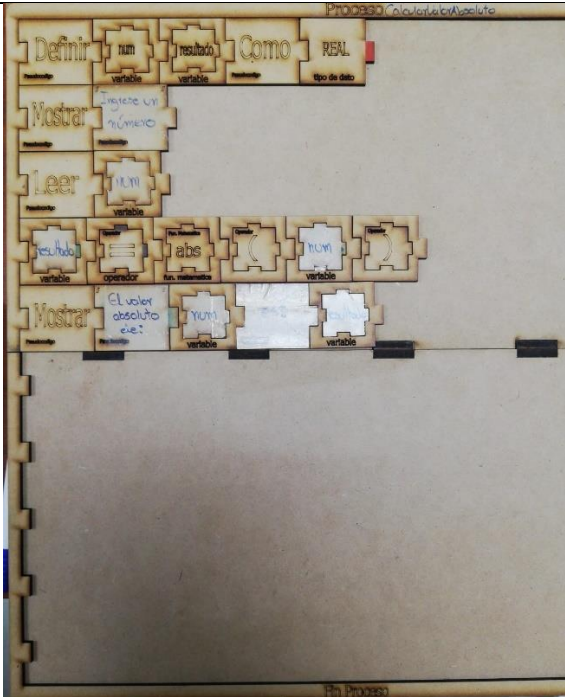
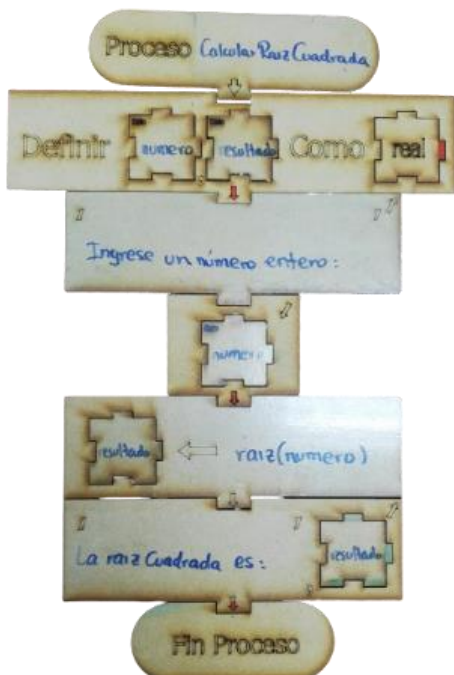
3.1 Funciones Matemáticas

3.1.1 Calcular la raíz cuadrada de un número

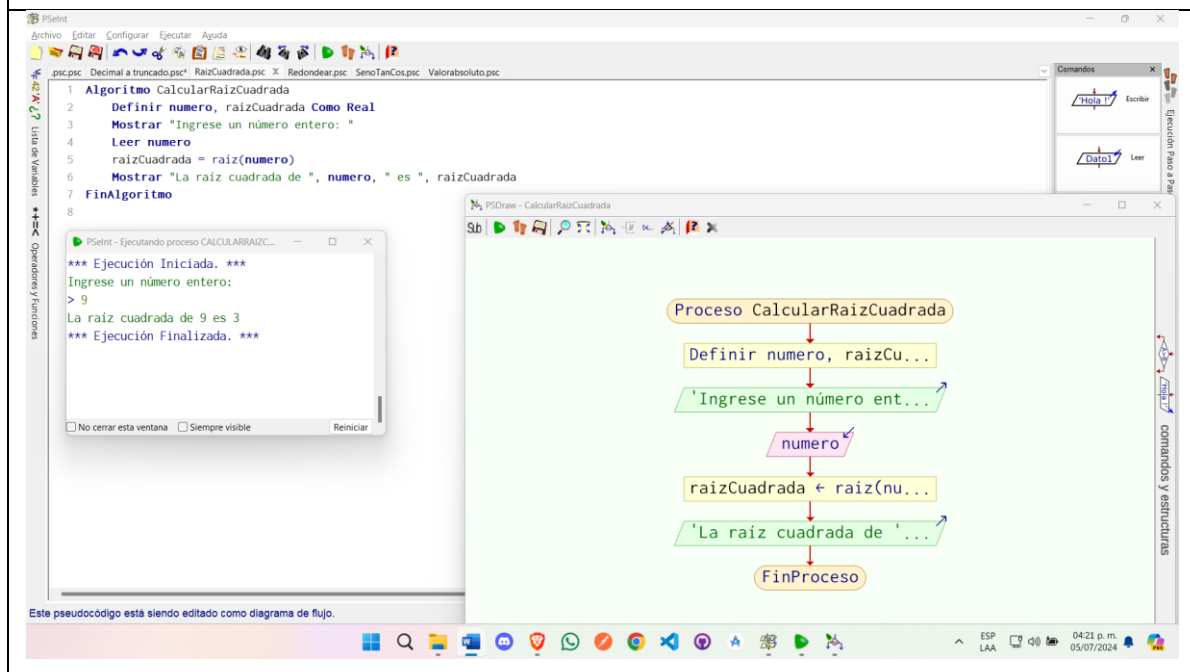
Escribe un programa que lea un número entero y calcule su raíz cuadrada utilizando la función 'raiz' de PSeInt.

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
	

Comprueba la solución en PSeInt



The screenshot shows the PSeInt software interface. On the left, the 'Pseudocódigo' editor displays the following code:

```

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
  
```

On the right, the 'Diagrama de Flujo' (Flowchart) editor shows a process diagram for 'CalcularRaizCuadrada'. The steps are:

- Proceso CalcularRaizCuadrada
- Definir numero, raizCu...
- 'Ingrese un número ent...
- numero
- raizCuadrada ← raiz(nu...
- 'La raíz cuadrada de '...
- FinProceso

3.1.2 Cálculo del valor absoluto

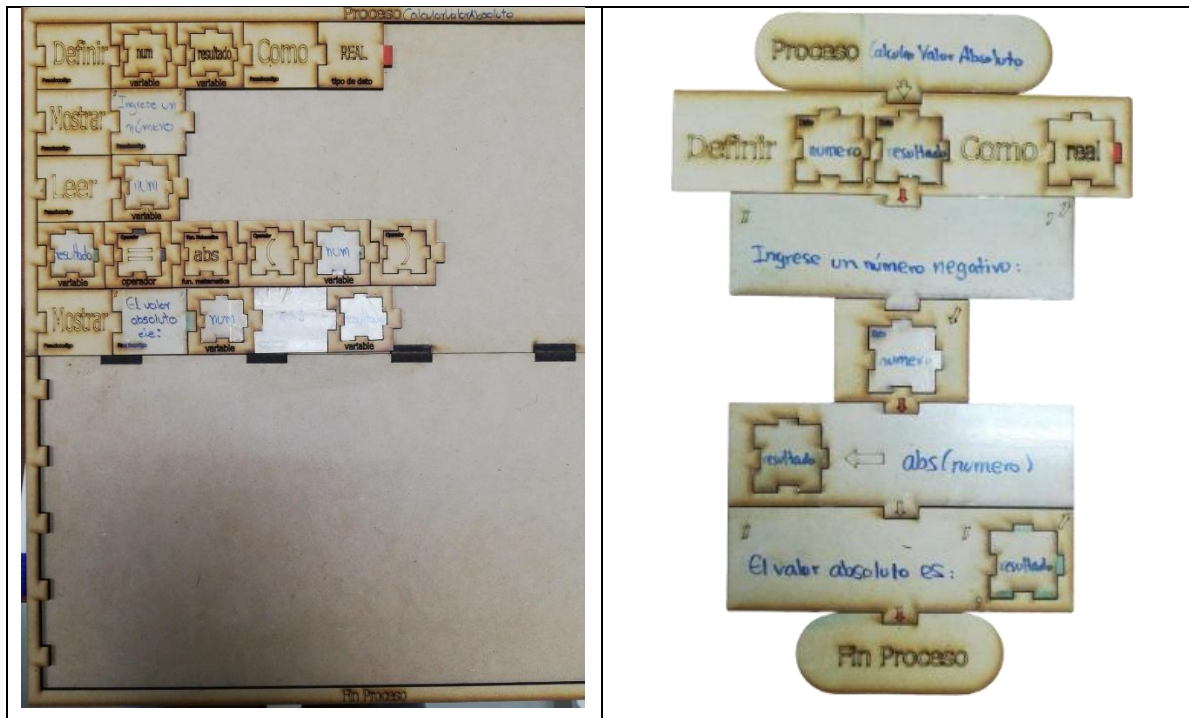
Crea un programa que pida al usuario un número entero y muestre su valor absoluto usando la función 'abs'.

Pasos a Seguir

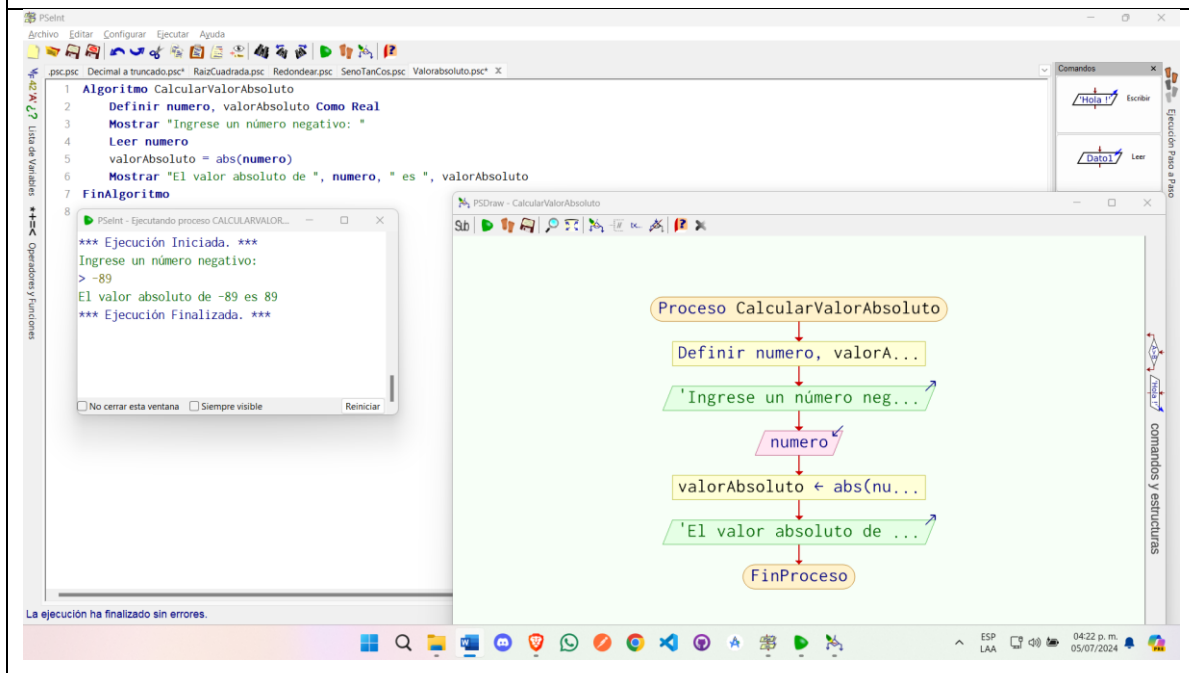
1. Dar un nombre a este proceso: 'CalcularValorAbsoluto'
2. Defina dos tipos de variables de tipo 'Real'
3. Utilice 'Mostrar para mostrar texto al usuario: 'Ingrese un número negativo: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso de un valor al usuario.
5. Asigne a la variable 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



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 **'trunc'**.
6. Utilice 'Mostrar' para mostrar un mensaje con el número ingresado y el resultado de la función matemática.
7. Finalice el Proceso

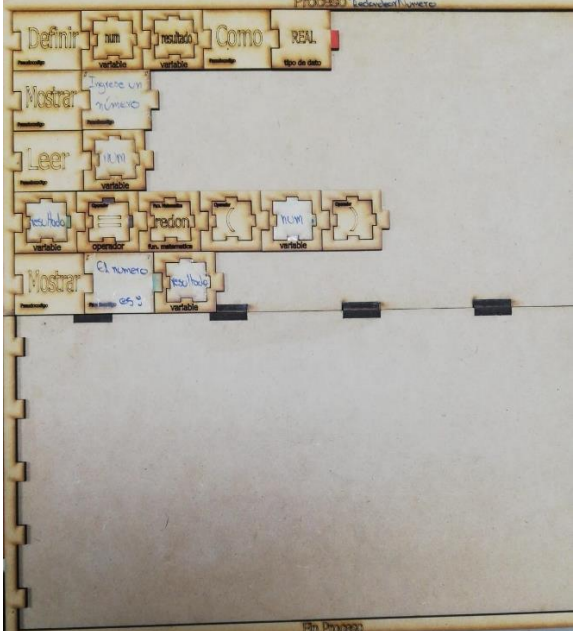

Pseudocódigo	Diagrama de Flujo
Comprueba la solución en PSeInt	

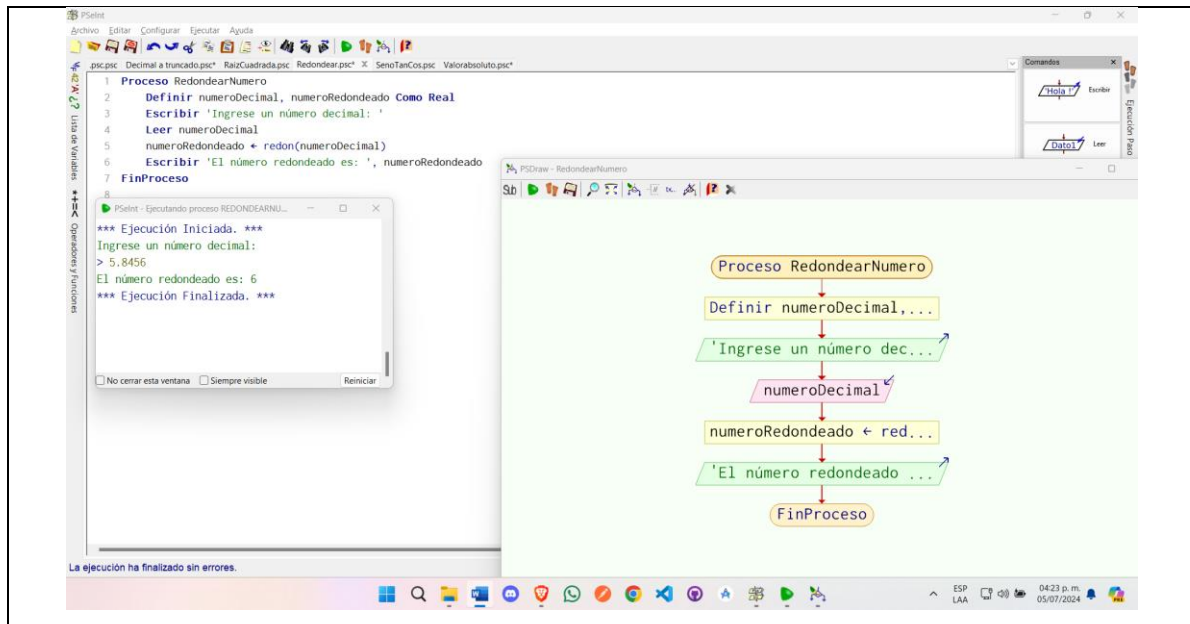
3.1.4 Redondear un número

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: 'RedonderarNumero'
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
	
<p>Comprueba la solución en PSeInt</p>	



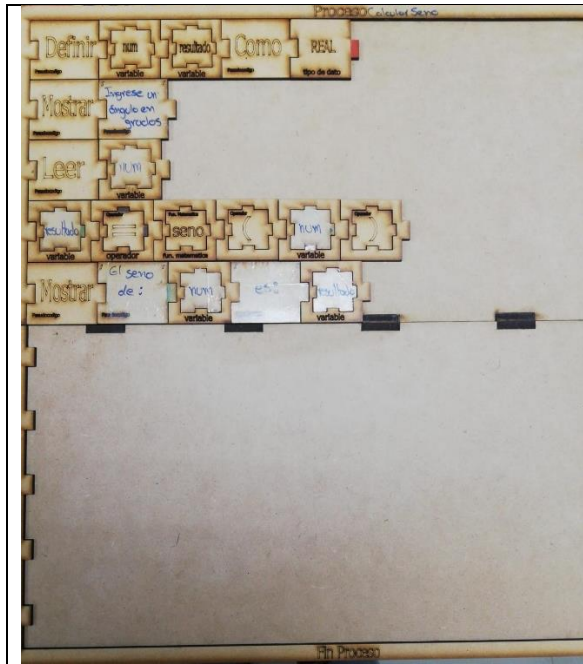
3.1.5 Cálculo del seno de un ángulo en grados

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

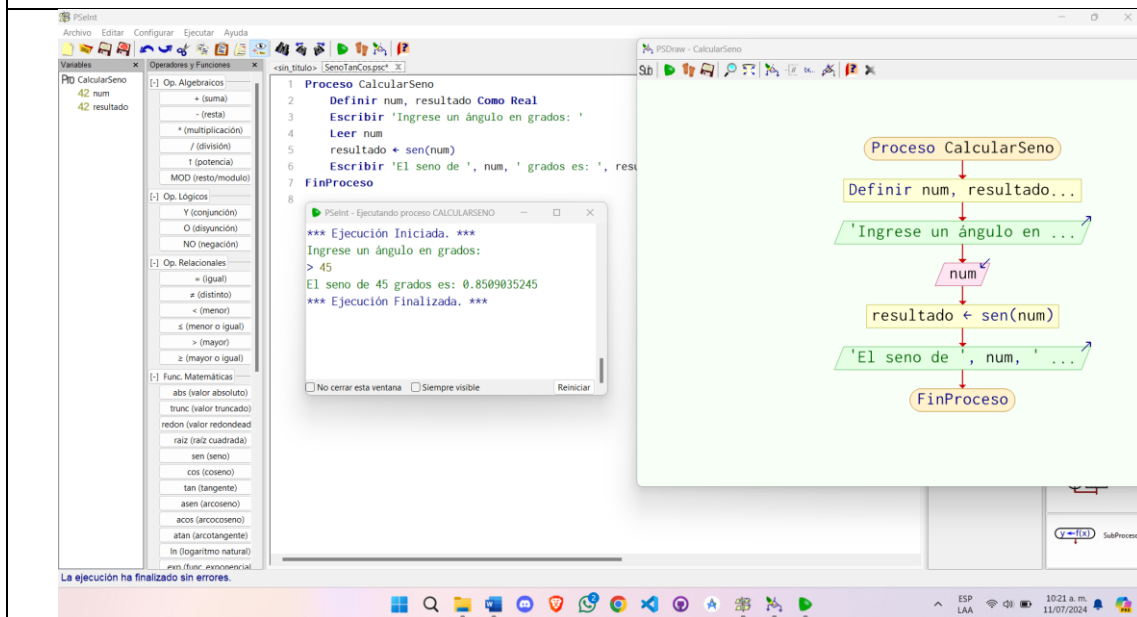
Pasos a Seguir

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 $\text{sen}(\text{variable})$.
6. Utilice 'Mostrar' para mostrar un mensaje con el resultado de la función seno.
7. Finalice el Proceso

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



Comprueba la solución en PSeInt



3.2 Operadores Lógicos

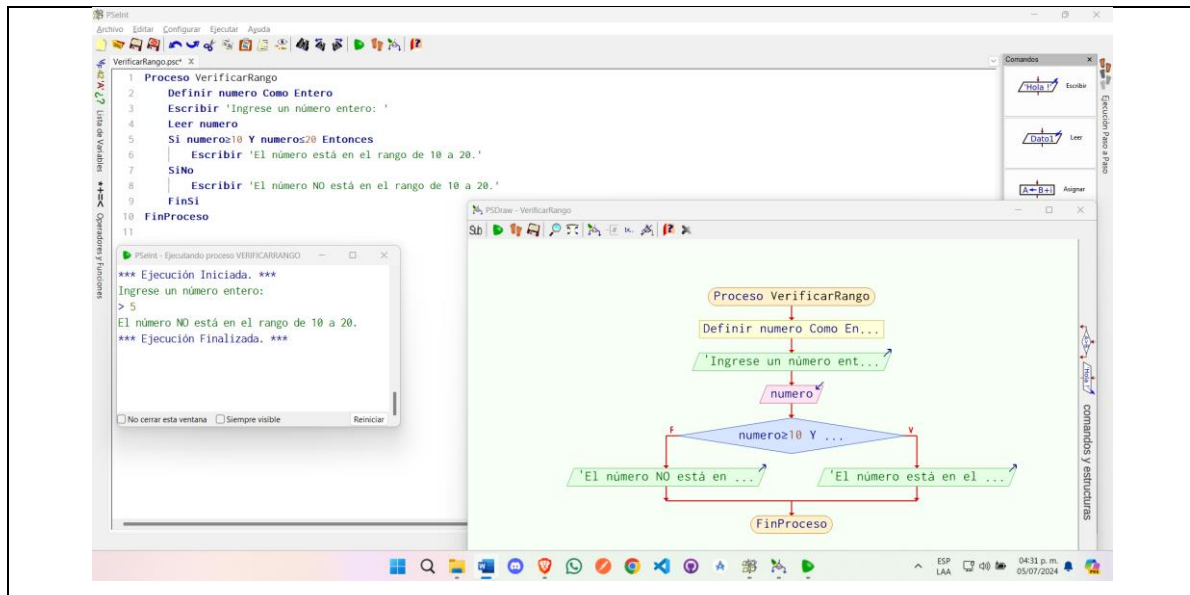
3.2.1 Verificar si un número está en un rango

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
<p>Compreba la solución en PSeInt</p>	



3.2.2 Validar si al menos una condición es verdadera

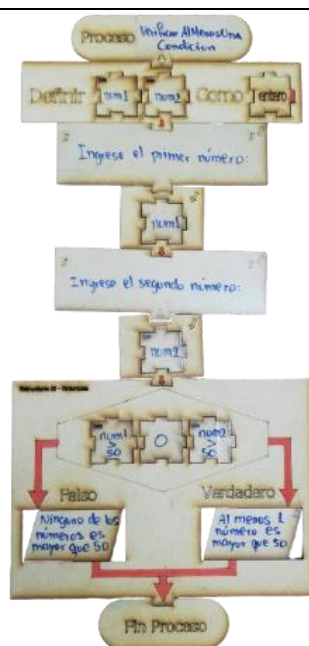
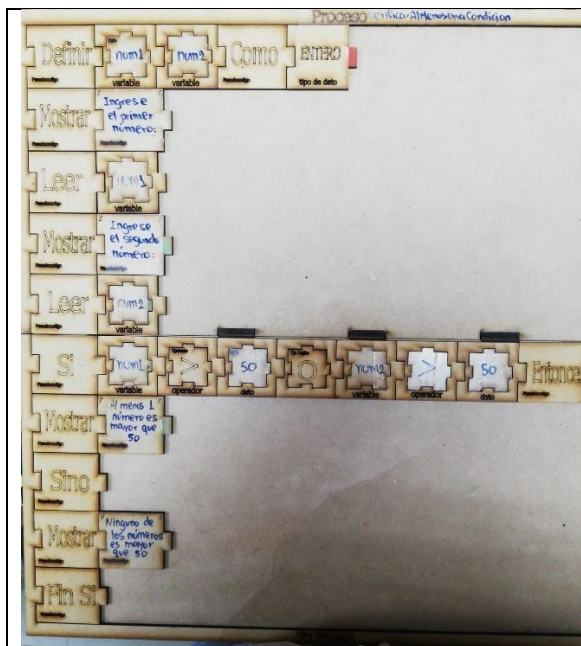
Escribe un programa que lea tres números y verifique si al menos uno de ellos es mayor que 50.

Pasos a Seguir

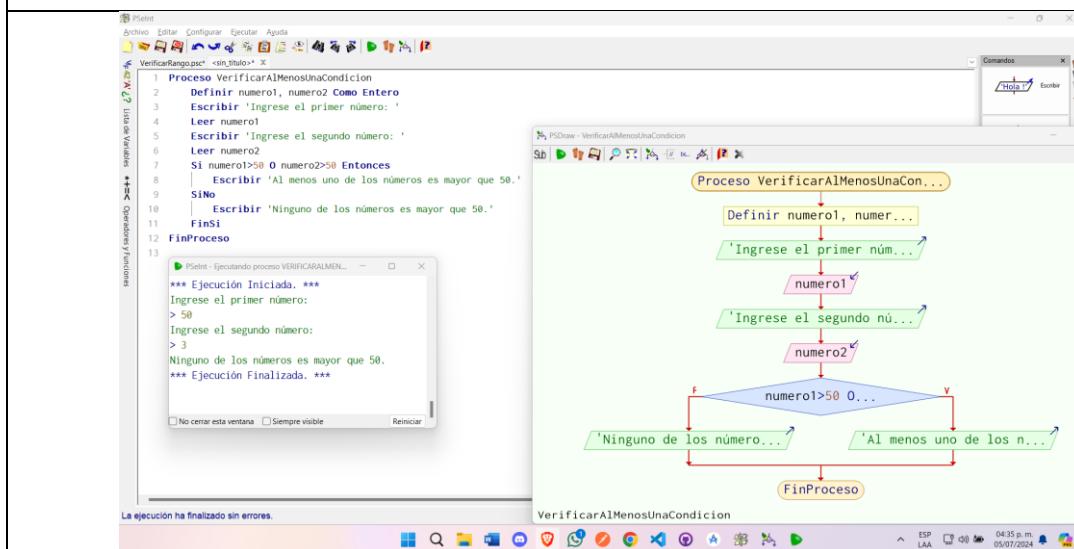
1. Dar un nombre a este proceso: 'VerificarCondicion'
2. Defina dos variables de tipo 'Entero'
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el primer número: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso de un valor al usuario.
5. Utilice 'Mostrar' para mostrar 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



Comprueba la solución en PSeInt



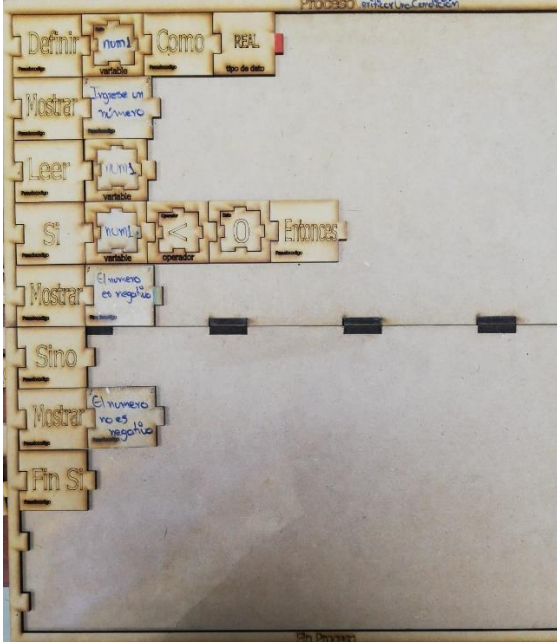
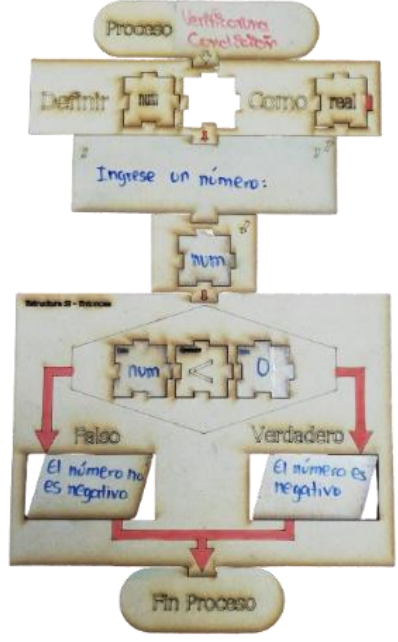
3.2.3 Verificar una condición

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

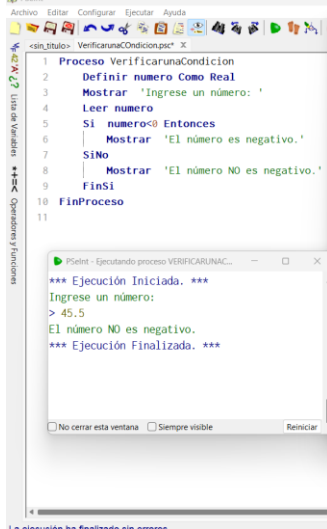
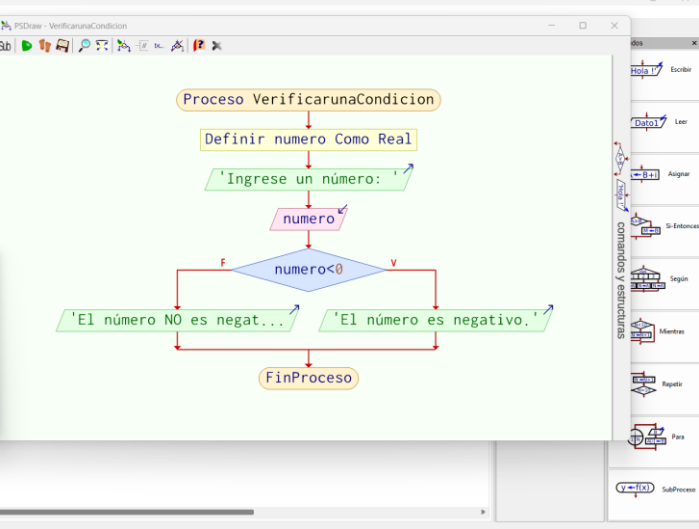
Pasos a Seguir

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 VerificarUnaCondición Definir num Como REAL Mostrar 'Ingrese un número:' Leer num Si num < 0 Entonces Mostrar 'El número es negativo' Sino Mostrar 'El número no es negativo' Fin Si Fin Proceso </pre>	 <pre> graph TD Start([Proceso VerificarUnaCondición]) --> Define[Definir num Como REAL] Define --> Input[Ingrese un número:] Input --> Read[num] Read --> Decision{num < 0} Decision -- Falso --> Output1[El número no es negativo] Decision -- Verdadero --> Output2[El número es negativo] Output1 --> End([Fin Proceso]) Output2 --> End </pre>

Comprueba la solución en PSeInt

 <pre> 1 Proceso VerificarUnaCondicion 2 Definir numero Como Real 3 Mostrar 'Ingrese un número: ' 4 Leer numero 5 Si numero < 0 Entonces 6 Mostrar 'El número es negativo.' 7 SiNo 8 Mostrar 'El número NO es negativo.' 9 FinSi 10 FinProceso 11 </pre>	 <pre> graph TD Start([Proceso VerificarUnaCondicion]) --> Define[Definir numero Como Real] Define --> Input[Ingrese un número:] Input --> Read[numero] Read --> Decision{numero < 0} Decision -- F --> Output1[El número NO es negat...] Decision -- V --> Output2[El número es negativo.] Output1 --> End([FinProceso]) Output2 --> End </pre>
--	---

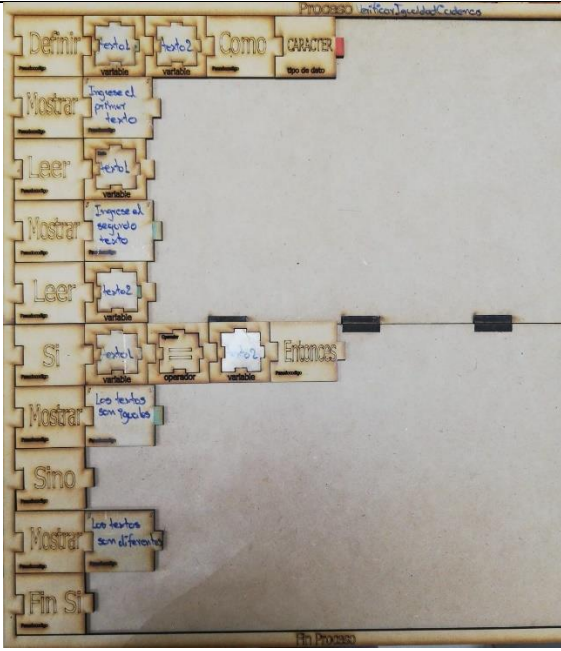
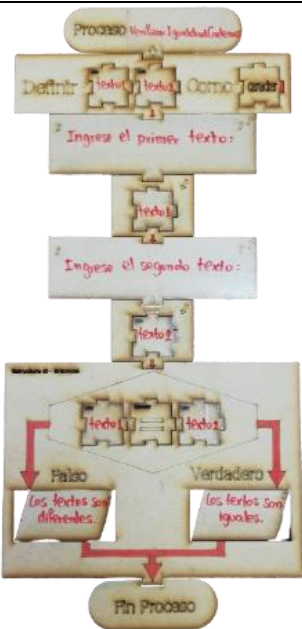
3.3 Operadores Relacionales

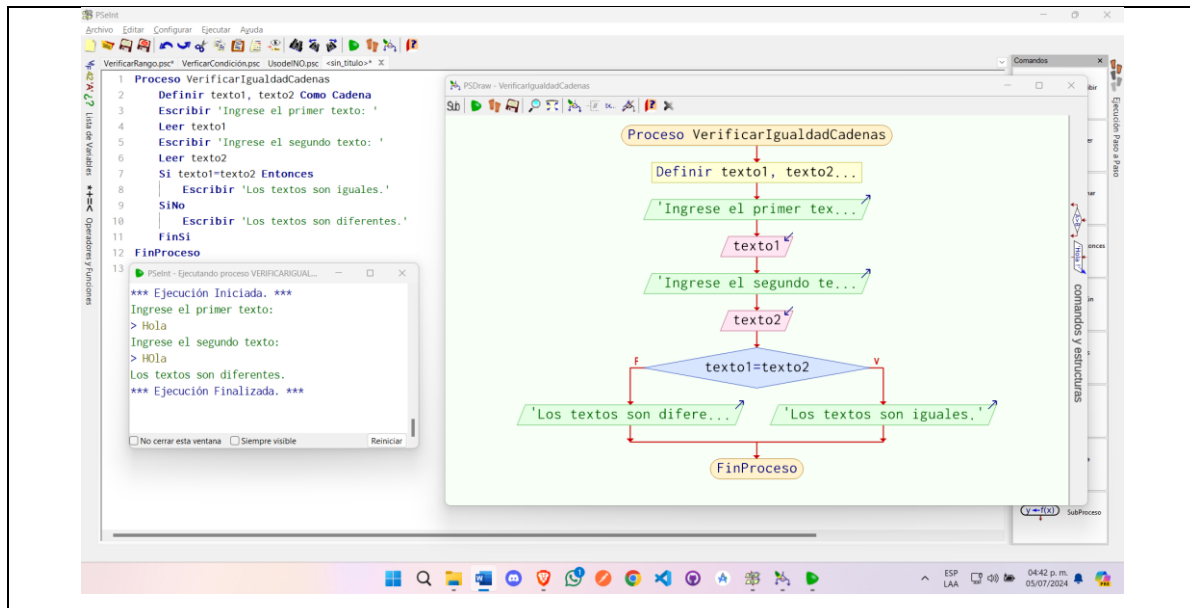
3.3.1 Verificar igualdad de datos 'CARACTER'

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 VerificarIgualdadCaracter Definir texto1, texto2 Como CARACTER Mostrar "Ingrese el primer texto:" Leer texto1 Mostrar "Ingrese el segundo texto:" Leer texto2 Si texto1 = texto2 Entonces Mostrar "Los textos son iguales" Sino Mostrar "Los textos son diferentes" Fin Si Fin Proceso </pre>	 <pre> graph TD Inicio([Inicio]) --> Definir[Definir texto1, texto2 Como CARACTER] Definir --> Mostrar1[Mostrar "Ingrese el primer texto:"] Mostrar1 --> Leer1[Leer texto1] Leer1 --> Mostrar2[Mostrar "Ingrese el segundo texto:"] Mostrar2 --> Leer2[Leer texto2] Leer2 --> Decision{texto1 = texto2} Decision -- Falso --> MostrarDif[Mostrar "Los textos son diferentes"] Decision -- Verdadero --> MostrarIgual[Mostrar "Los textos son iguales"] MostrarDif --> Fin([Fin Proceso]) MostrarIgual --> Fin </pre>
<p align="center">Comprueba la solución en PSeInt</p>	



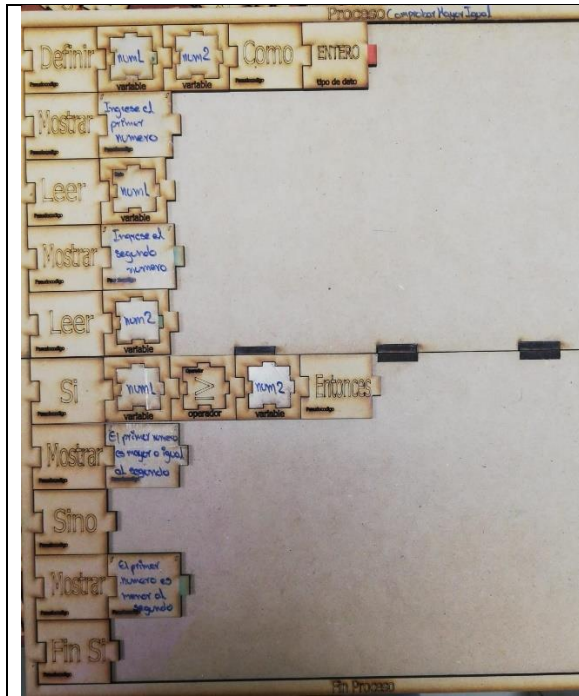
3.3.2 Comprobar si un número es mayor o igual a otro

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
--------------	-------------------



Comprueba la solución en PSeInt

```

1 Proceso ComprobarMayorIgual
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 >= 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 FinSi
12 FinProceso
  
```

*** Ejecución Iniciada. ***
 Ingrese el primer número:
 > 15
 Ingrese el segundo número:
 > 5
 El primer número es mayor o igual al segundo.
 *** Ejecución Finalizada. ***

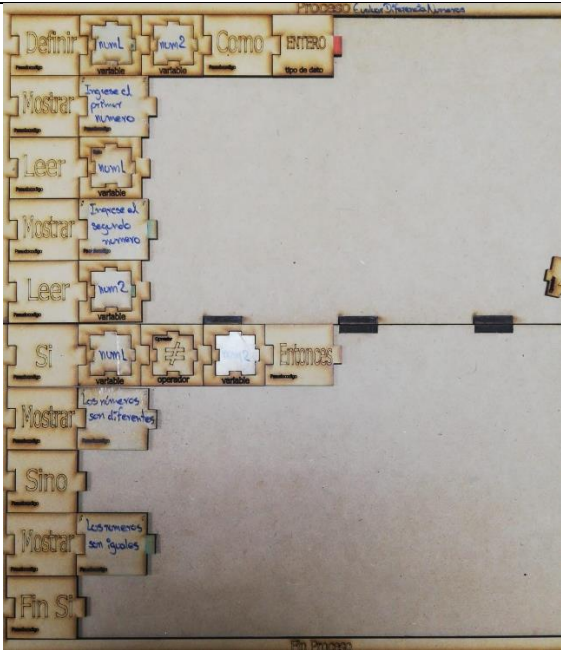
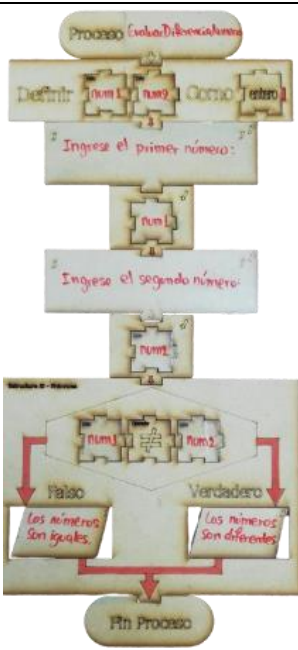
3.3.3 Evaluar la diferencia entre dos números

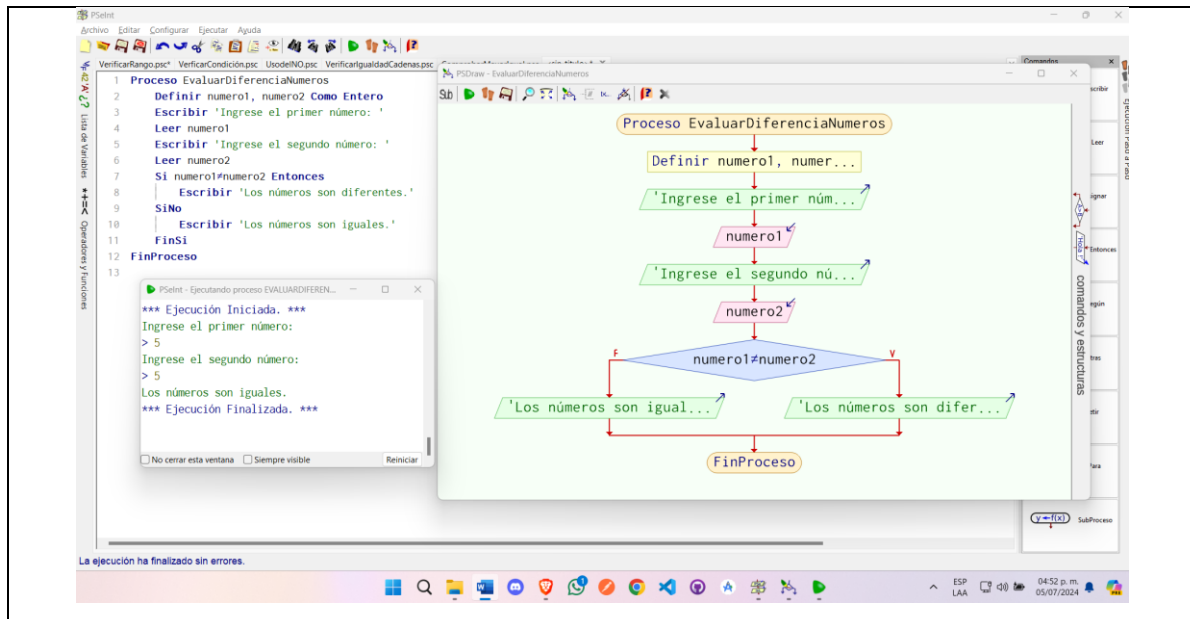
Escribe un programa que lea dos números y verifique si son diferentes.

Pasos a Seguir

1. Dar un nombre a este proceso: 'EvaluarDiferencia'.
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 'Los números son diferentes'.
 - c. Sino Mostrar 'Los numero son iguales'.
 - d. Fin Si
8. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
 <pre> Proceso CompararNumeros Definir num1 Como Entero Definir 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 números son iguales" Fin Si Fin Proceso </pre>	 <pre> graph TD Inicio([Inicio]) --> Definir[Definir num1, num2 Como Entero] Definir --> Ingrese1[Ingrese el primer número:] Ingrese1 --> num1[num1] num1 --> Ingrese2[Ingrese el segundo número:] Ingrese2 --> num2[num2] num2 --> Condicion{num1 <> num2} Condicion -- Falso --> MuestraIgual[Mostrar "Los números son iguales."] Condicion -- Verdadero --> MuestraDif[Mostrar "Los números son diferentes."] MuestraIgual --> Fin([Fin Proceso]) MuestraDif --> Fin </pre>
<p align="center">Comprueba la solución en PSeInt</p>	

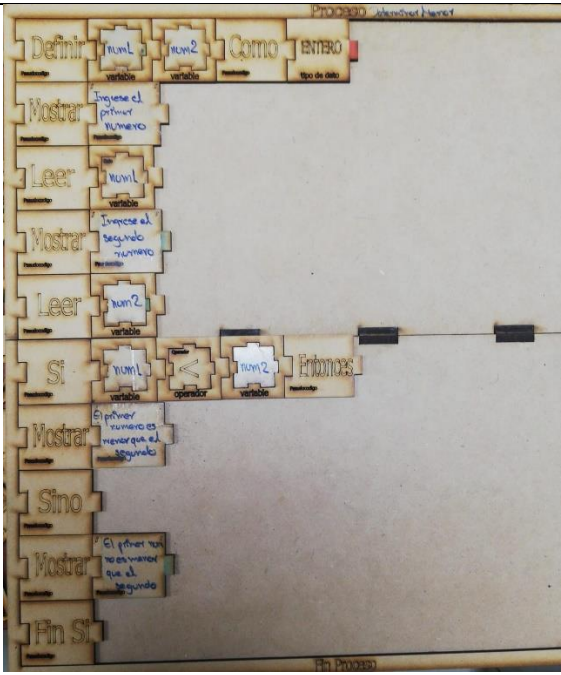
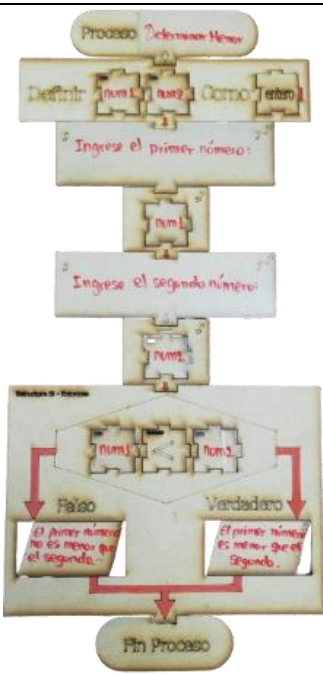
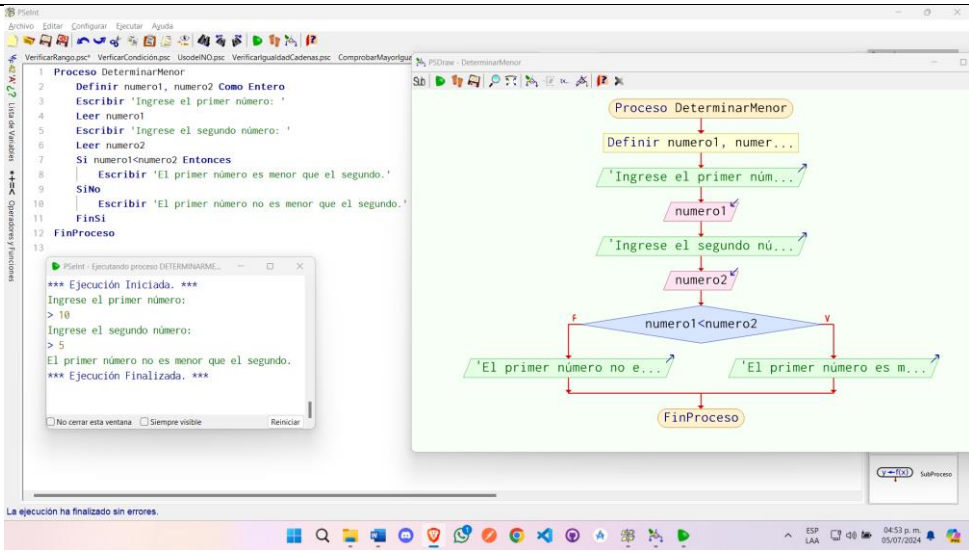


3.3.4 Determinar si un número es menor que otro

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 Definir num1, num2 Como Entero Mostrar 'Ingrese el primer número:' Leer num1 Mostrar 'Ingrese el segundo número:' Leer num2 Si num1 < num2 Entonces Mostrar 'El primer número es menor que el segundo.' Sino Mostrar 'El primer número no es menor que el segundo.' Fin Si Fin Proceso </pre>	 <pre> graph TD Inicio([Inicio]) --> Definir[Definir num1, num2 Como Entero] Definir --> Ingrese1[Ingrese el primer número:] Ingrese1 --> num1[num1] num1 --> Ingrese2[Ingrese el segundo número:] Ingrese2 --> num2[num2] num2 --> Decision{num1 < num2} Decision -- Falso --> Pasa[El primer número no es menor que el segundo.] Decision -- Verdadero --> Pasa Pasa --> FinProceso([Fin Proceso]) </pre>
Comprueba la solución en PSeInt	
 <pre> Proceso DeterminarMenor Definir numero1, numero2 Como Entero Escribir 'Ingrese el primer número:' Leer numero1 Escribir 'Ingrese el segundo número:' Leer numero2 Si numero1 < numero2 Entonces Escribir 'El primer número es menor que el segundo.' Sino Escribir 'El primer número no es menor que el segundo.' FinSi FinProceso </pre> <p>*** Ejecución Iniciada. *** Ingrese el primer número: > 10 Ingrese el segundo número: > 5 El primer número no es menor que el segundo. *** Ejecución Finalizada. ***</p>	

3.4 Operadores Algebraicos

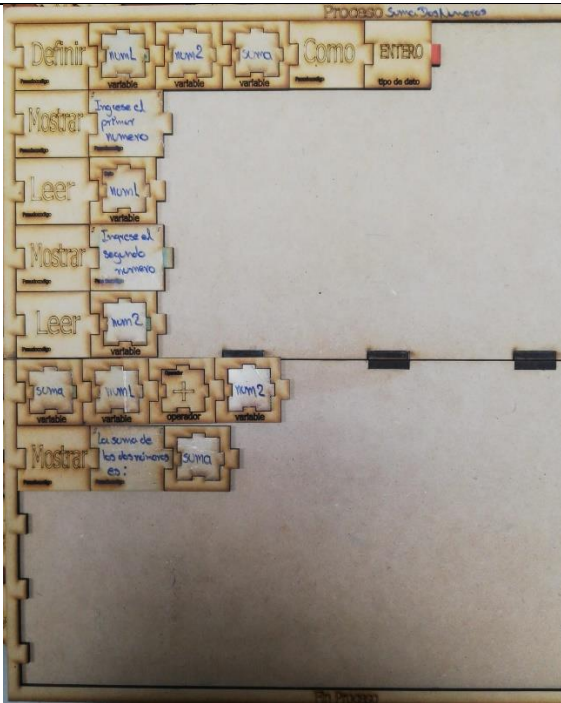

3.4.1 Suma de dos números

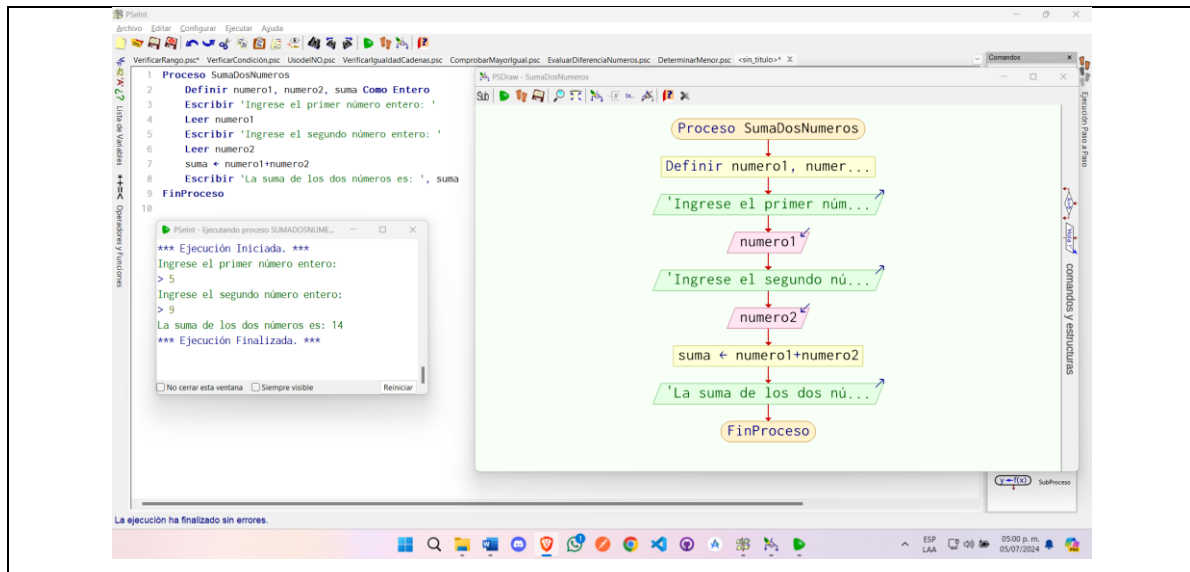
Escribe un programa que lea dos números enteros y muestre su suma.

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 Suma Dos Números Definir num1, num2, suma Como ENTERO Mostrar "Ingrese el primer número:" Leer num1 Mostrar "Ingrese el segundo número:" Leer num2 suma ← num1 + num2 Mostrar "La suma de los dos números es: ", suma </pre>	 <pre> graph TD Start([Proceso Suma Dos Números]) --> Def1[Definir num1, num2 Como ENTERO] Def1 --> Def2[Definir suma Como ENTERO] Def2 --> P1[/Ingrese el primer número:/] P1 --> L1[Leer num1] L1 --> P2[/Ingrese el segundo número:/] P2 --> L2[Leer num2] L2 --> A1[suma ← num1 + num2] A1 --> P3[/La suma de los dos números, suma es:/] P3 --> End([Fin Proceso]) </pre>
<p align="center">Comprueba la solución en PSeInt</p>	



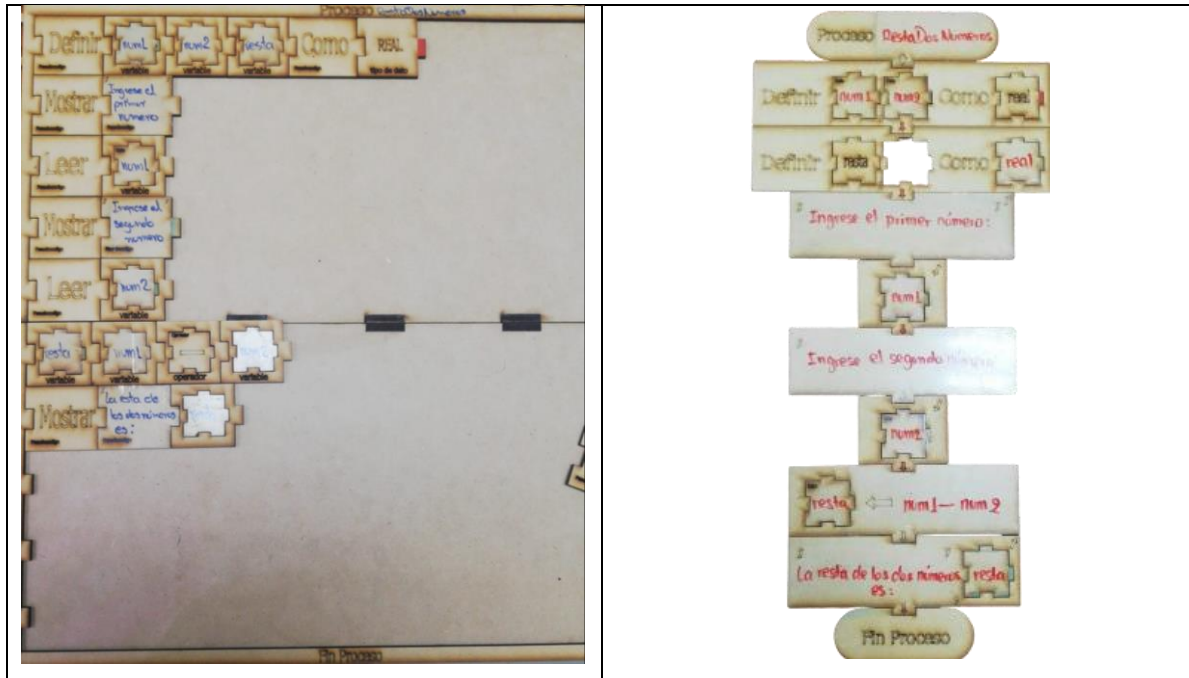
3.4.2 Resta de dos números

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

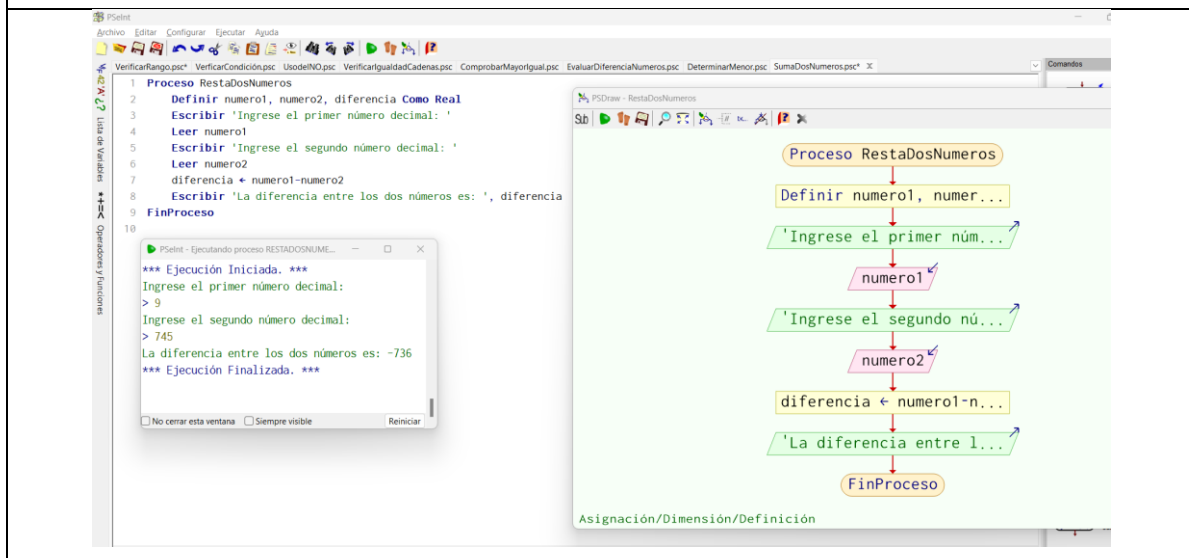
Pasos a Seguir

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
--------------	-------------------



Comprueba la solución en PSeInt



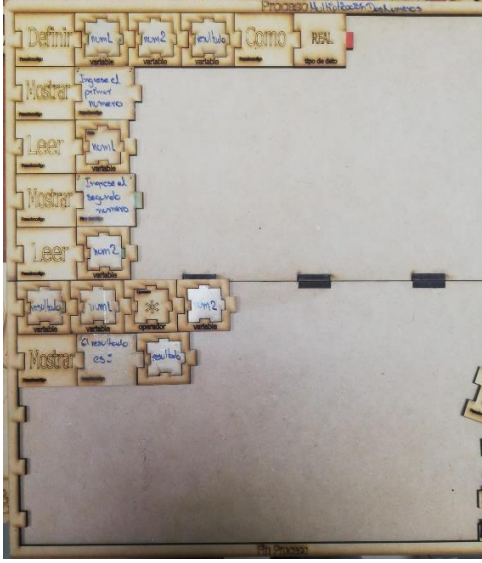

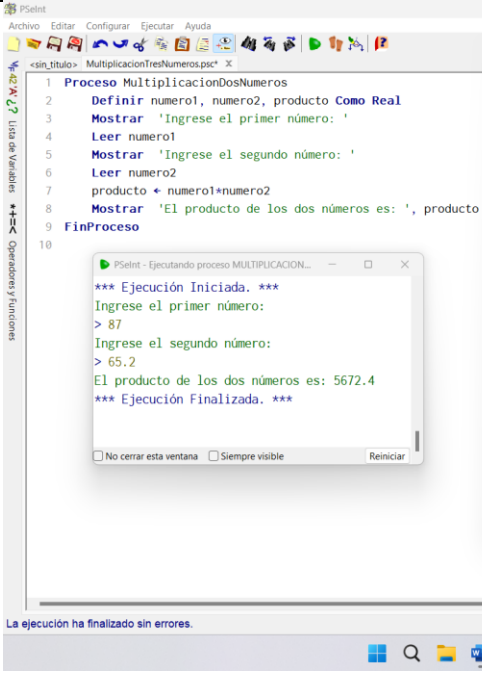
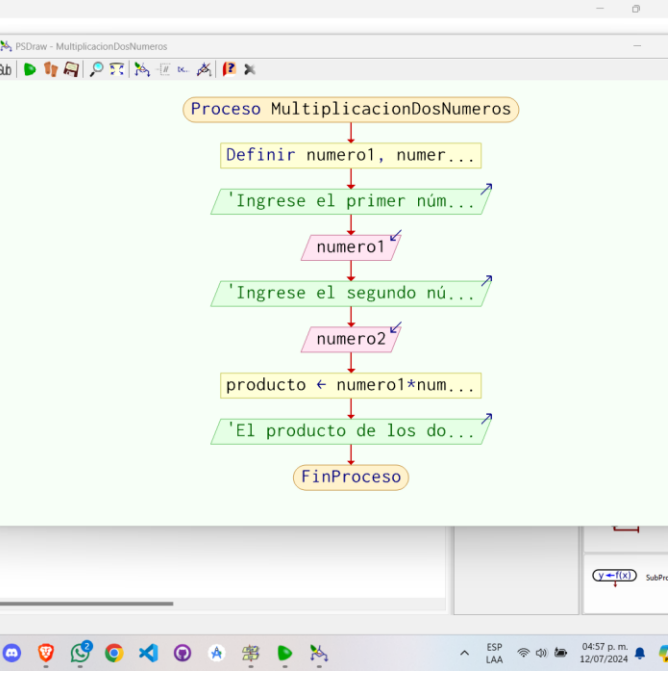
3.4.3 Multiplicación de dos números

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 número: '
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 multiplicación es: ', resultado.
9. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
	
Comprueba la solución en PSeInt	
	

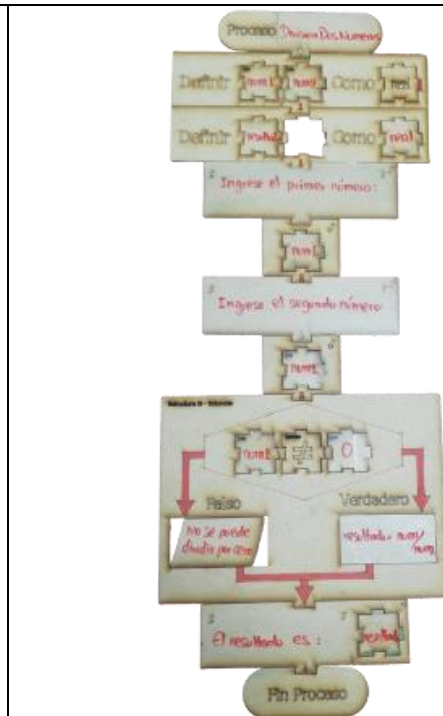
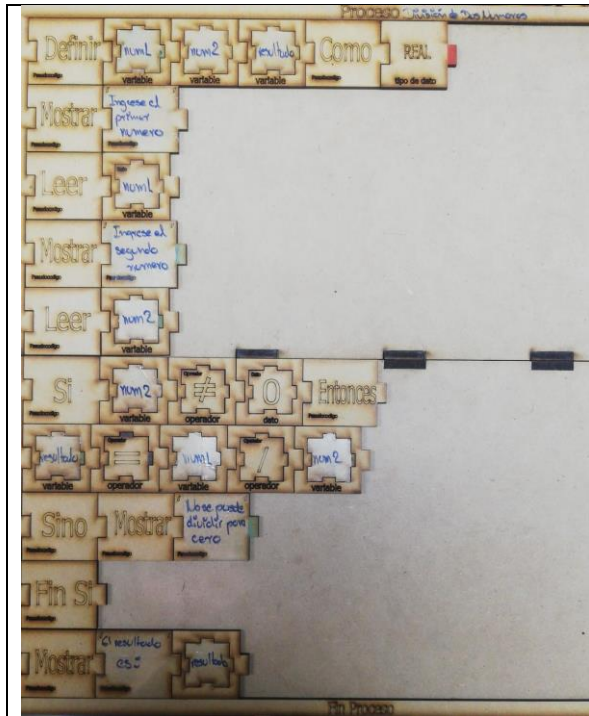
3.4.4 División de dos números

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' \neq 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

```

1 Proceso DivisionDosNumeros
2 Definir numero1, numero2, cociente, residuo Como Real
3 Mostrar 'Ingrese el primer número: '
4 Leer numero1
5 Mostrar 'Ingrese el segundo número: '
6 Leer numero2
7 Si numero2 ≠ 0 Entonces
8     cociente ← numero1/numero2
9
10 SiNo
11     Mostrar 'No se puede dividir por cero.'
12
13 FinSi
14
15 Mostrar 'El cociente de la división es: ', cociente
16 FinProceso
  
```

*** Ejecución Iniciada. ***
 Ingrese el primer número:
 > 45
 Ingrese el segundo número:
 > 2.6
 El cociente de la división es: 17.3076923077
 *** Ejecución Finalizada. ***

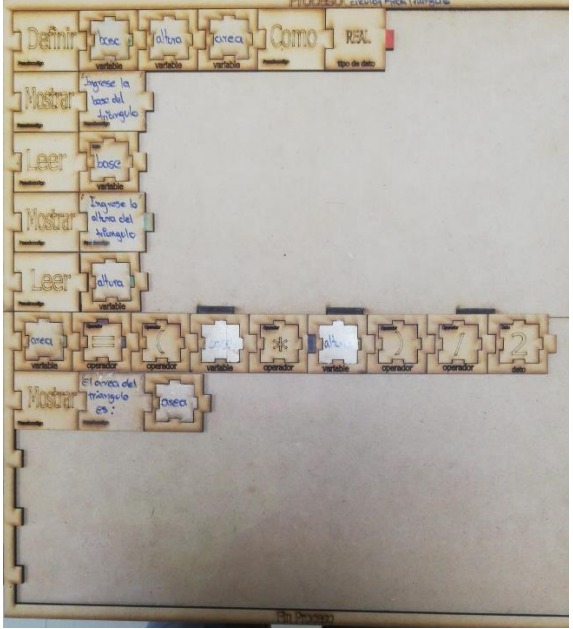

3.4.5 Cálculo del área de un triángulo

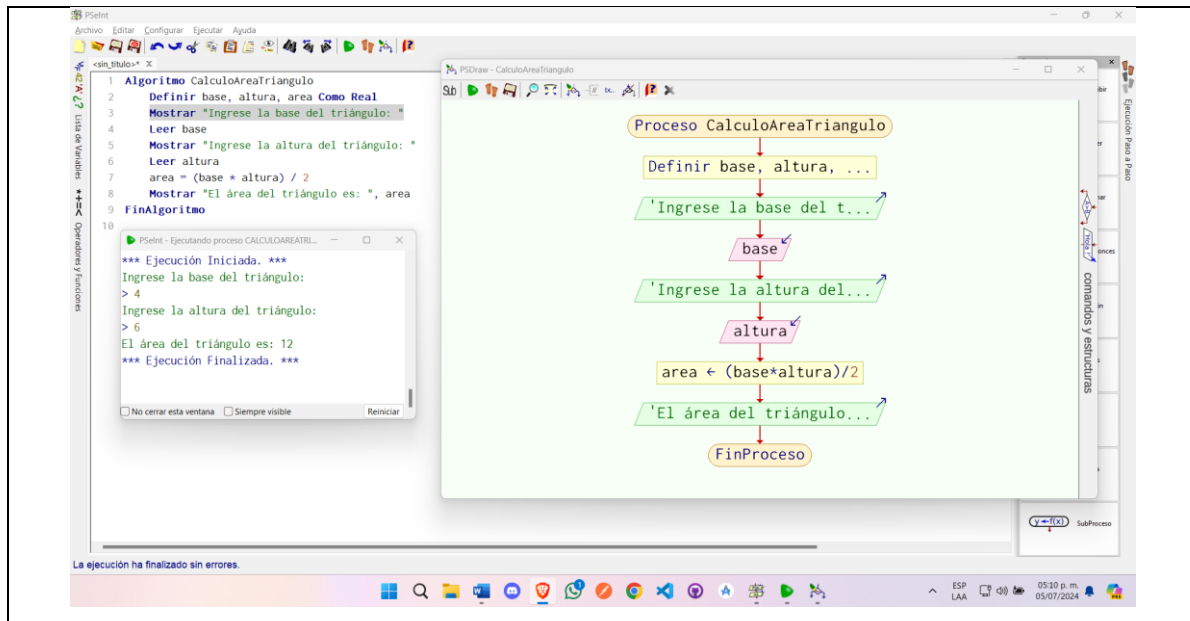
Escribe un programa que lea la base y la altura de un triángulo y calcule su área utilizando la fórmula $\text{Área} = (\text{base} * \text{altura}) / 2$.

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 triángulo: '
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 triángulo: '
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 triángulo es: ', area.
9. Finalice el Proceso

Pseudocódigo	Diagrama de Flujo
 <pre> Proceso Calcular Area Triangulo Definir base, altura, area Como REAL Mostrar "Ingrese la base del triángulo:" Leer base Mostrar "Ingrese la altura del triángulo:" Leer altura area ← (base * altura) / 2 Mostrar "El area del triángulo es: ", area Fin Proceso </pre>	 <pre> graph TD Start([Proceso Calcular Area Triangulo]) --> Def1[Definir base, altura, area Como REAL] Def1 --> Def2[Definir area Como real] Def2 --> Input1[2 Ingrese la base del triángulo:] Input1 --> Var1[base] Var1 --> Input2[4 Ingrese la altura del triángulo:] Input2 --> Var2[altura] Var2 --> Calc[area ← (base * altura) / 2] Calc --> Output[5 El area del triángulo es: area] Output --> End([Fin Proceso]) </pre>
<p>Comprueba la solución en PSeInt</p>	



3.5 Ejercicios con 'SI'

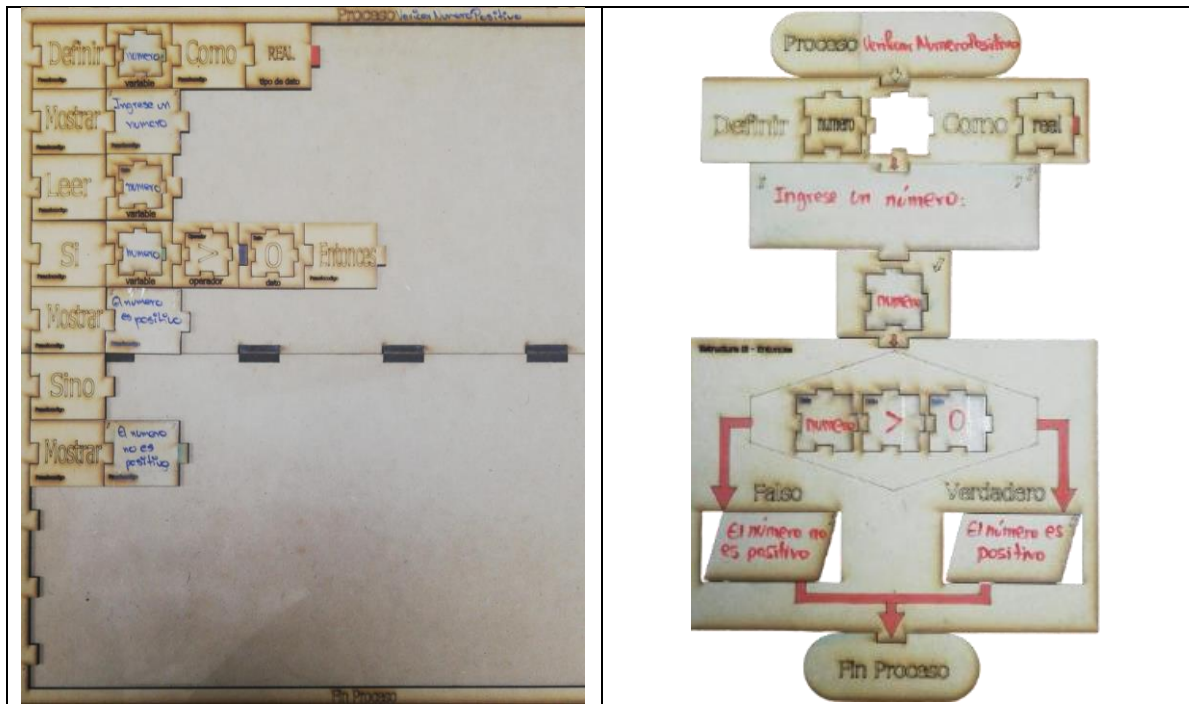
3.5.1 Verificar si un número es positivo

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

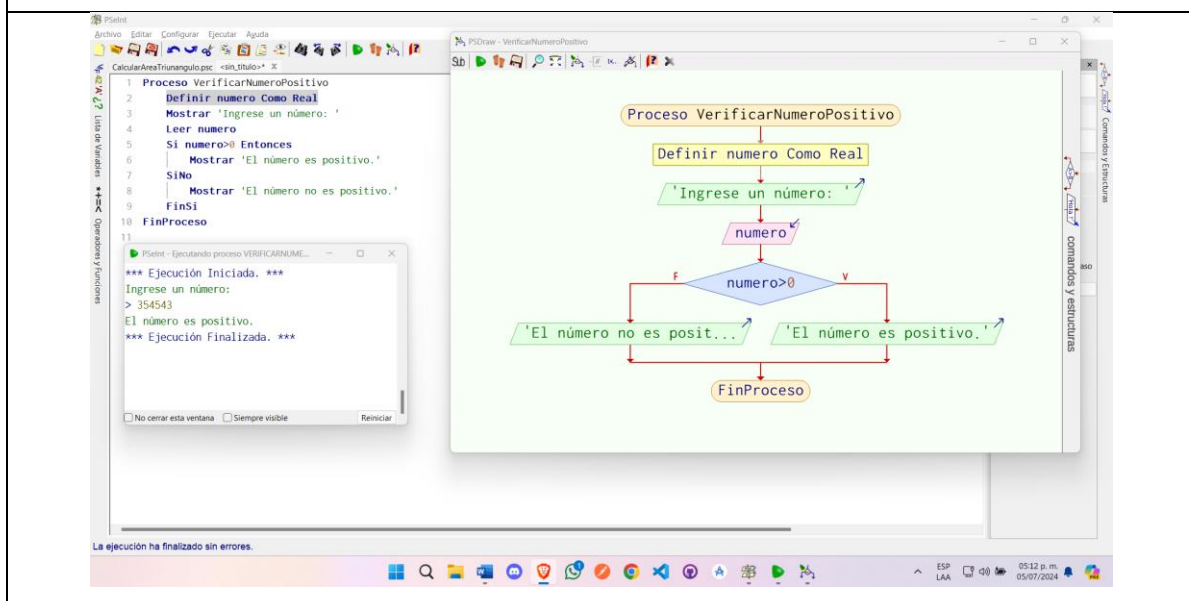
Pasos a Seguir

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



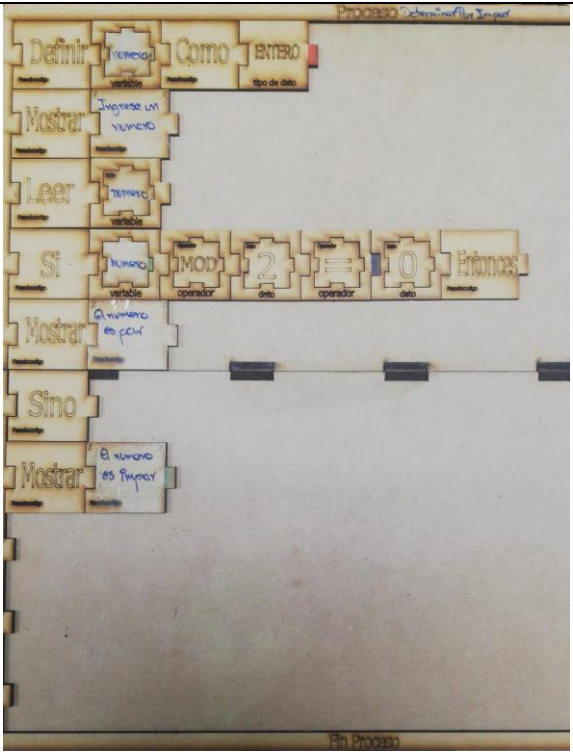
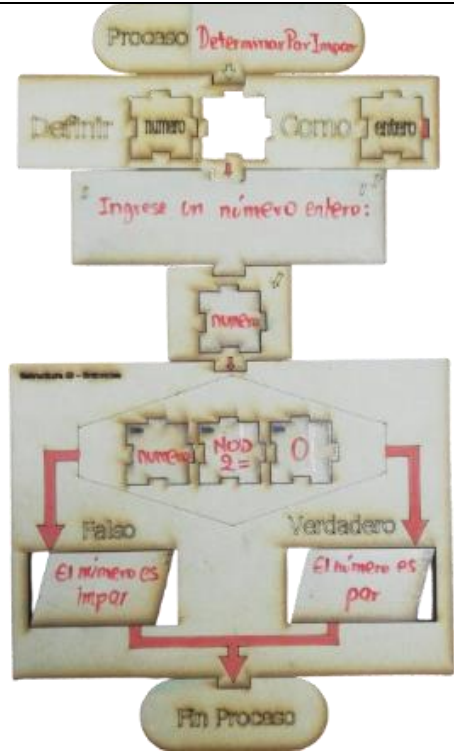
3.5.2 Determinar si un número es par o impar

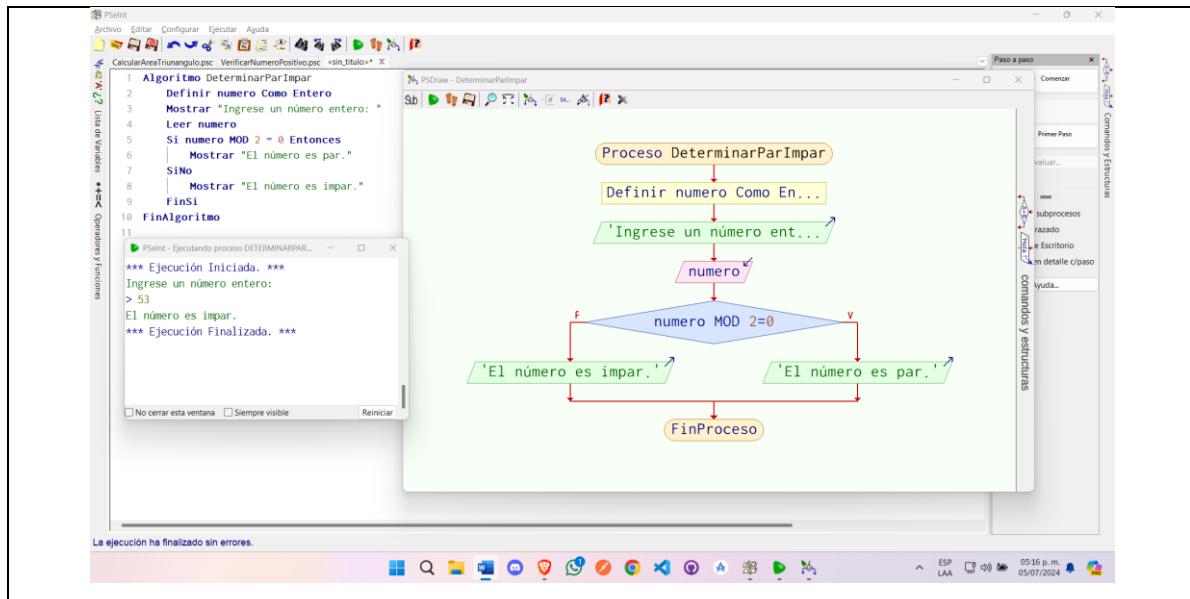
Crea un programa que lea un número entero y determine si es par o impar. Muestra un mensaje acorde a cada caso.

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
	
<p>Comprueba la solución en PSeInt</p>	



3.5.3 Calcular descuento por compra

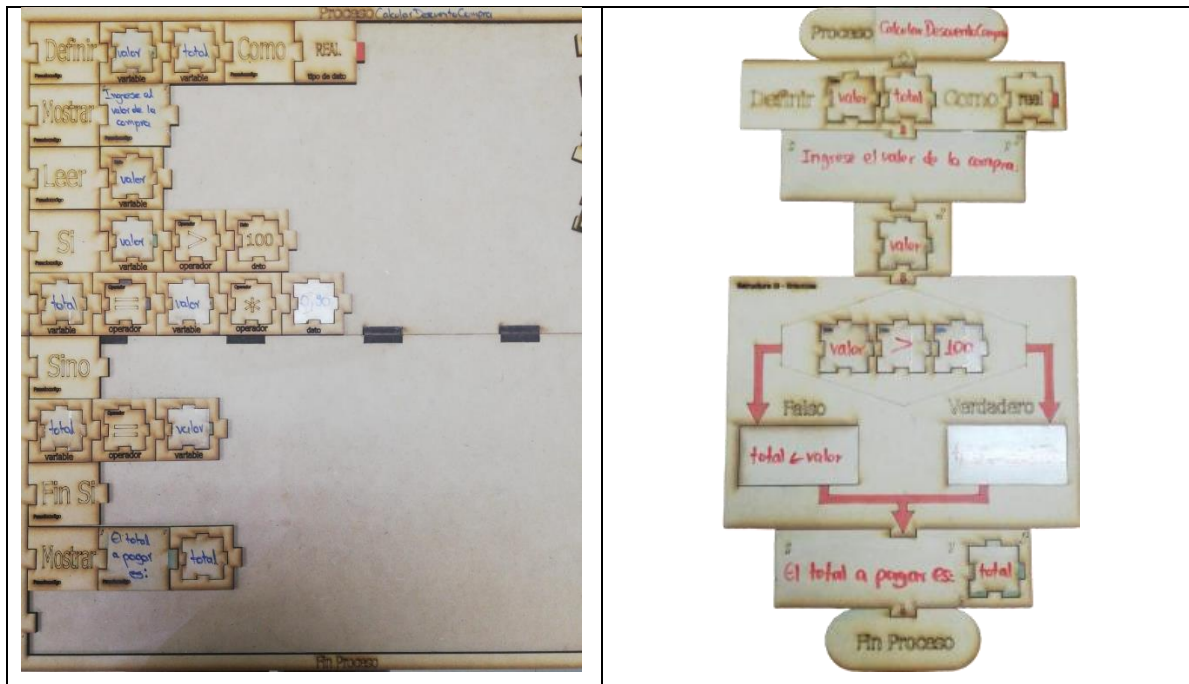
Diseña un programa que lea el valor de una compra y aplique un descuento del 10% si el monto es mayor a 100. Muestra el total a pagar después del descuento.

Pasos a Seguir

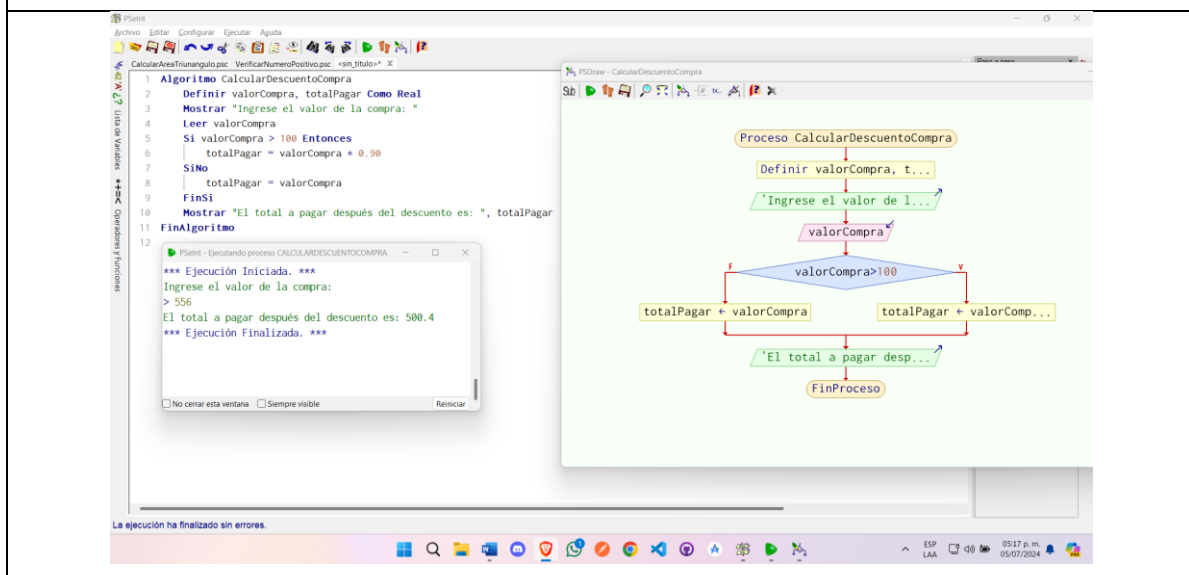
1. Dar un nombre a este proceso: 'DescuentoCompra'.
2. Defina dos variables de tipo 'Real'.
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese el valor total de la compra: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del valor al usuario.
5. Use la estructura de control '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



Comprueba la solución en PSeInt



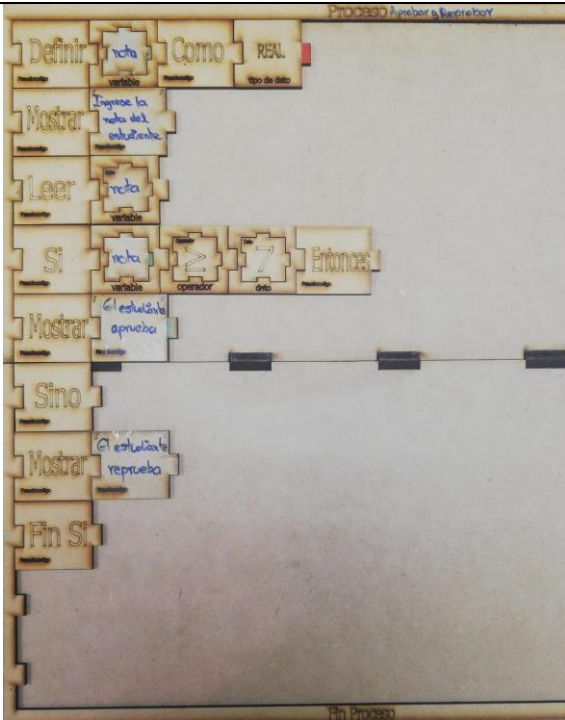
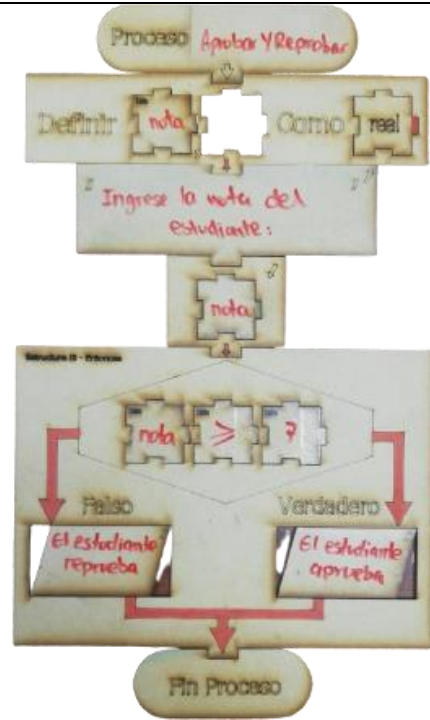
3.5.4 Evaluar si un estudiante aprueba o reprueba

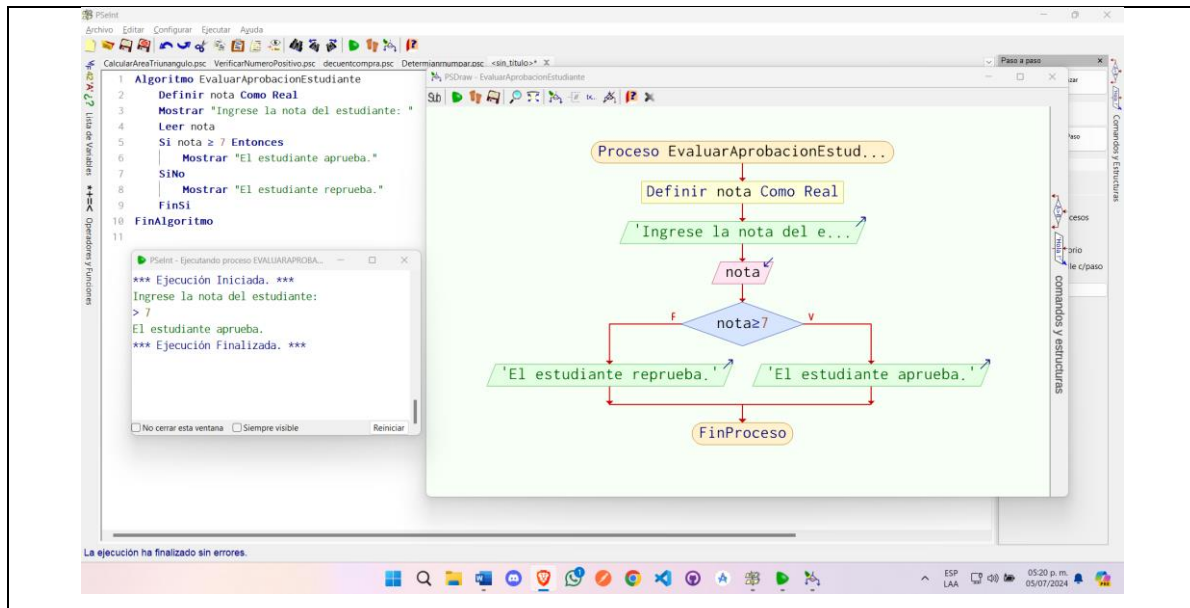
Programa que lea la nota de un estudiante y determine si aprueba (nota mayor o igual a 70) o reprueba. Muestra un mensaje con el resultado.

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' \geq 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
	
Comprueba la solución en PSeInt	



3.5.5 Comprobar si una persona es mayor de edad

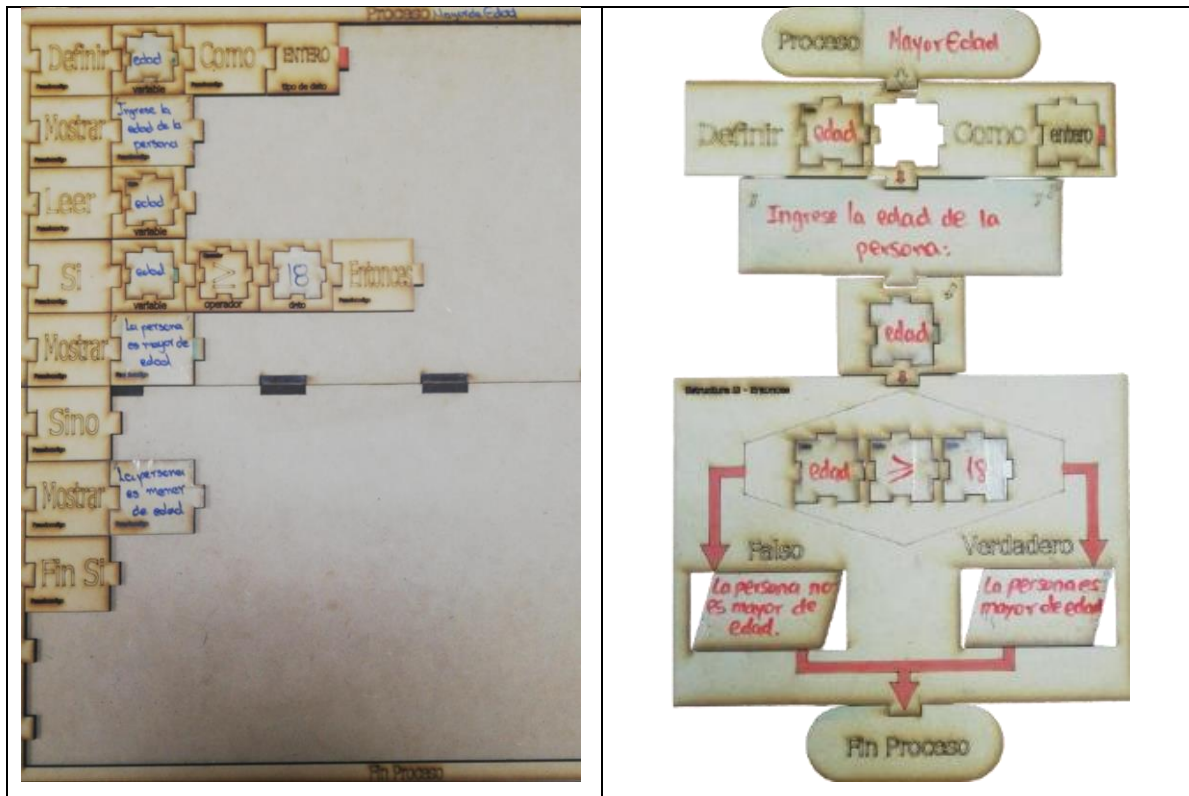
Escribe un programa que lea la edad de una persona y determine si es mayor de edad (18 años o más). Muestra un mensaje correspondiente.

Pasos a Seguir

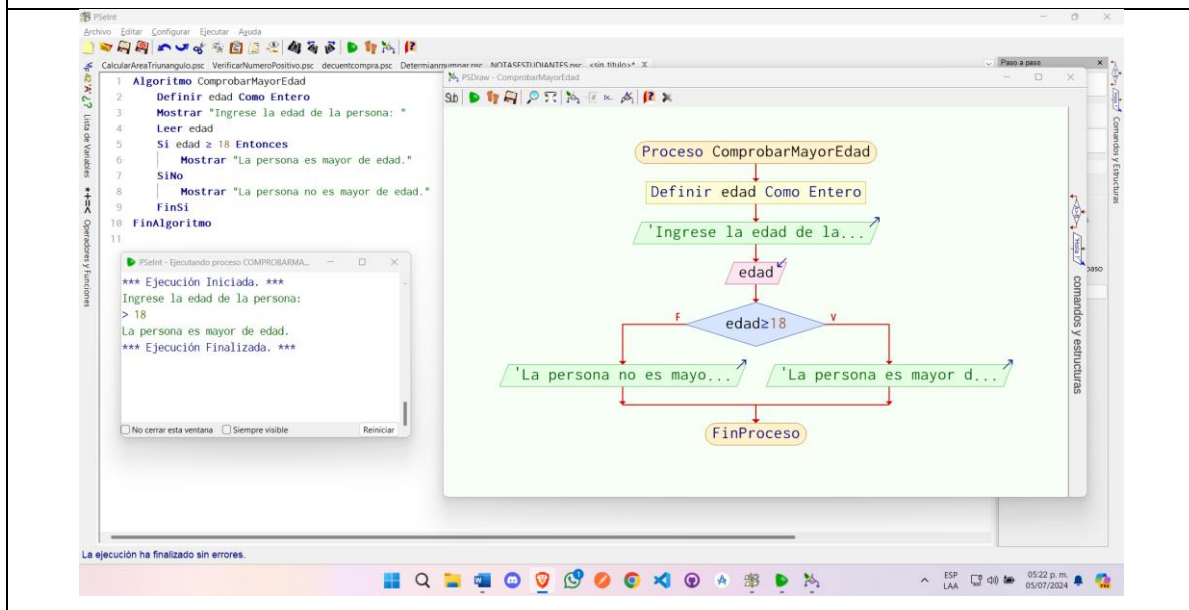
1. Dar un nombre a este proceso: 'EdadPersona'.
2. Defina una variable de tipo 'Entero'.
3. Utilice 'Mostrar' para mostrar texto al usuario: 'Ingrese la edad de la persona: '
4. Utilice 'Leer' y el nombre de la 'variable' para pedir el ingreso del valor al usuario.
5. Use la estructura de control 'SI'
 - 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 PSeInt



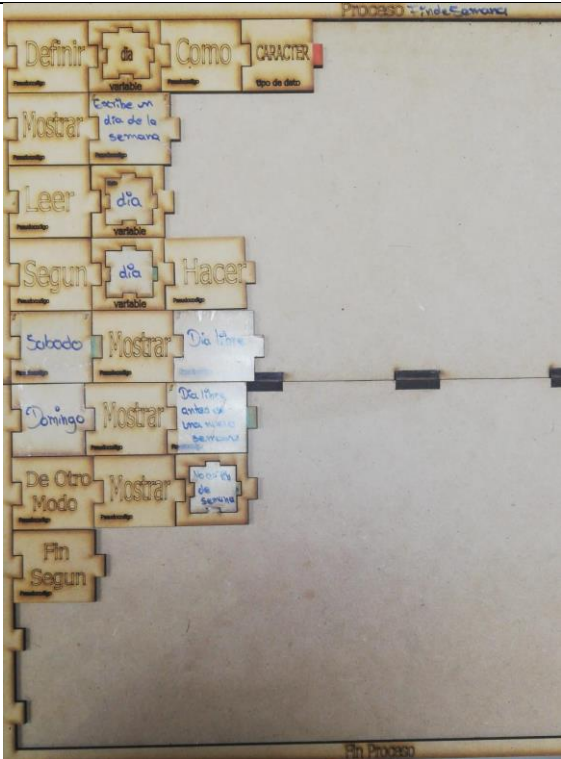
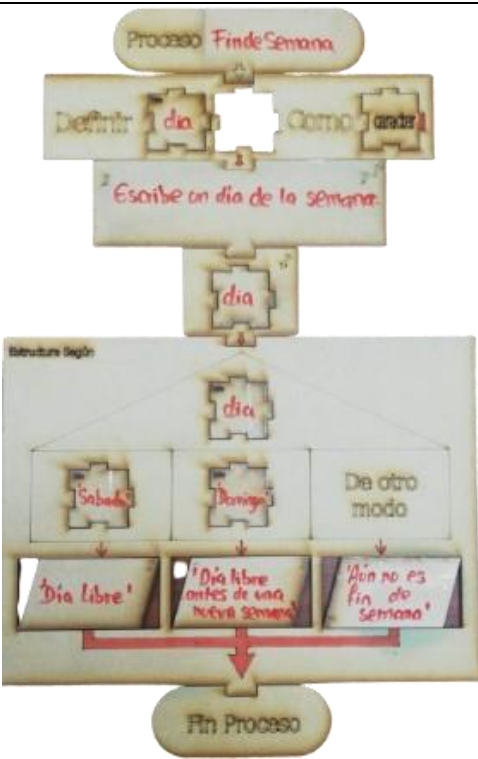
3.6 Ejercicios con 'SEGUN'

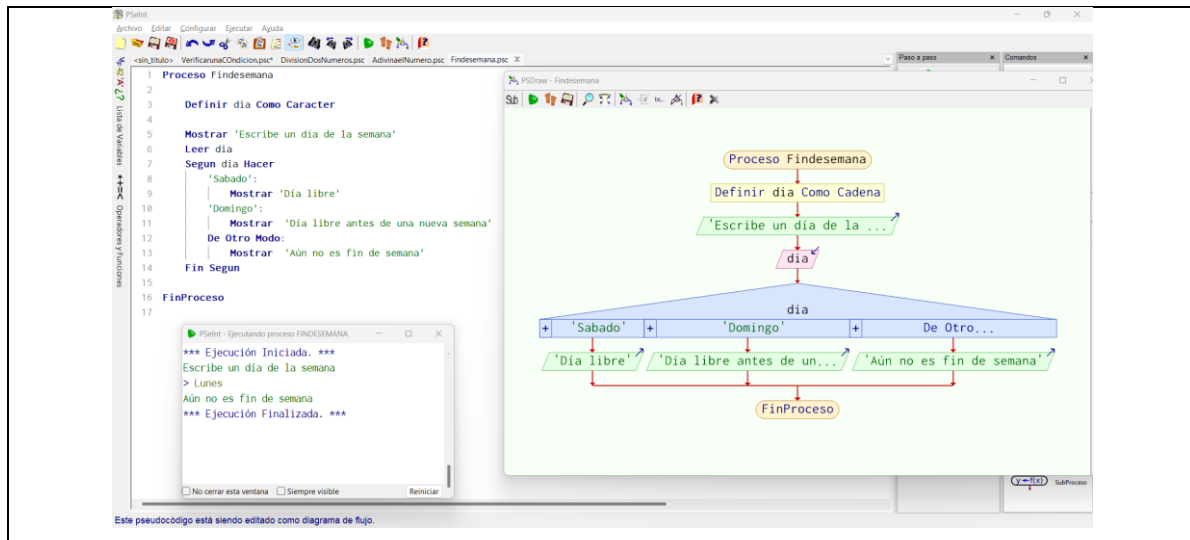
3.6.1 Mostrar día de la semana

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
	
<p>Comprueba la solución en PSeInt</p>	



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: 'Ingresa 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
Comprueba la solución en PSeInt	

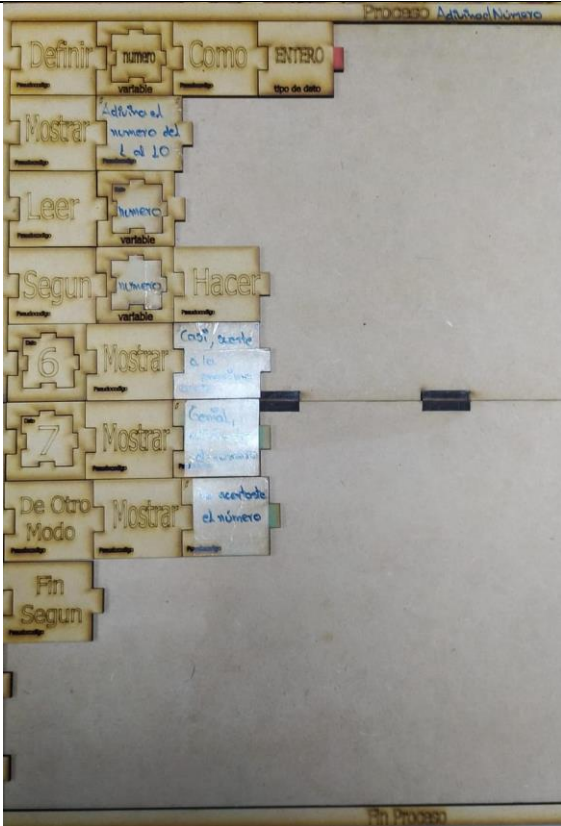
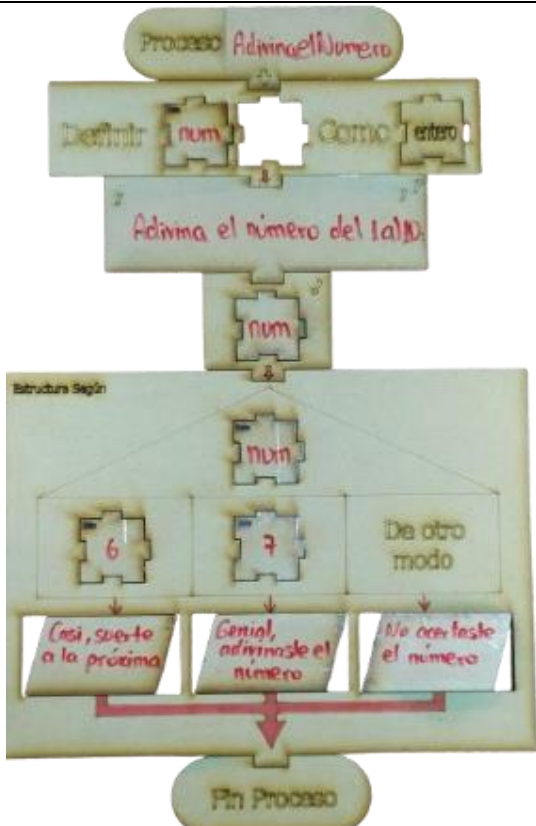
3.6.3 Adivina el número

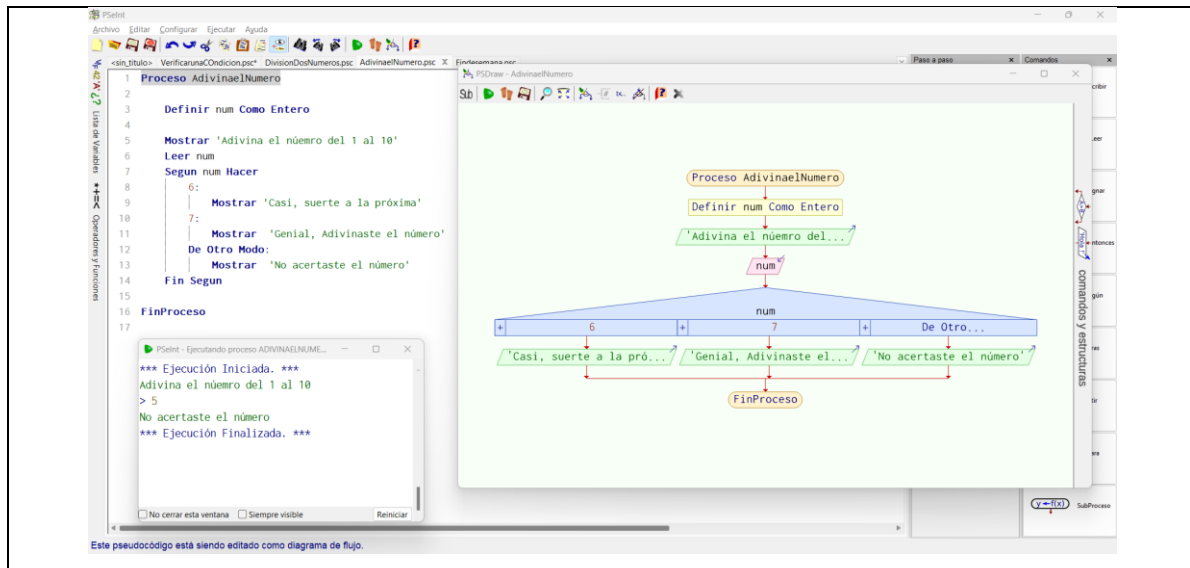
Programa que lea un número del 1 al 10 y muestre los siguientes mensajes 6 = 'Casi, suerte a la próxima', 7 = 'Genial, Adivinaste el número', De otro modo = 'No acertaste el número'.

Pasos a Seguir

8. Dar un nombre a este proceso: 'AdivinaelNumero'.

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
 <p>The pseudocode blocks are arranged as follows:</p> <ul style="list-style-type: none"> Process: Adivina el Numero Define: numero Como ENTERO Mostrar: Adivina el numero del 1 al 10 Leer: numero Segun: numero Hacer <ul style="list-style-type: none"> 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 	 <p>The flowchart steps are:</p> <ol style="list-style-type: none"> Start: Proceso Adivina el Numero Define: num Como entero Mostrar: Adivina el numero del 1 al 10 Leer: num Decision: Segun num <ul style="list-style-type: none"> Caso 6: Mostrar: Casi, suerte a la proxima Caso 7: Mostrar: Genial, adivinaste el numero De otro modo: Mostrar: No acertaste el numero End: Fin Proceso
<p>Comprueba la solución en PSeInt</p>	



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
Comprueba la solución en PSeInt	

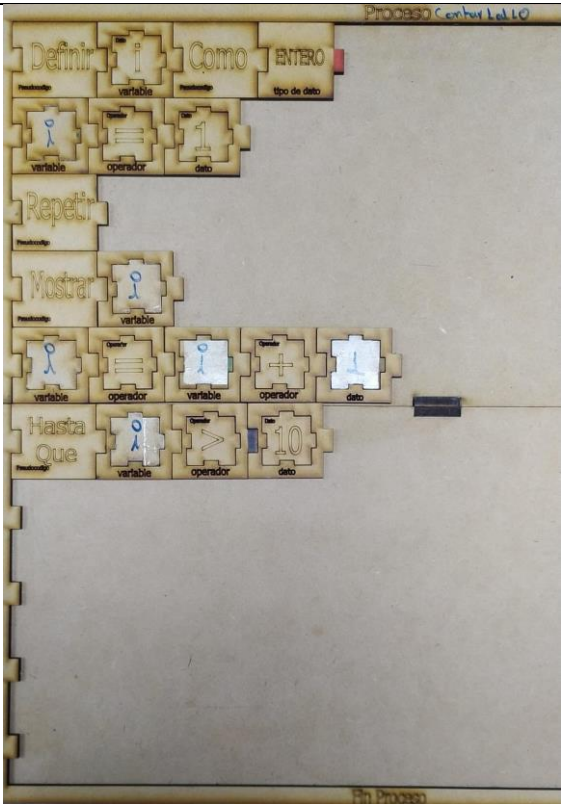
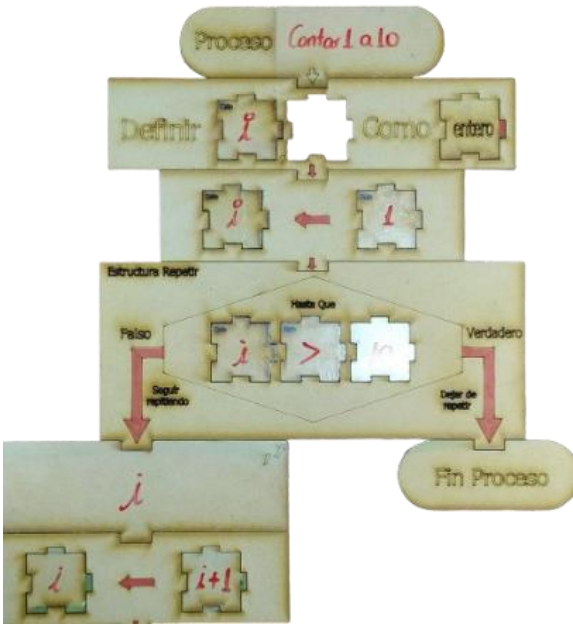
3.7 Ejercicios con 'REPETIR'

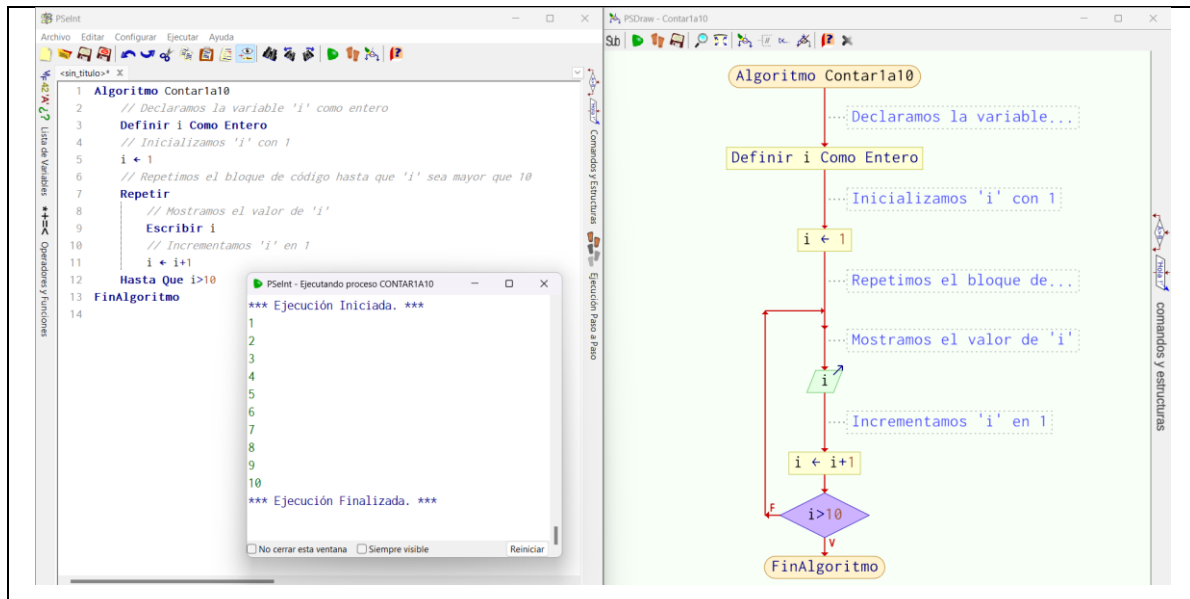
3.7.1 Contar del 1 al 10

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
	
<p>Comprueba la solución en PSeInt</p>	



3.7.2 Sumar del 1 al 5

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

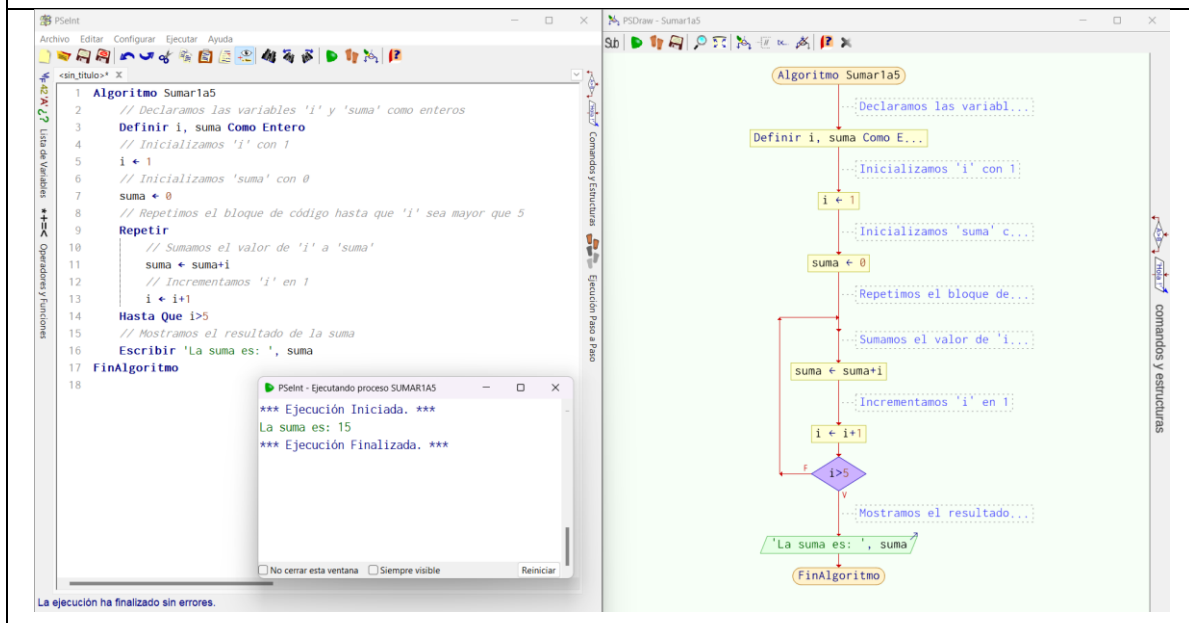
Pasos a Seguir:

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

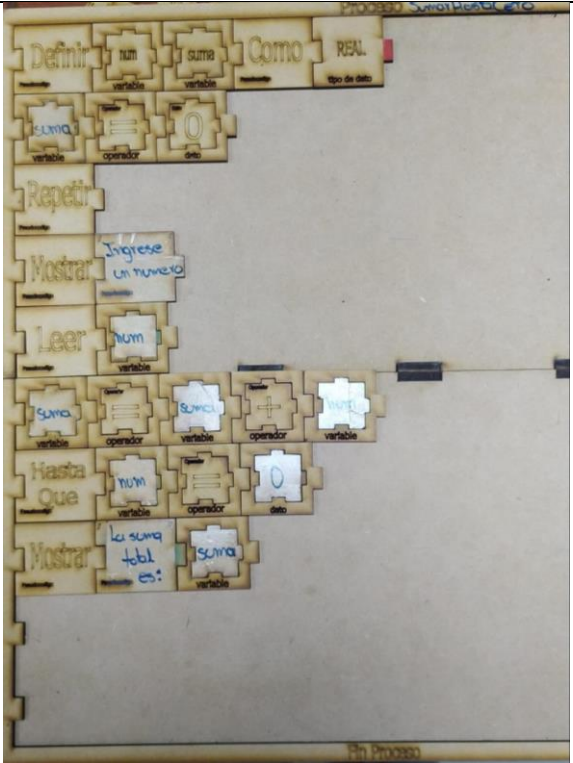
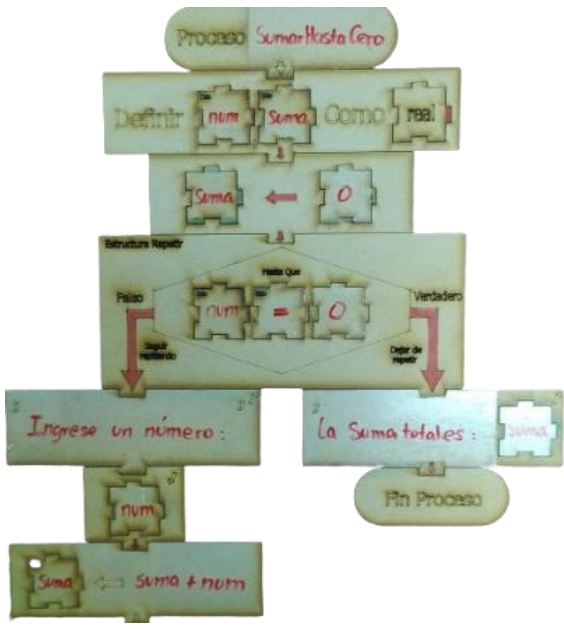


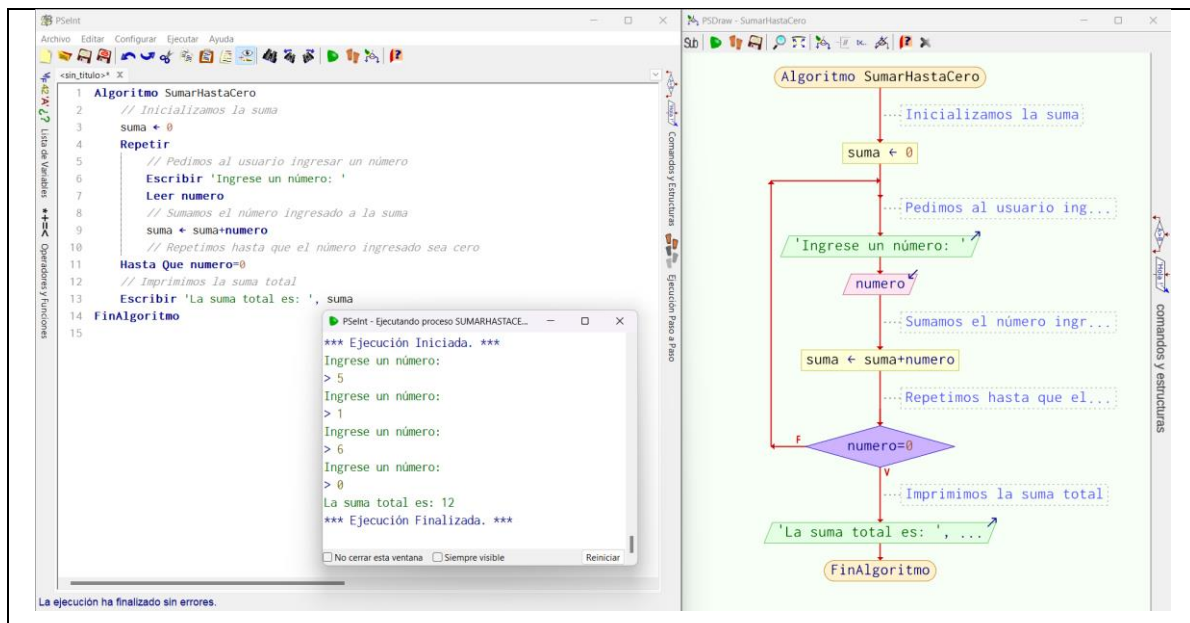
3.7.3 Sumar hasta ingresar cero

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
	
<p>Comprueba la solución en PSeInt</p>	



3.7.4 Verificar contraseña

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

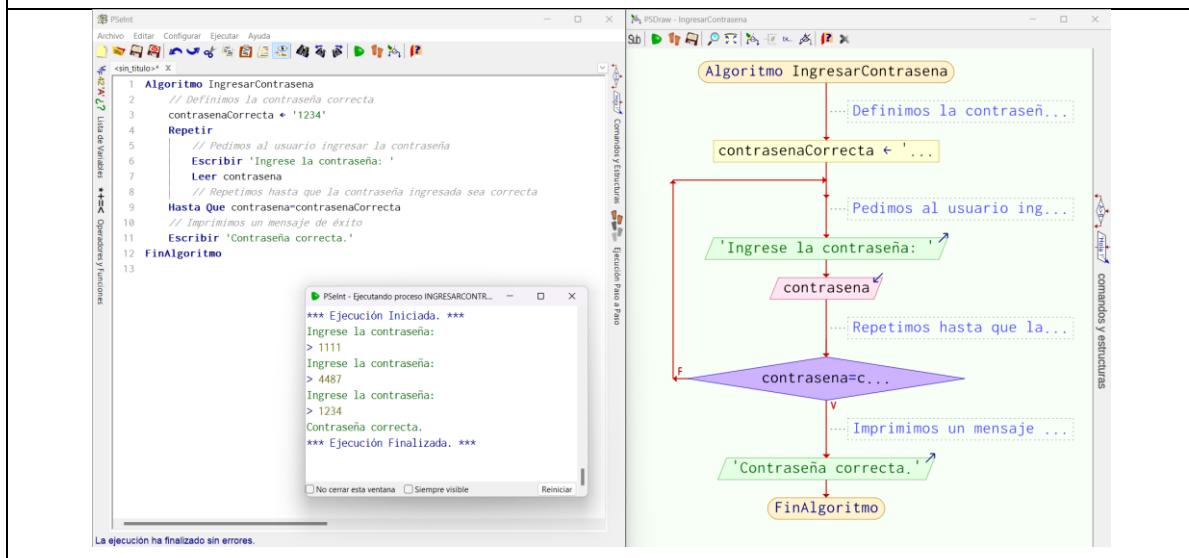
Pasos a Seguir:

1. Dar un nombre a este proceso: 'IngresarContraseña'.
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 PSeInt



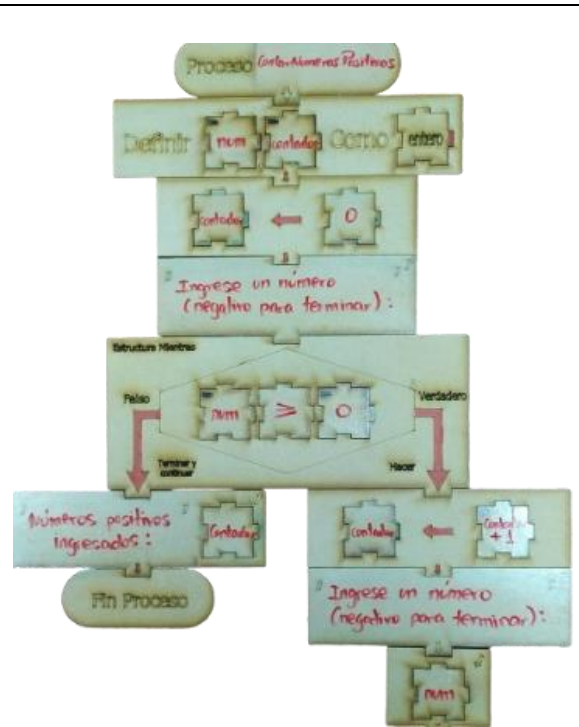
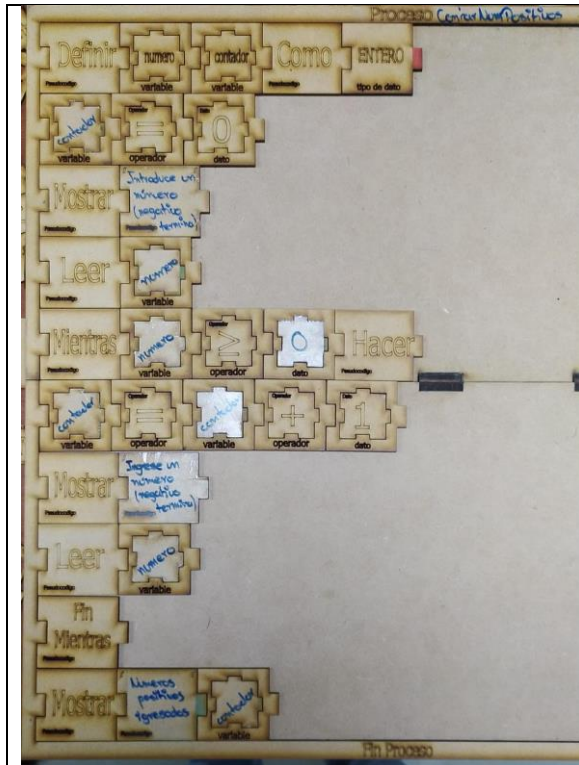
3.8.1 Contar números positivos

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

```

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

```

```

graph TD
    Inicio([Inicio]) --> Definir[Definir num, contador ...]
    Definir --> Inicializar[contador ← 0]
    Inicializar --> Introducir[Introduce un número (...)]
    Introducir --> Decidir{num ≥ 0}
    Decidir -- V --> Incrementar[contador ← contador + 1]
    Incrementar --> Introducir
    Decidir -- F --> Leer[Leer el siguiente número]
    Leer --> Mostrar[Mostrar Números positivos ingresados]
    Mostrar --> Fin([Fin])

```

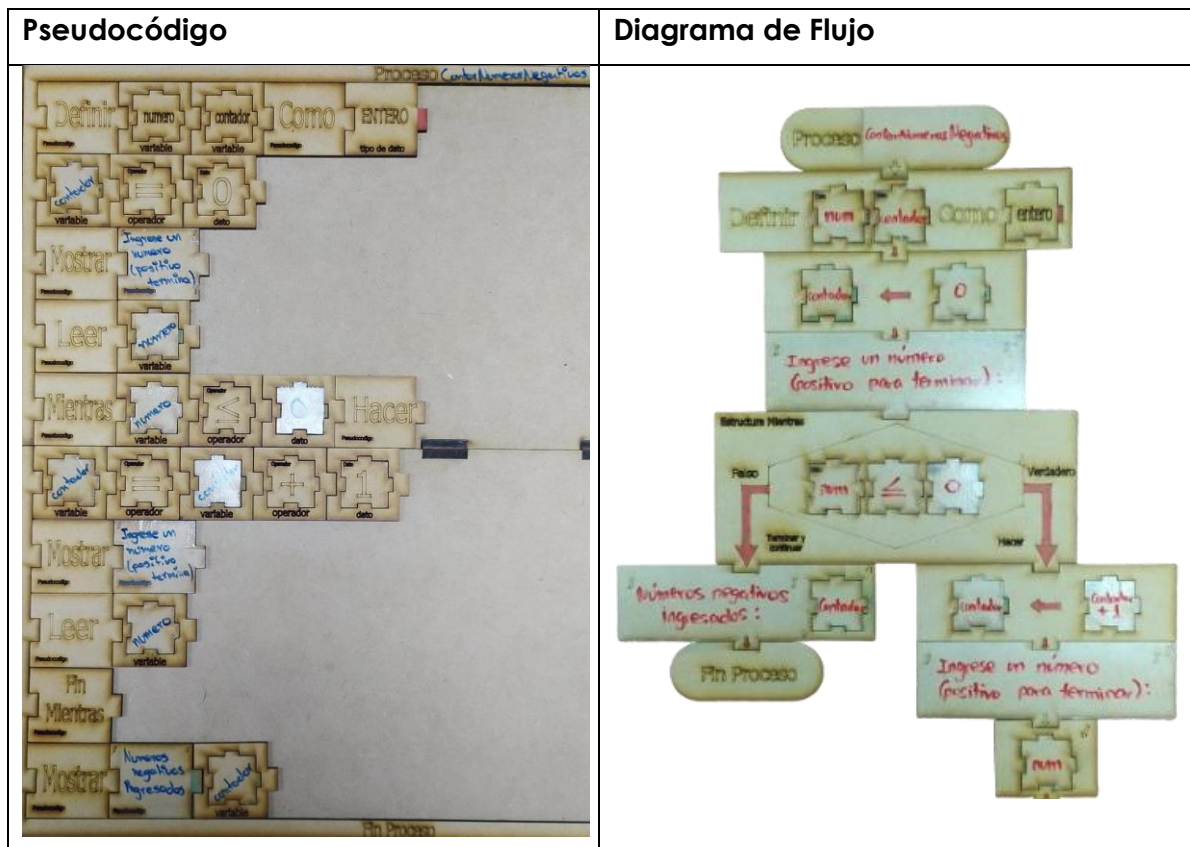
Contar números negativos

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.



Comprueba la solución en PSeInt

The screenshot displays the PSeInt interface. On the left, the pseudocode for 'Algoritmo ContarNumerosNegativos' is visible, including variable definitions, input prompts, and a loop structure. On the right, a corresponding flowchart illustrates the logic, starting with 'Definir num, contador', followed by 'contador ← 0', a loop 'Mientras nums0 Hacer', and ending with 'FinAlgoritmo'. A small window in the foreground shows the execution output, indicating that the process completed successfully without errors.

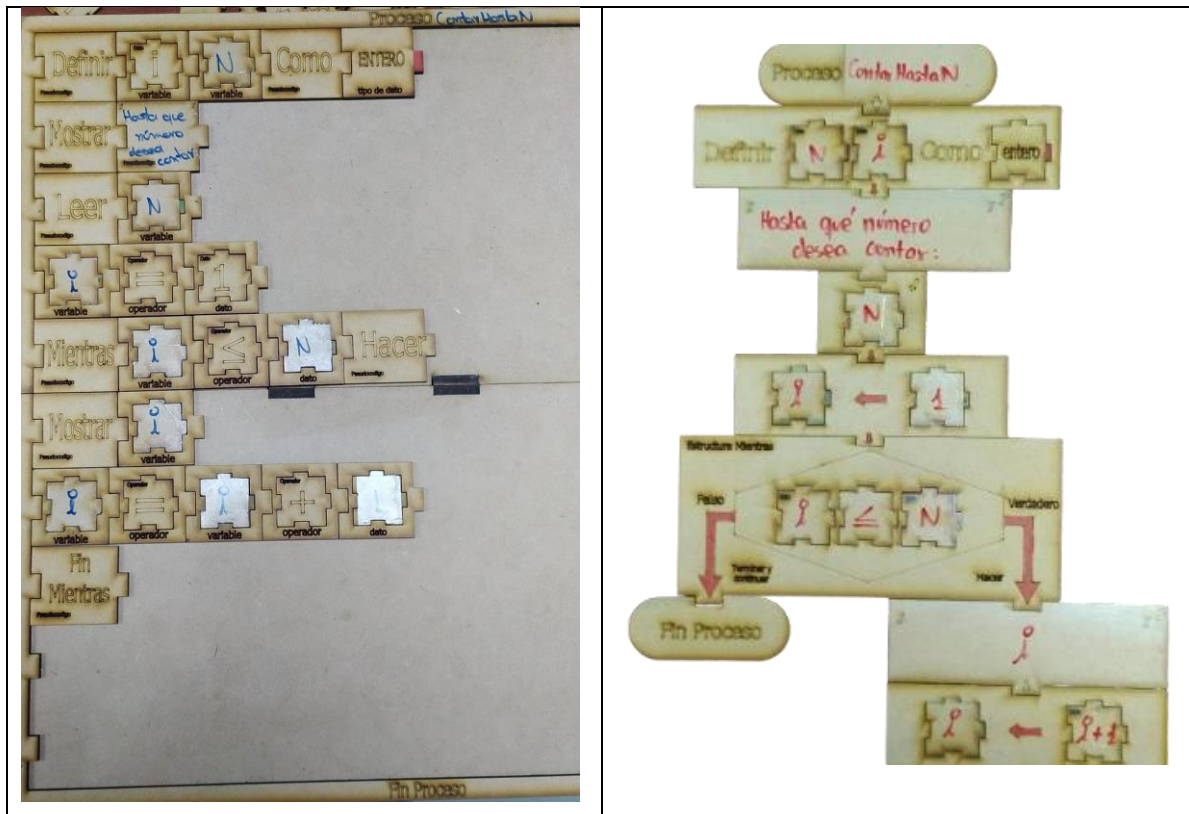
3.8.3 Contar hasta N

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

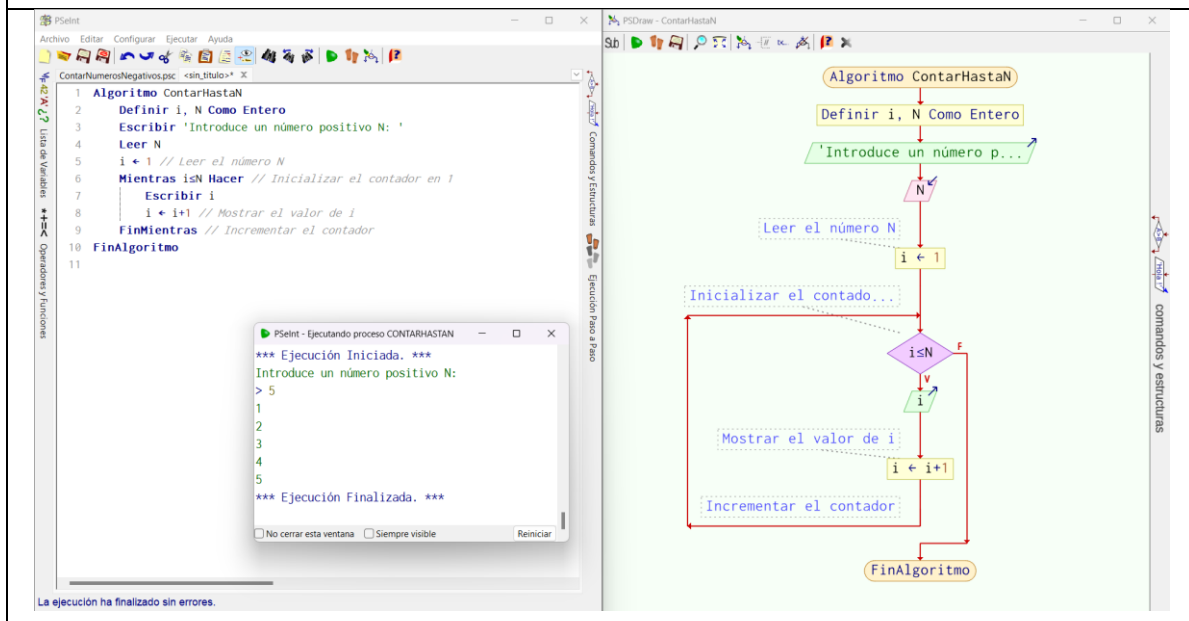
Pasos a Seguir:

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



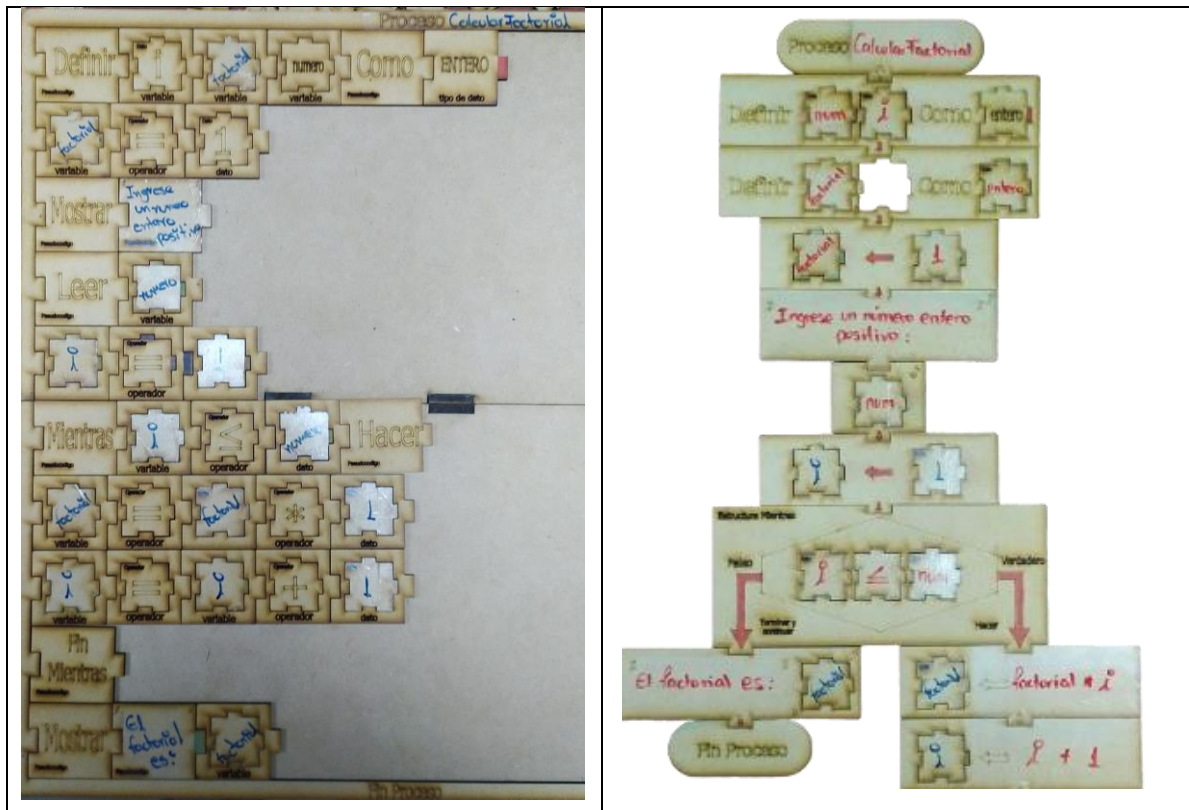
3.8.4 Calcular factorial

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

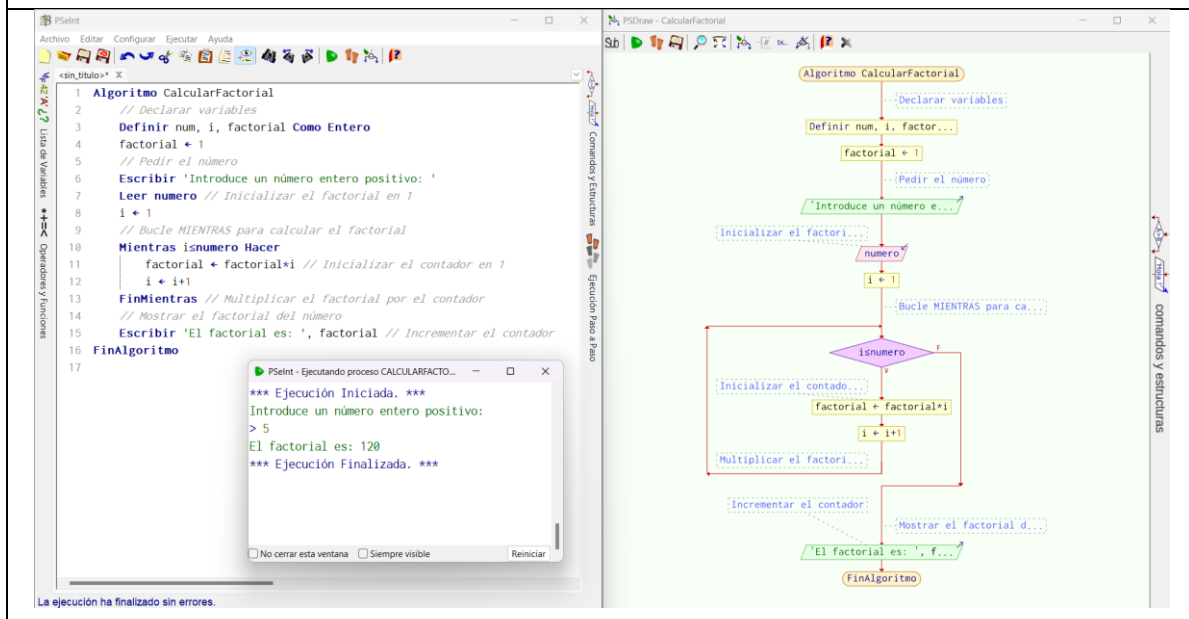
Pasos a Seguir:

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 PSeInt




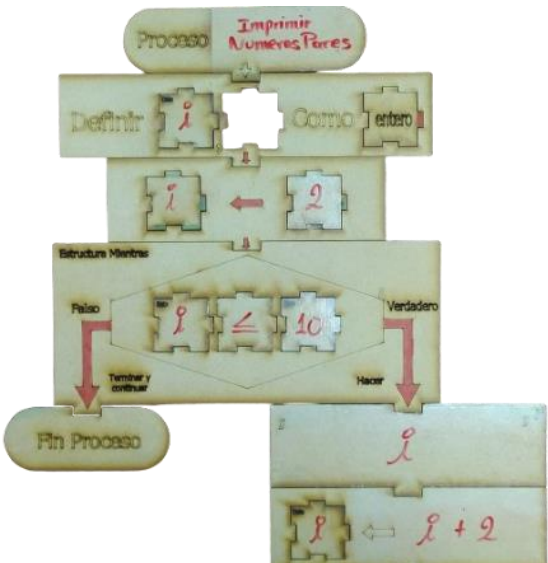
3.8.5 Imprimir números pares

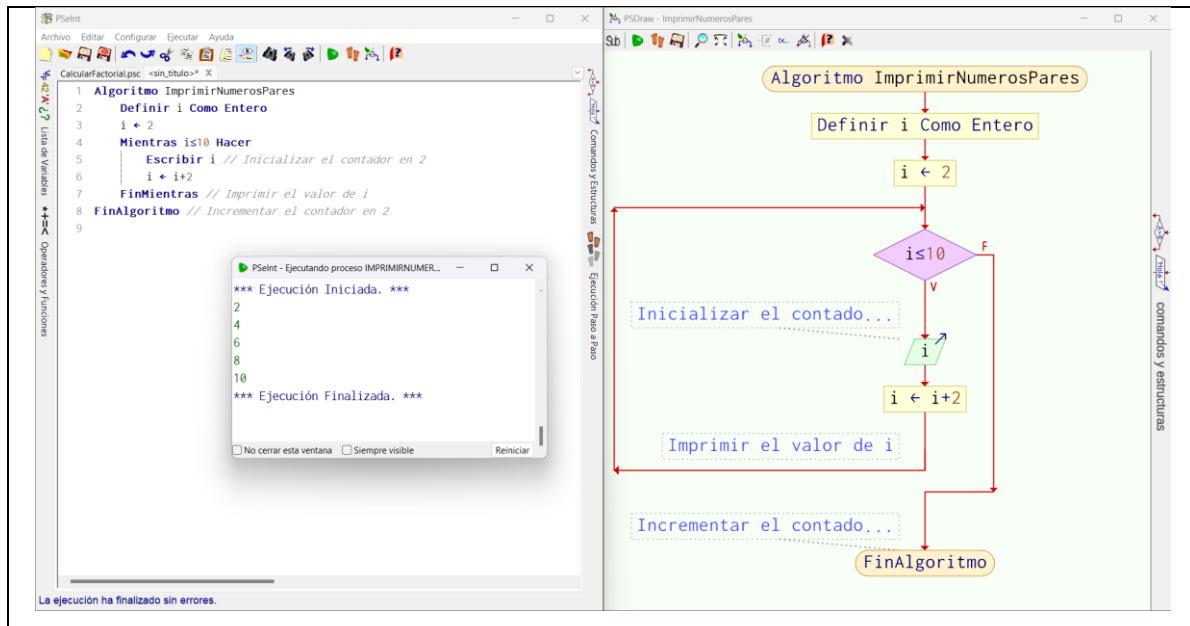
Crea un programa que imprima los números pares del 2 al 10.

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
	
<p>Comprueba la solución en PSeInt</p>	



3.9 Ejercicios con 'PARA'

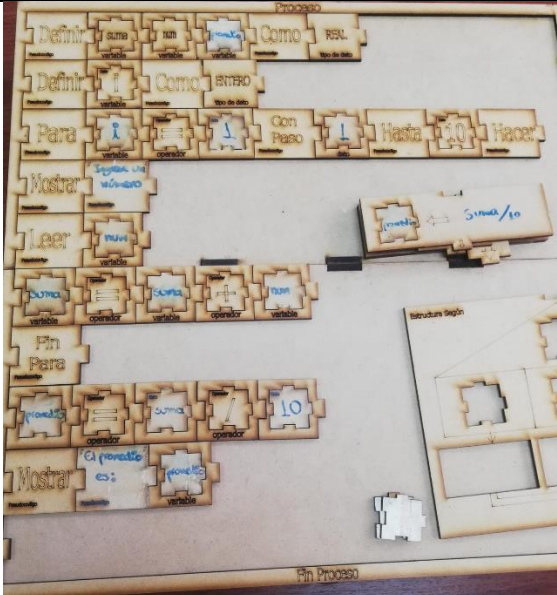
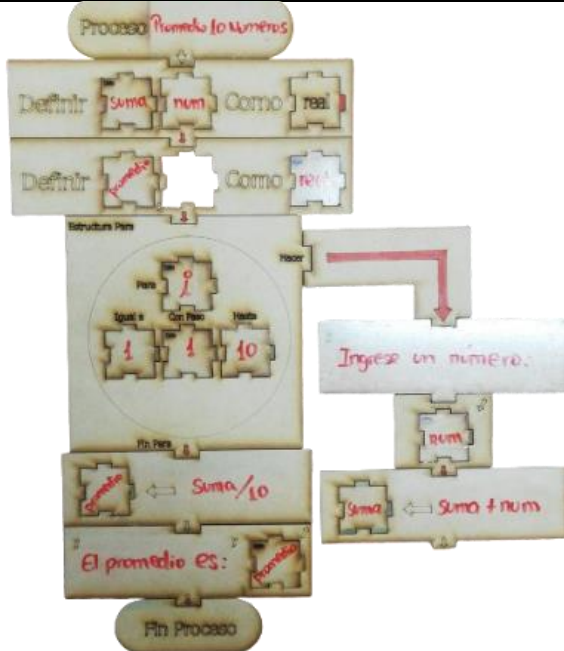
3.9.1 Calcular promedio de 10 números

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

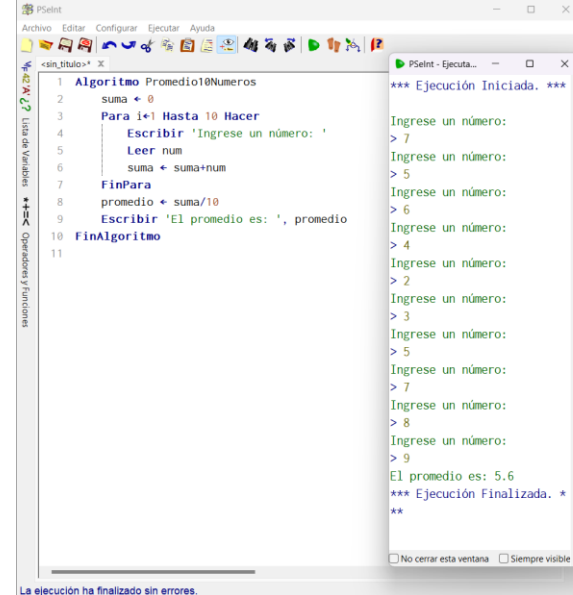
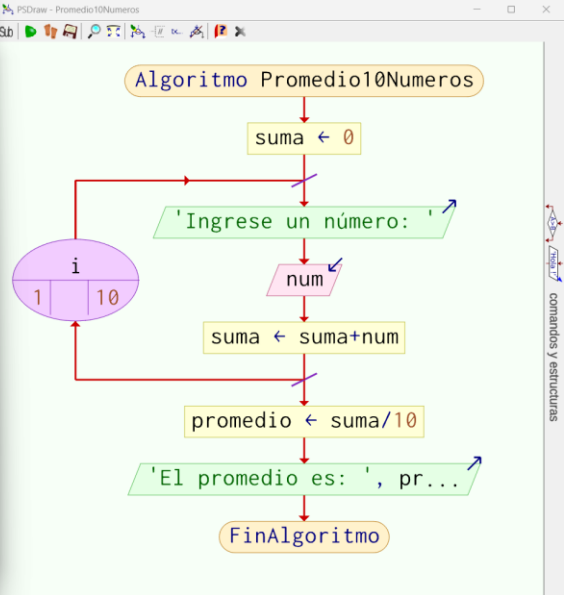
Pasos a Seguir:

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.
4. Dentro del bucle:
 - Muestra el mensaje 'Ingresa 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
	

Comprueba la solución en PSeInt

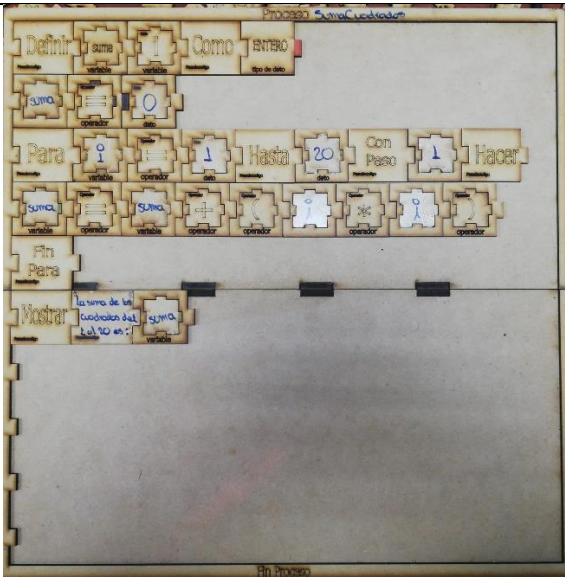
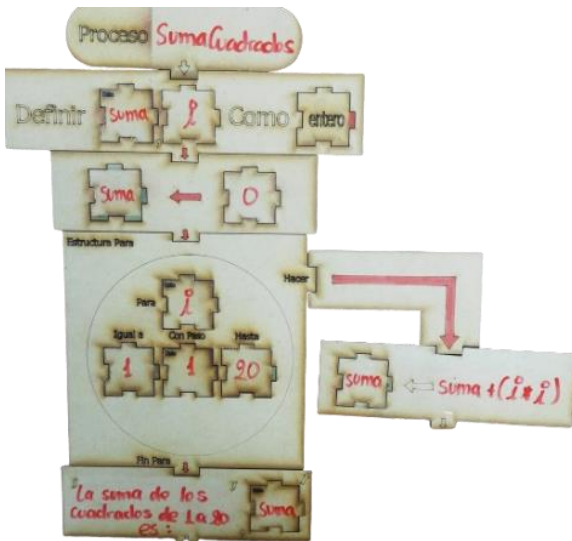
	
---	--

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
	
<p>Comprueba la solución en PSeInt</p>	

