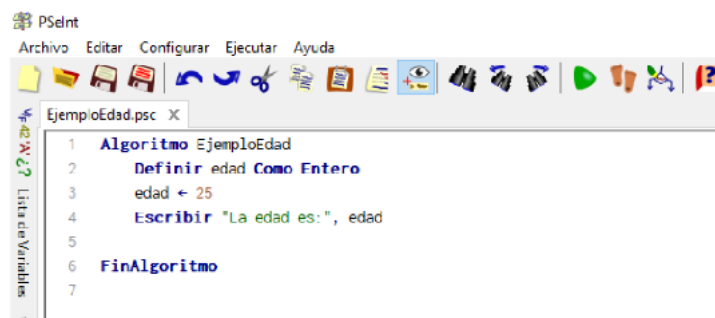




Realizar los siguientes ejercicios:

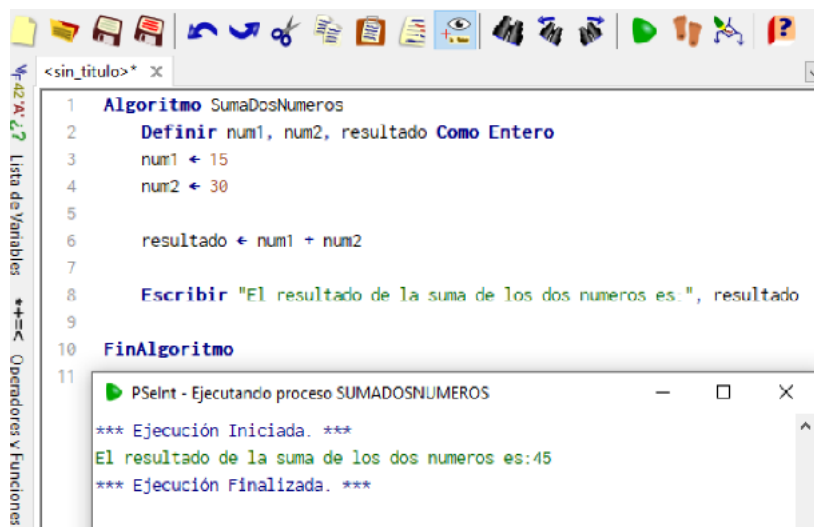
Ejercicios con Enteros

1. Declara una variable entera llamada `edad` y asígnale el valor 25.



```
1 Algoritmo EjemploEdad
2   Definir edad Como Entero
3   edad ← 25
4   Escribir "La edad es:", edad
5
6 FinAlgoritmo
7
```

2. Suma dos números enteros 15 y 30 y muestra el resultado.

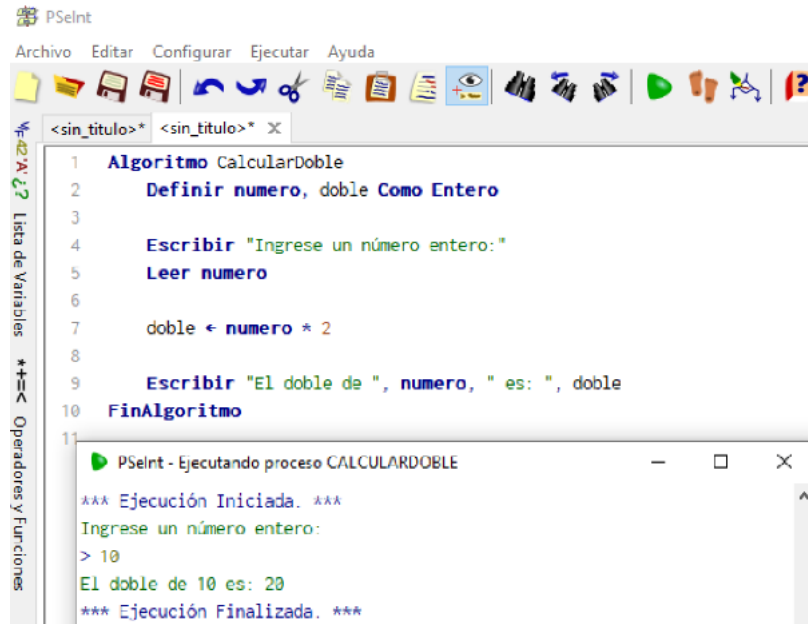


```
1 Algoritmo SumaDosNumeros
2   Definir num1, num2, resultado Como Entero
3   num1 ← 15
4   num2 ← 30
5
6   resultado ← num1 + num2
7
8   Escribir "El resultado de la suma de los dos numeros es.", resultado
9
10 FinAlgoritmo
11
```

PSeInt - Ejecutando proceso SUMADOSNUMEROS

```
*** Ejecución Iniciada. ***
El resultado de la suma de los dos numeros es:45
*** Ejecución Finalizada. ***
```

3. Calcula el doble de un número entero ingresado por el usuario.



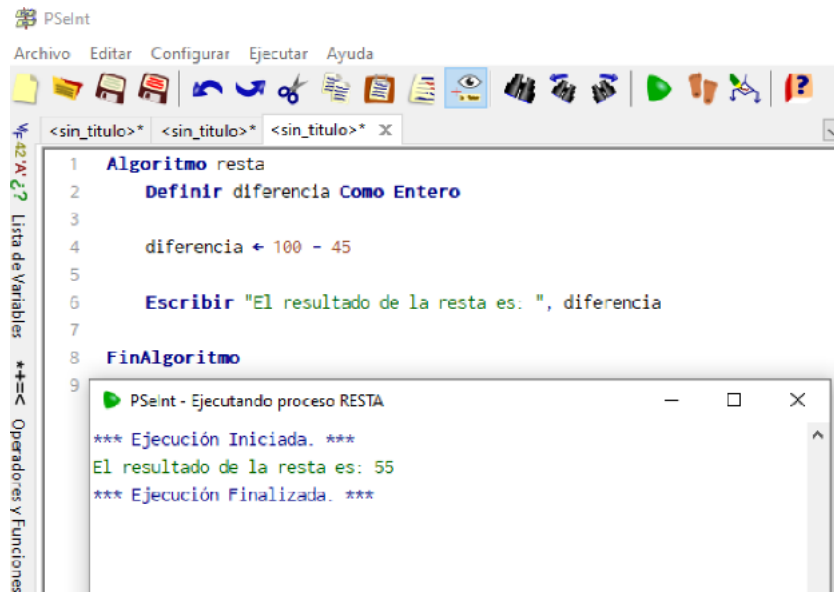
The screenshot shows the PSeInt IDE with a menu bar (Archivo, Editar, Configurar, Ejecutar, Ayuda) and a toolbar. The main editor displays the following pseudocode:

```
1  Algoritmo CalcularDoble
2    Definir numero, doble Como Entero
3
4    Escribir "Ingrese un número entero:"
5    Leer numero
6
7    doble ← numero * 2
8
9    Escribir "El doble de ", numero, " es: ", doble
10 FinAlgoritmo
```

Below the editor, a console window titled "PSeInt - Ejecutando proceso CALCULARDOBLE" shows the execution output:

```
*** Ejecución Iniciada. ***
Ingrese un número entero:
> 10
El doble de 10 es: 20
*** Ejecución Finalizada. ***
```

4. Resta $100 - 45$ y guarda el resultado en una variable llamada diferencia.



The screenshot shows the PSeInt IDE with a menu bar (Archivo, Editar, Configurar, Ejecutar, Ayuda) and a toolbar. The main editor displays the following pseudocode:

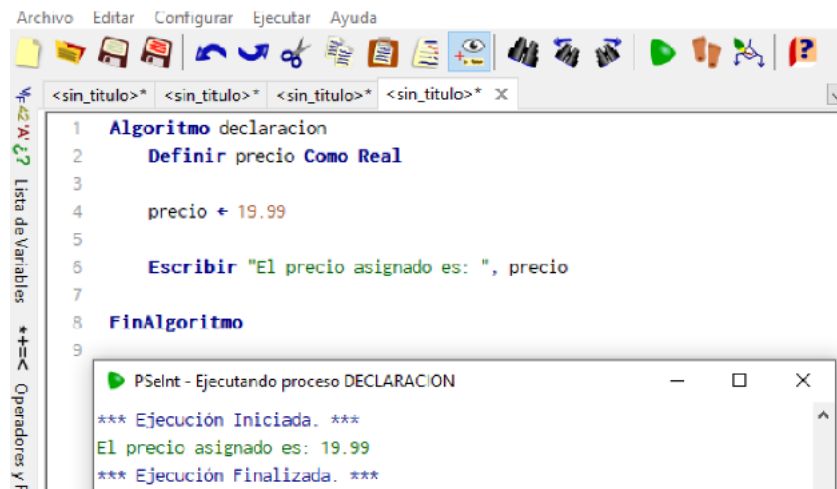
```
1  Algoritmo resta
2    Definir diferencia Como Entero
3
4    diferencia ← 100 - 45
5
6    Escribir "El resultado de la resta es: ", diferencia
7
8  FinAlgoritmo
```

Below the editor, a console window titled "PSeInt - Ejecutando proceso RESTA" shows the execution output:

```
*** Ejecución Iniciada. ***
El resultado de la resta es: 55
*** Ejecución Finalizada. ***
```

Ejercicios con Reales (Números Decimales)

5. Declara una variable `precio` y asígnale el valor 19.99.

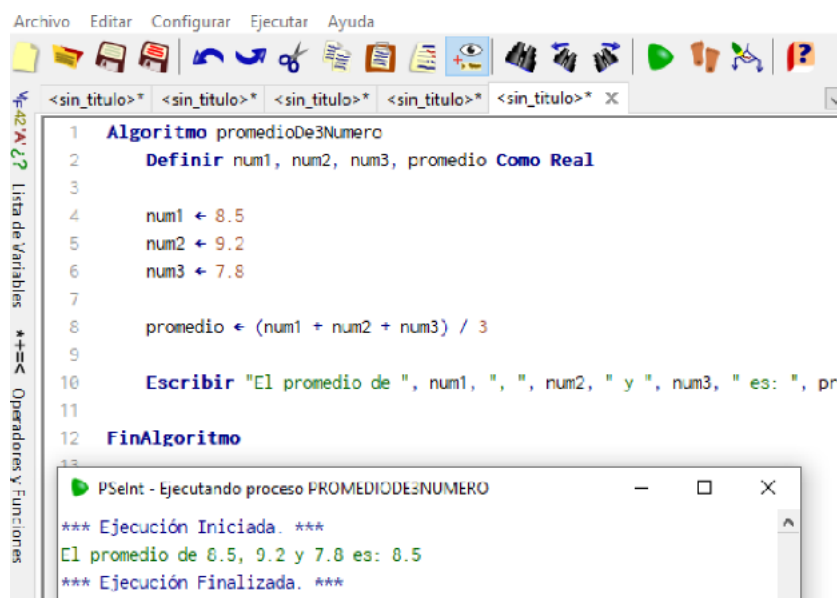


```
Archivo  Editar  Configurar  Ejecutar  Ayuda
<sin_titulo>* <sin_titulo>* <sin_titulo>* <sin_titulo>* X

1  Algoritmo declaracion
2      Definir precio Como Real
3
4      precio ← 19.99
5
6      Escribir "El precio asignado es: ", precio
7
8  FinAlgoritmo
9

PSeInt - Ejecutando proceso DECLARACION
*** Ejecución Iniciada. ***
El precio asignado es: 19.99
*** Ejecución Finalizada. ***
```

6. Calcula el promedio de tres números decimales 8.5, 9.2 y 7.8.

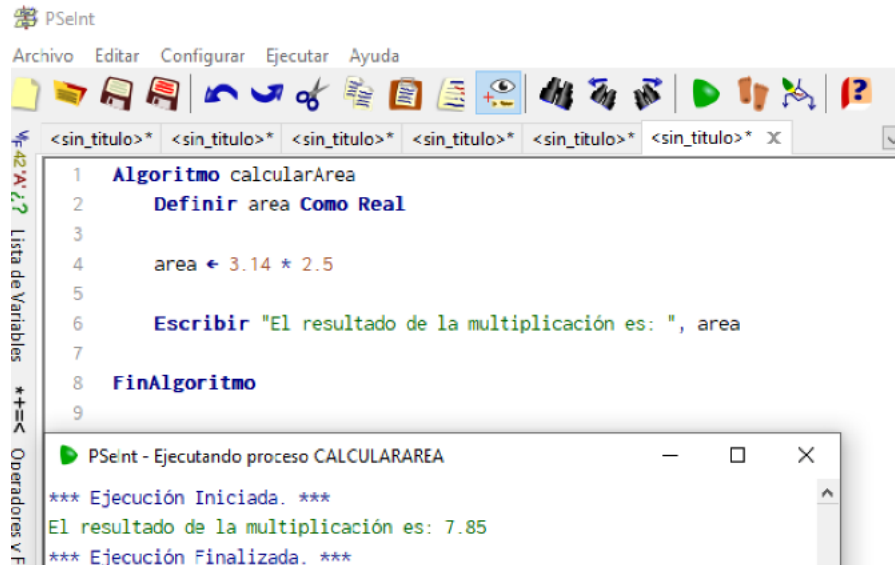


```
Archivo  Editar  Configurar  Ejecutar  Ayuda
<sin_titulo>* <sin_titulo>* <sin_titulo>* <sin_titulo>* <sin_titulo>* X

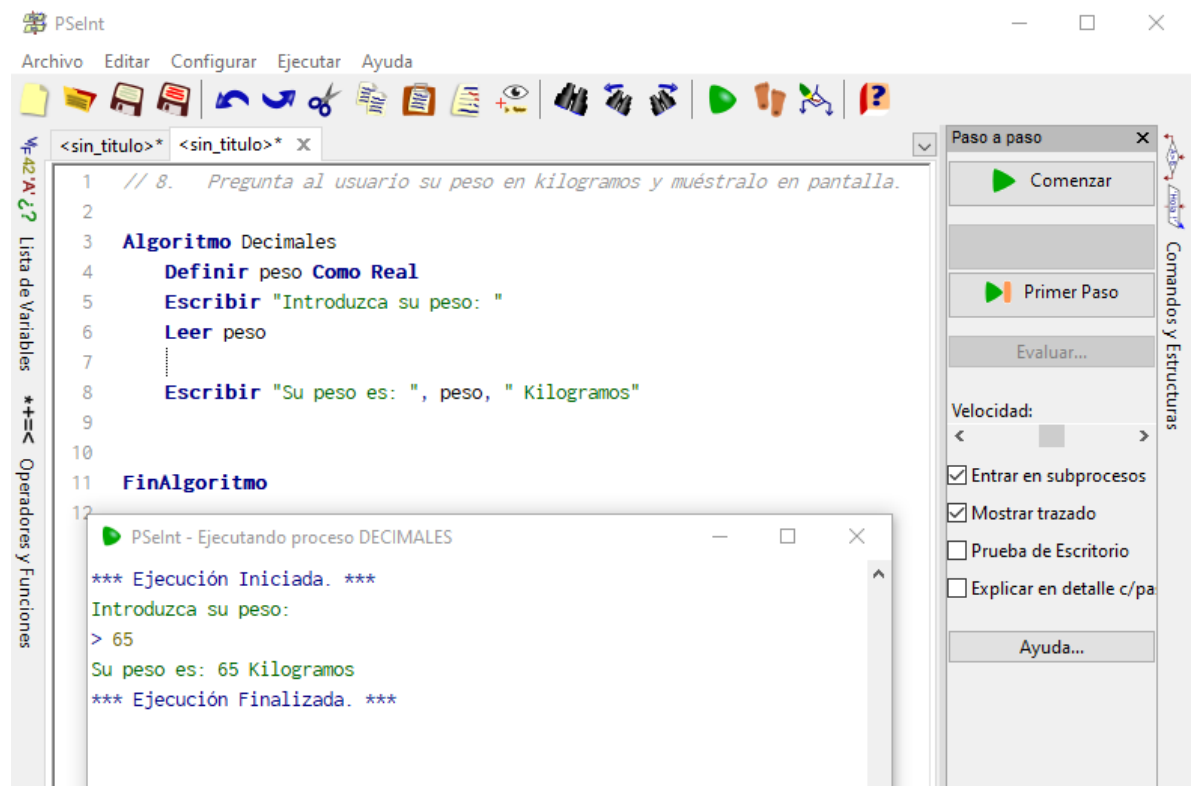
1  Algoritmo promedioDe3Numero
2      Definir num1, num2, num3, promedio Como Real
3
4      num1 ← 8.5
5      num2 ← 9.2
6      num3 ← 7.8
7
8      promedio ← (num1 + num2 + num3) / 3
9
10     Escribir "El promedio de ", num1, ", ", num2, " y ", num3, " es: ", promedio
11
12  FinAlgoritmo
13

PSeInt - Ejecutando proceso PROMEDIODE3NUMERO
*** Ejecución Iniciada. ***
El promedio de 8.5, 9.2 y 7.8 es: 8.5
*** Ejecución Finalizada. ***
```

7. Multiplica $3.14 * 2.5$ y guarda el resultado en area.



8. Pregunta al usuario su peso en kilogramos y muéstralo en pantalla.



Ejercicios con Valores Lógicos (Booleanos)

9. Declara una variable `esMayor` y asígnale `Verdadero` si `edad` es mayor de 18.

```

1 // 8. 9. Declara una variable esMayor y asigne Verdadero si edad es mayor de 18.
2 Algoritmo Booleanos
3   Definir edad Como Entero
4   Definir esMayor Como Logico
5
6   Escribir "Ingrese la edad: "
7   Leer edad
8
9   esMayor ← edad > 18
10
11  Escribir "Es mayor de edad?: ", esMayor
12
13 FinAlgoritmo
14

```

*** Ejecución Iniciada. ***
Ingrese la edad:
> 17
Es mayor de edad?: FALSO
*** Ejecución Finalizada. ***

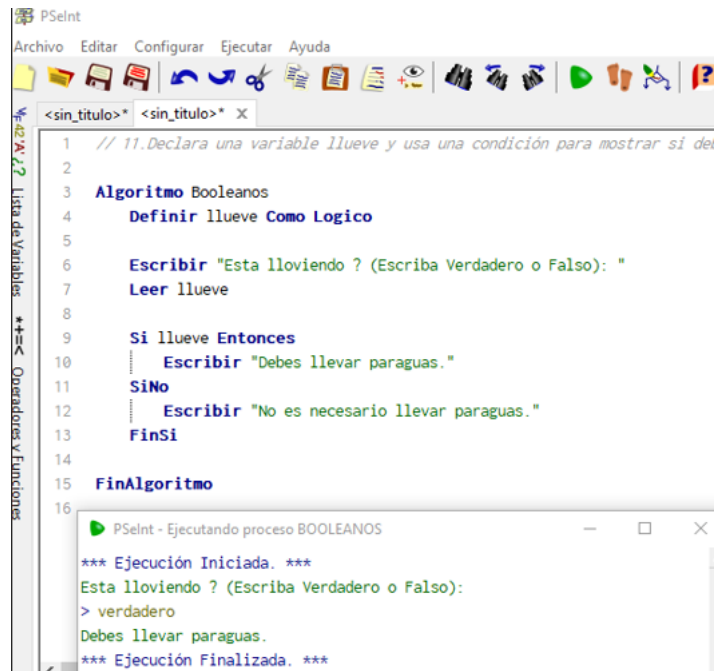
10. Crea un programa que verifique si un número ingresado es positivo o negativo.

```

1 // 9. 10. Crea un programa que verifique si un número ingresado es positivo o negativo.
2
3 Algoritmo Booleanos
4   Definir numero Como Real
5
6   Escribir "Ingrese un numero: "
7   Leer numero
8
9   Si numero > 0 Entonces
10    Escribir "El numero es positivo."
11  SiNo
12    Si numero < 0 Entonces
13      Escribir "El numero es negativo."
14    Sino
15      Escribir "El numero es cero."
16    FinSi
17  FinSi
18
19 FinAlgoritmo

```

11. Declara una variable `llueve` y usa una condición para mostrar si debes llevar paraguas.



```

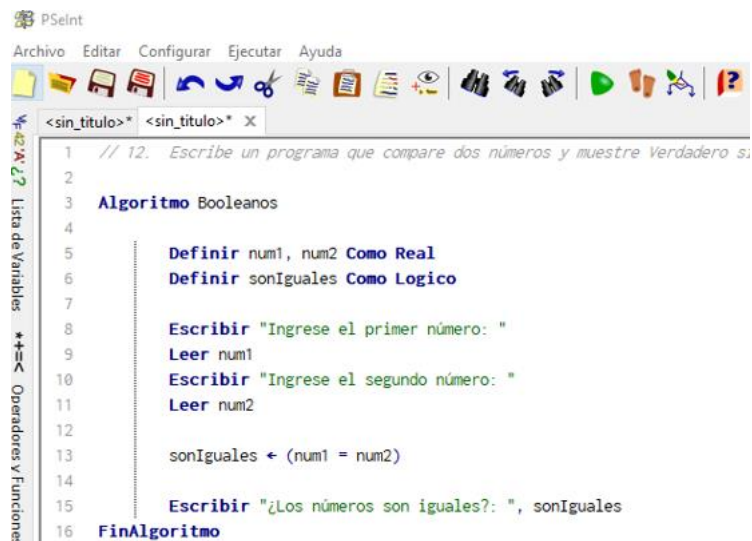
1 // 11. Declara una variable llueve y usa una condición para mostrar si debes
2
3 Algoritmo Booleanos
4     Definir llueve Como Logico
5
6     Escribir "Esta lloviendo ? (Escriba Verdadero o Falso): "
7     Leer llueve
8
9     Si llueve Entonces
10        Escribir "Debes llevar paraguas."
11    SiNo
12        Escribir "No es necesario llevar paraguas."
13    FinSi
14
15 FinAlgoritmo
  
```

Execution window output:

```

*** Ejecución Iniciada. ***
Esta lloviendo ? (Escriba Verdadero o Falso):
> verdadero
Debes llevar paraguas.
*** Ejecución Finalizada. ***
  
```

12. Escribe un programa que compare dos números y muestre Verdadero si son iguales.

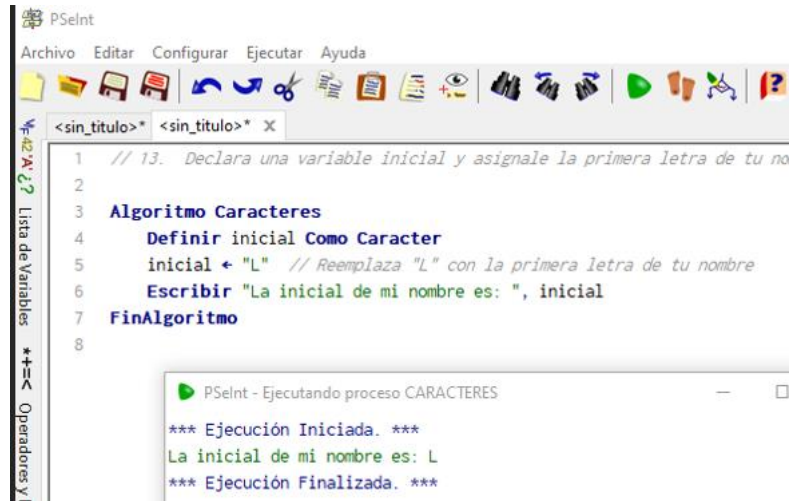


```

1 // 12. Escribe un programa que compare dos números y muestre Verdadero si
2
3 Algoritmo Booleanos
4
5     Definir num1, num2 Como Real
6     Definir sonIguales Como Logico
7
8     Escribir "Ingrese el primer número: "
9     Leer num1
10    Escribir "Ingrese el segundo número: "
11    Leer num2
12
13    sonIguales ← (num1 = num2)
14
15    Escribir "¿Los números son iguales?: ", sonIguales
16 FinAlgoritmo
  
```

Ejercicios con Caracteres

13. Declara una variable `inicial` y asígnale la primera letra de tu nombre.



```

1 // 13. Declara una variable inicial y asigne la primera letra de tu no
2
3 Algoritmo Caracteres
4   Definir inicial Como Caracter
5   inicial ← "L" // Reemplaza "L" con la primera letra de tu nombre
6   Escribir "La inicial de mi nombre es: ", inicial
7 FinAlgoritmo
8

```

PSeInt - Ejecutando proceso CARACTERES

```

*** Ejecución Iniciada. ***
La inicial de mi nombre es: L
*** Ejecución Finalizada. ***

```

14. Pide al usuario que ingrese una letra y muéstrala en pantalla.

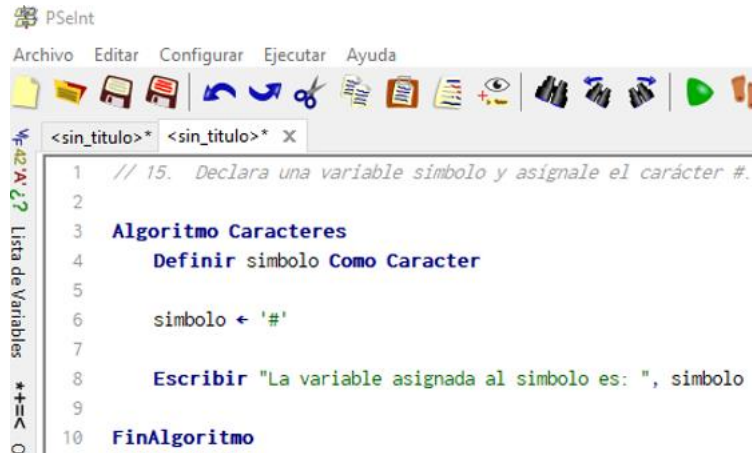


```

1 // 14. Pide al usuario que ingrese una letra y muéstrala
2
3 Algoritmo Caracteres
4   Definir letra Como Caracter
5   Escribir "Ingrese una letra: "
6   Leer letra
7
8   Escribir "La letra ingresada es: ", letra
9
10  FinAlgoritmo

```

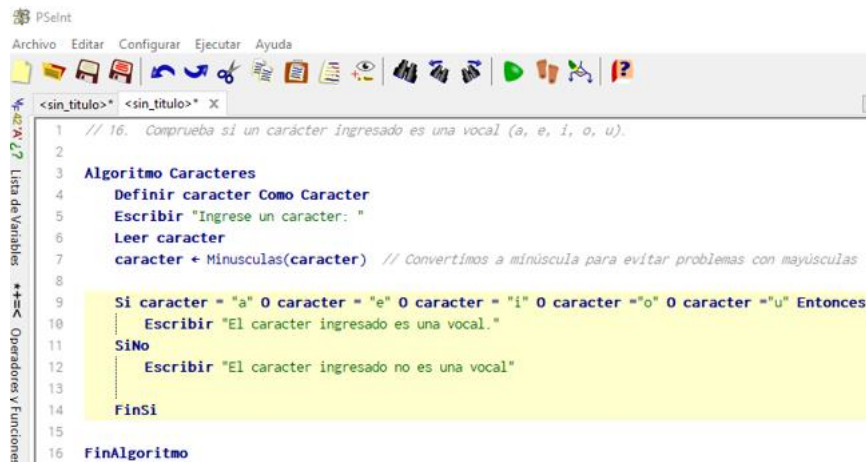
15. Declara una variable `simbolo` y asigne el carácter `#`.



```

1 // 15. Declara una variable simbolo y asigne el carácter #.
2
3 Algoritmo Caracteres
4     Definir simbolo Como Caracter
5
6     simbolo ← '#'
7
8     Escribir "La variable asignada al simbolo es: ", simbolo
9
10 FinAlgoritmo
  
```

16. Comprueba si un carácter ingresado es una vocal (a, e, i, o, u).

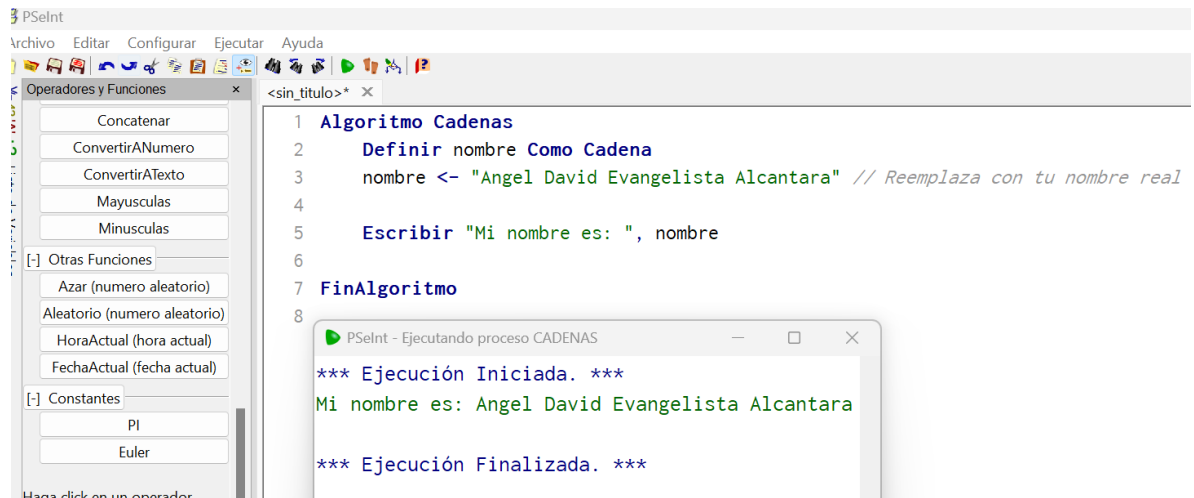


```

1 // 16. Comprueba si un carácter ingresado es una vocal (a, e, i, o, u).
2
3 Algoritmo Caracteres
4     Definir caracter Como Caracter
5     Escribir "Ingrese un caracter: "
6     Leer caracter
7     caracter ← Minusculas(caracter) // Convertimos a minúscula para evitar problemas con mayúsculas
8
9     Si caracter = "a" O caracter = "e" O caracter = "i" O caracter = "o" O caracter = "u" Entonces
10         Escribir "El caracter ingresado es una vocal."
11     SiNo
12         Escribir "El caracter ingresado no es una vocal"
13
14     FinSi
15
16 FinAlgoritmo
  
```


Ejercicios con Cadenas (Texto)

17. Declara una variable `nombre` y asígnale tu nombre completo.



18. Une dos cadenas "Hola" y "Mundo" para formar "Hola Mundo".



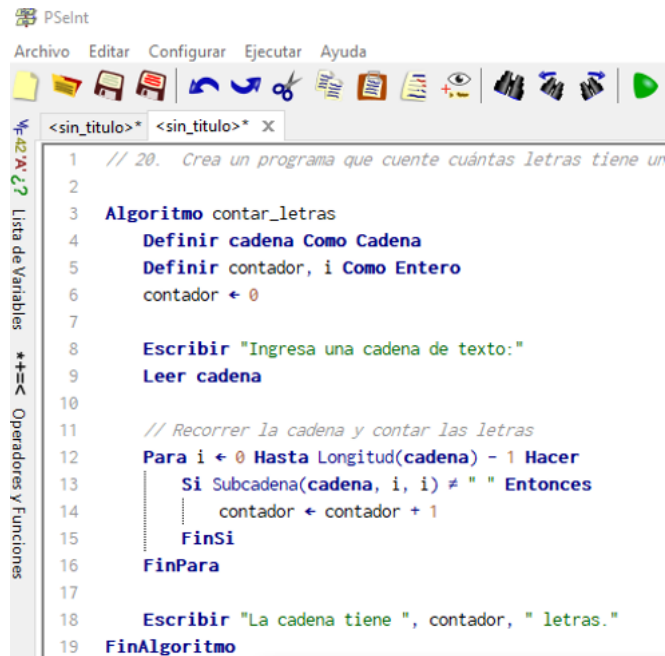
19. Pide al usuario su nombre y muéstralo junto con un mensaje de bienvenida.



```

1 // 19. Pide al usuario su nombre y muéstralo junto con un mensaje de bienvenida.
2
3 Algoritmo Cadenas
4     Definir nombre Como Cadena
5     Escribir "Ingrese su nombre: "
6     Leer nombre
7
8     Escribir "Bienvenido ", nombre
9
10
11 FinAlgoritmo
  
```

20. Crea un programa que cuente cuántas letras tiene una cadena ingresada.

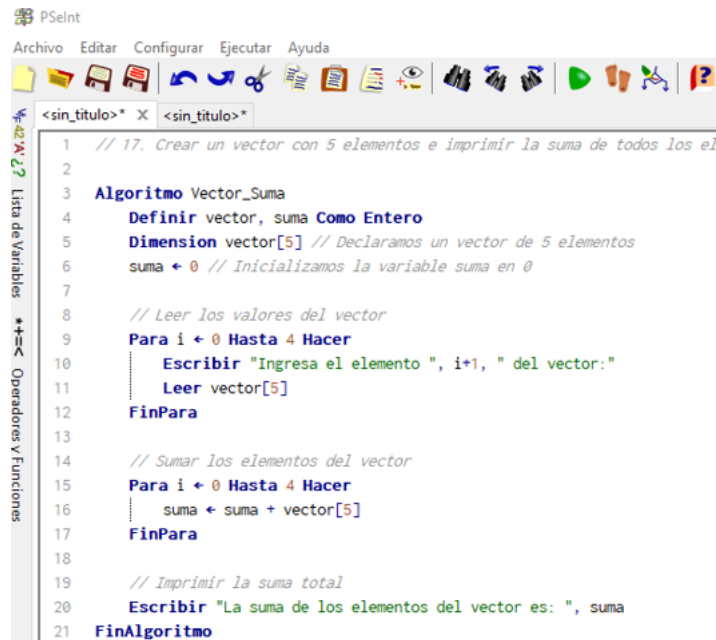


```

1 // 20. Crea un programa que cuente cuántas letras tiene una cadena ingresada.
2
3 Algoritmo contar_letras
4     Definir cadena Como Cadena
5     Definir contador, i Como Entero
6     contador ← 0
7
8     Escribir "Ingresa una cadena de texto:"
9     Leer cadena
10
11     // Recorrer la cadena y contar las letras
12     Para i ← 0 Hasta Longitud(cadena) - 1 Hacer
13         Si Subcadena(cadena, i, i) ≠ " " Entonces
14             contador ← contador + 1
15         FinSi
16     FinPara
17
18     Escribir "La cadena tiene ", contador, " letras."
19 FinAlgoritmo
  
```

Ejercicios con Vectores (Arreglos)

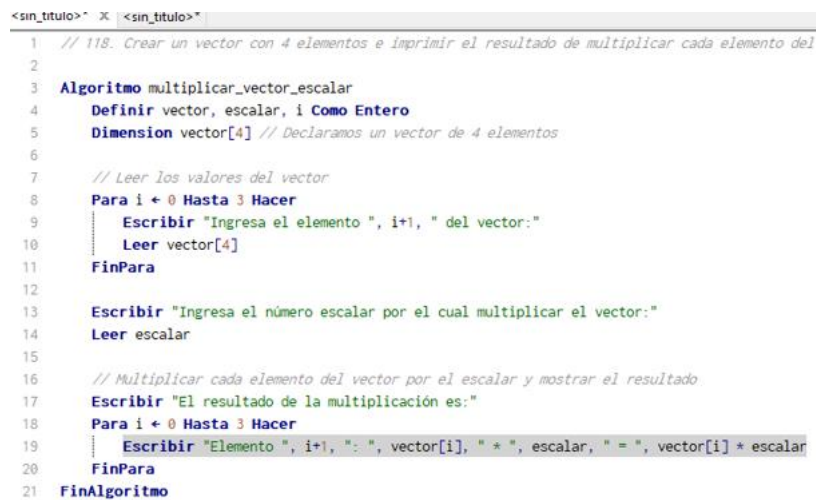
17. Crear un vector con 5 elementos e imprimir la suma de todos los elementos del vector.



```

1 // 17. Crear un vector con 5 elementos e imprimir la suma de todos los el
2
3 Algoritmo Vector_Suma
4   Definir vector, suma Como Entero
5   Dimension vector[5] // Declaramos un vector de 5 elementos
6   suma ← 0 // Inicializamos la variable suma en 0
7
8   // Leer los valores del vector
9   Para i ← 0 Hasta 4 Hacer
10      Escribir "Ingresa el elemento ", i+1, " del vector:"
11      Leer vector[i]
12   FinPara
13
14   // Sumar los elementos del vector
15   Para i ← 0 Hasta 4 Hacer
16      suma ← suma + vector[i]
17   FinPara
18
19   // Imprimir la suma total
20   Escribir "La suma de los elementos del vector es: ", suma
21 FinAlgoritmo
  
```

18. Crear un vector con 4 elementos e imprimir el resultado de multiplicar cada elemento del vector por un escalar.

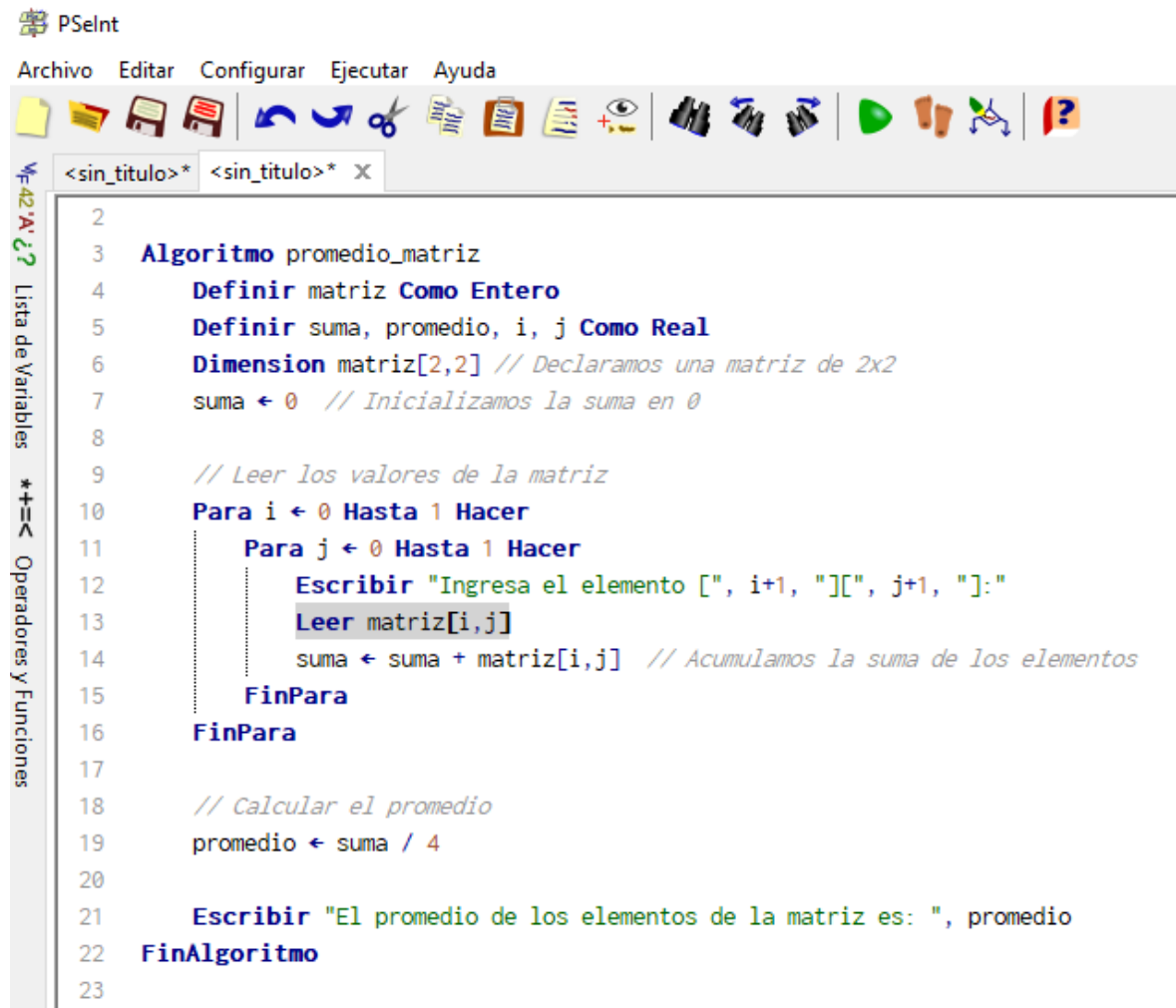


```

1 // 18. Crear un vector con 4 elementos e imprimir el resultado de multiplicar cada elemento del
2
3 Algoritmo multiplicar_vector_escalares
4   Definir vector, escalar, i Como Entero
5   Dimension vector[4] // Declaramos un vector de 4 elementos
6
7   // Leer los valores del vector
8   Para i ← 0 Hasta 3 Hacer
9      Escribir "Ingresa el elemento ", i+1, " del vector:"
10      Leer vector[i]
11   FinPara
12
13   Escribir "Ingresa el número escalar por el cual multiplicar el vector:"
14   Leer escalar
15
16   // Multiplicar cada elemento del vector por el escalar y mostrar el resultado
17   Escribir "El resultado de la multiplicación es:"
18   Para i ← 0 Hasta 3 Hacer
19      Escribir "Elemento ", i+1, ": ", vector[i], " * ", escalar, " = ", vector[i] * escalar
20   FinPara
21 FinAlgoritmo
  
```

Ejercicios con Matrices (Arreglos)

Crear una matriz de 2x2 e imprimir el promedio de todos sus elementos.



The image shows a screenshot of the PSeInt (Pseudointerpreter) software interface. The window title is "PSeInt". The menu bar includes "Archivo", "Editar", "Configurar", "Ejecutar", and "Ayuda". The toolbar contains various icons for file operations, editing, and execution. The main editor area displays a pseudocode algorithm for calculating the average of a 2x2 matrix. The algorithm is as follows:

```

2
3  Algoritmo promedio_matriz
4      Definir matriz Como Entero
5      Definir suma, promedio, i, j Como Real
6      Dimension matriz[2,2] // Declaramos una matriz de 2x2
7      suma ← 0 // Inicializamos la suma en 0
8
9      // Leer los valores de la matriz
10     Para i ← 0 Hasta 1 Hacer
11         Para j ← 0 Hasta 1 Hacer
12             Escribir "Ingresa el elemento [" , i+1, "][" , j+1, "]: "
13             Leer matriz[i,j]
14             suma ← suma + matriz[i,j] // Acumulamos la suma de los elementos
15         FinPara
16     FinPara
17
18     // Calcular el promedio
19     promedio ← suma / 4
20
21     Escribir "El promedio de los elementos de la matriz es: " , promedio
22 FinAlgoritmo
23
  
```

On the left side of the editor, there is a sidebar with the following options: "Lista de Variables" and "Operadores y Funciones".

Crear una matriz 2x3 y luego transponerla (convertir filas en columnas y viceversa).

```

1 Algoritmo TransponerMatriz
2 Definir matriz, transpuesta, i, j Como Real
3 Dimension matriz[2,3]
4 Dimension transpuesta[3,2]
5
6 // Asignamos valores a la matriz 2x3
7 matriz[0,0] = 1
8 matriz[0,1] = 2
9 matriz[0,2] = 3
10 matriz[1,0] = 4
11 matriz[1,1] = 5
12 matriz[1,2] = 6
13
14 // Mostramos la matriz original
15 Escribir "Matriz original (2x3):"
16 Para i = 0 Hasta 1 Hacer
17 | Para j = 0 Hasta 2 Hacer
18 | | Escribir Sin Saltar matriz[i,j], " "
19 | FinPara
20 | Escribir " " // Salto de línea
21 FinPara
22
23 // Transponemos la matriz (convertimos filas en columnas)
24 Para i = 0 Hasta 1 Hacer
25 | Para j = 0 Hasta 2 Hacer
26 | | transpuesta[j,i] = matriz[i,j]
27 | FinPara
28 FinPara
29
30 // Mostramos la matriz transpuesta (3x2)
31 Escribir "Matriz transpuesta (3x2):"
32 Para i = 0 Hasta 2 Hacer
33 | Para j = 0 Hasta 1 Hacer
34 | | Escribir Sin Saltar transpuesta[i,j], " "
35 | FinPara
36 | Escribir " " // Salto de línea
37 FinPara
38
39 FinAlgoritmo
40

```

```

22
23 // Transponemos la matriz (convertimos filas en columnas)
24 Para i = 0 Hasta 1 Hacer
25 | Para j = 0 Hasta 2 Hacer
26 | | transpuesta[j,i] = matriz[i,j]
27 | FinPara
28 FinPara
29
30 // Mostramos la matriz transpuesta (3x2)
31 Escribir "Matriz transpuesta (3x2):"
32 Para i = 0 Hasta 2 Hacer
33 | Para j = 0 Hasta 1 Hacer
34 | | Escribir Sin Saltar transpuesta[i,j], " "
35 | FinPara
36 | Escribir " " // Salto de línea
37 FinPara
38
39 FinAlgoritmo
40

```