

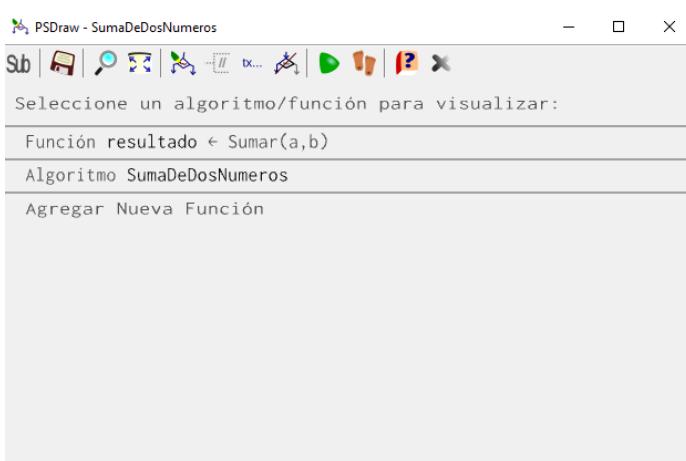
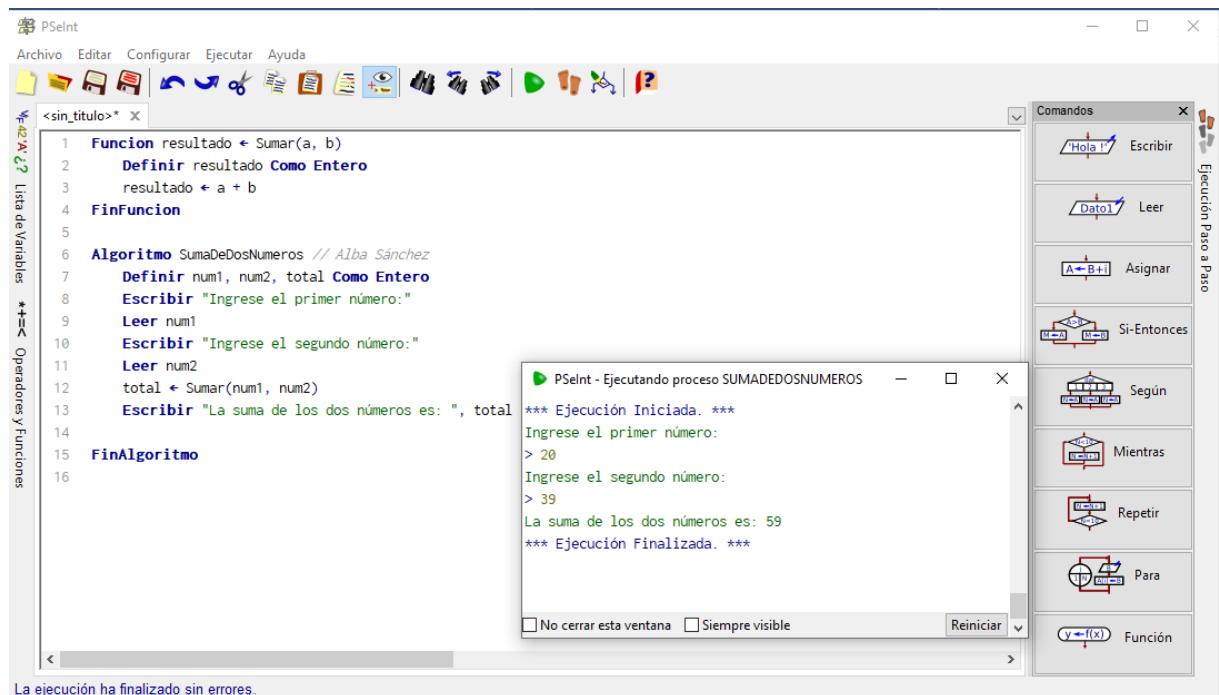
## Ejercicios de Funciones y Vectores en PSeInt

Este documento contiene ejercicios básicos de PSeInt para practicar el uso de funciones y vectores.

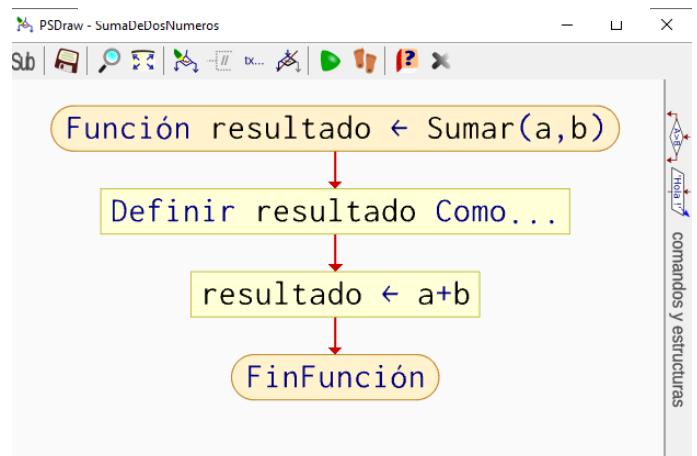
### Ejercicios de Funciones

#### 1. Suma de dos números:

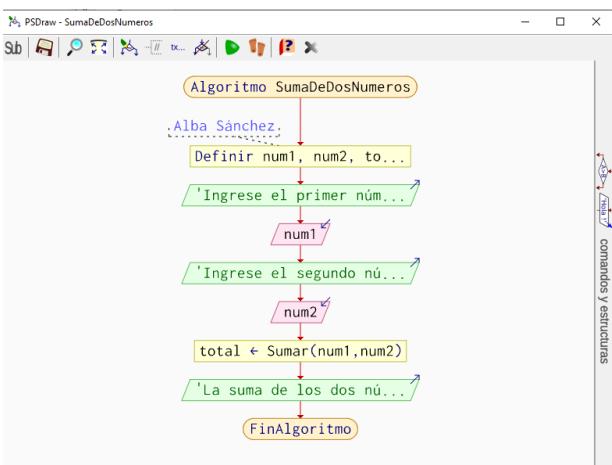
Crea una función llamada 'Sumar' que reciba dos números enteros y devuelva su suma.  
Muestra el resultado en el algoritmo principal.



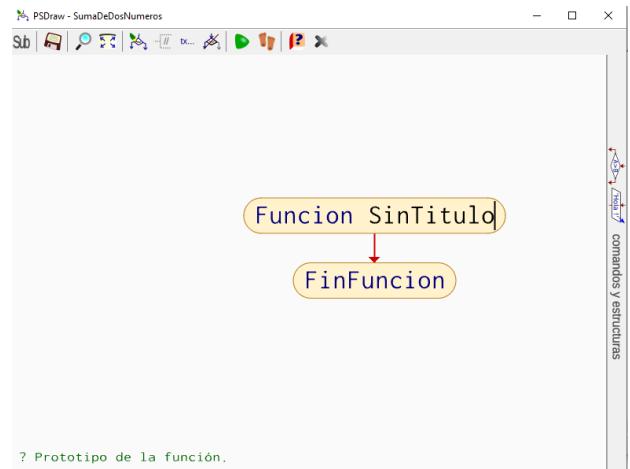
#### Función resultado <- Sumar(a, b)



## Algoritmo Suma\_Dos\_Numeros



## Agregar nueva Función



## 2. Número par o impar:

Crea una función 'EsPar' que reciba un número entero y devuelva 'Verdadero' si es par o 'Falso' si es impar.

La ejecución ha finalizado sin errores.

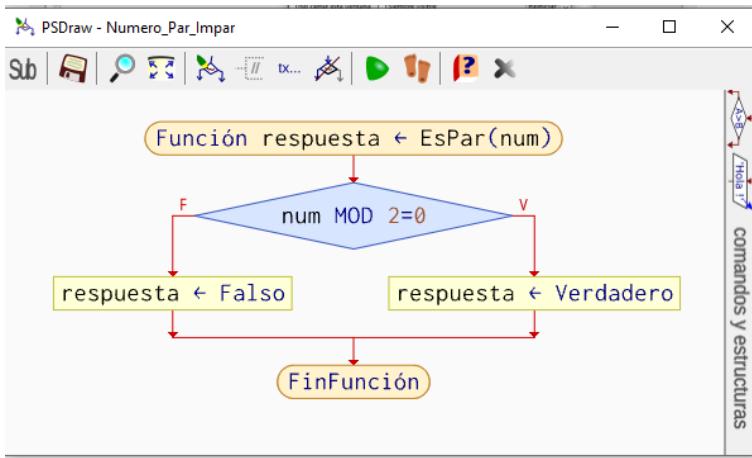
```

1 Funcion respuesta ← EsPar(numero)
2 Si num MOD 2 = 0 Entonces
3   respuesta ← Verdadero
4 SiNo
5   respuesta ← Falso
6 FinSi
7 FinFuncion
8
9 Algoritmo Numero_Par_Impar
10 Definir numero Como Entero
11
12 Escribir "Ingrese un número:"
13 Leer numero
14
15 Si EsPar(numero) Entonces
16   Escribir "El número es PAR."
17 SiNo
18   Escribir "El número es IMPAR."
19 FinSi
20
21 FinAlgoritmo
22
    
```

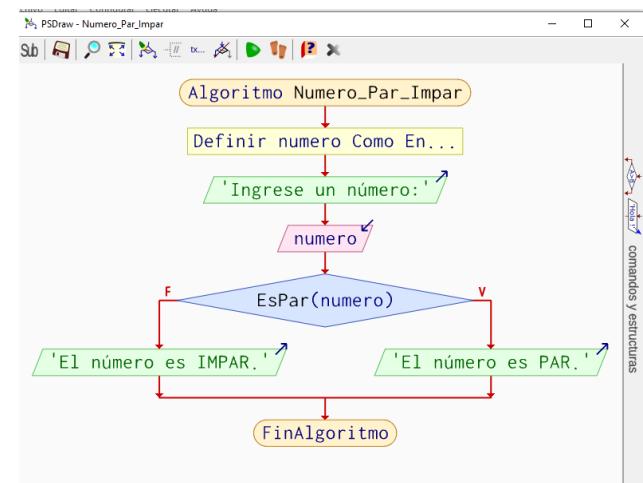
Seleccione un algoritmo/función para visualizar:

- Función respuesta ← EsPar(numero)
- Algoritmo Numero\_Par\_Impar
- Agregar Nueva Función

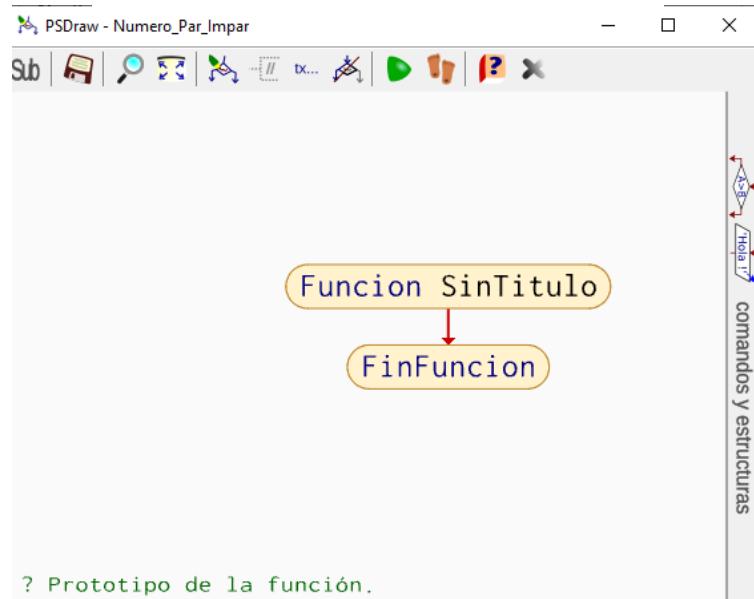
## Función respuesta <- EsPar(num)



## Algoritmo Numero\_Par\_Impar



## Agregar nueva Función



### 3. Calcular el promedio:

Diseña una función 'Promedio' que reciba tres notas y devuelva el promedio. El algoritmo principal debe mostrar si el estudiante aprobó ( $\geq 70$ ) o reprobó ( $<70$ ).

The screenshot shows the PSeInt Integrated Development Environment (IDE) interface. On the left, the code editor displays an algorithm in pseudocode. On the right, the execution window shows the step-by-step execution of the algorithm, including user input for grades and the final output indicating the student passed.

```

1 Función resultado ← CalcularPromedio(n1, n2, n3)
2
3 Definir resultado Como Real
4
5 resultado ← (n1 + n2 + n3) / 3
6 FinFunción
7
8 Algoritmo PromedioEstudiante //Alba Sánchez
9   Definir nota1, nota2, nota3, prom Como Real
10
11   Escribir "Ingrese la primera nota:"
12   Leer nota1
13   Escribir "Ingrese la segunda nota:"
14   Leer nota2
15   Escribir "Ingrese la tercera nota:"
16   Leer nota3
17
18   prom ← CalcularPromedio(nota1, nota2, nota3)
19
20   Escribir "El promedio es: ", prom
21
22   Si prom ≥ 70 Entonces
23     Escribir "El estudiante aprobó."
24   Sino
25     Escribir "El estudiante reprobó."
26   FinSi
27
28 FinAlgoritmo
  
```

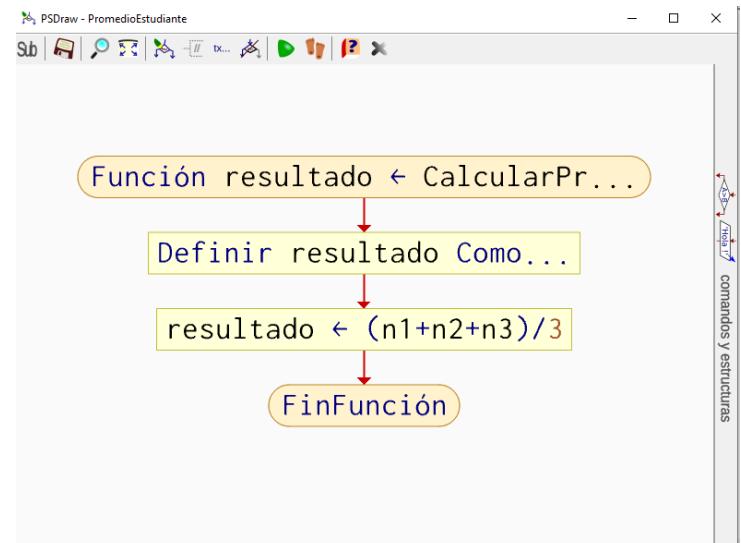
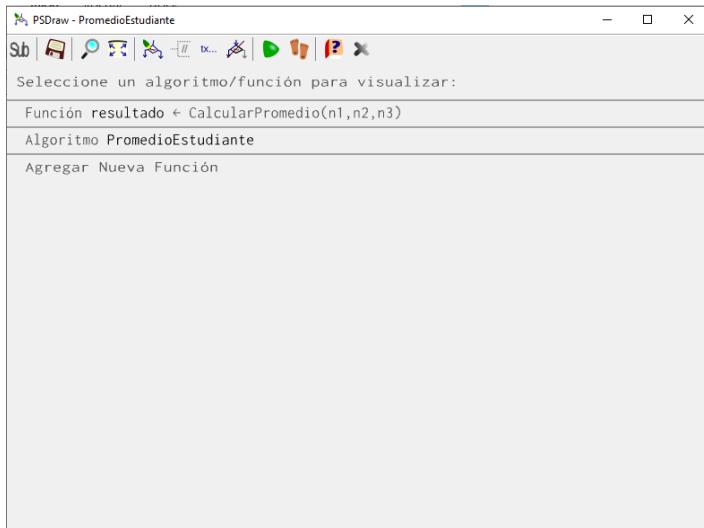
La ejecución ha finalizado sin errores.

Ejecución:

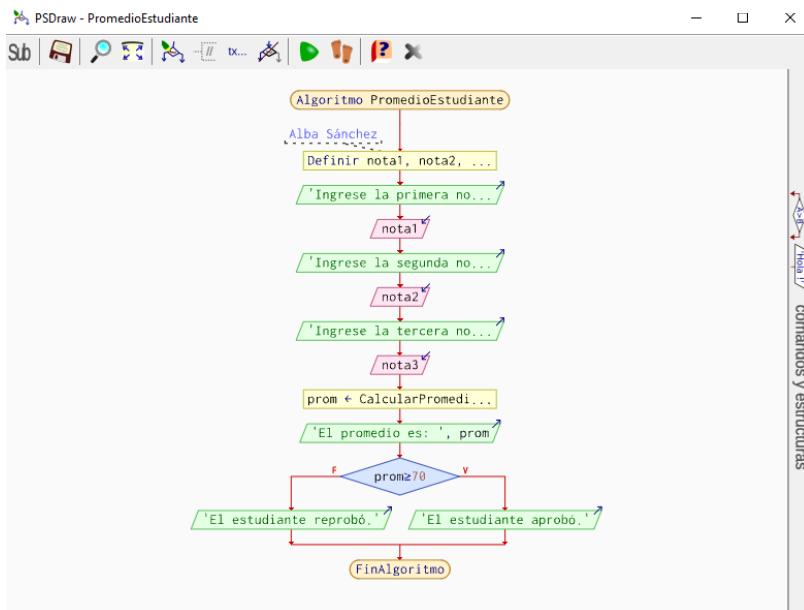
```

*** Ejecución Iniciada. ***
Ingrese la primera nota:
> 87
Ingrese la segunda nota:
> 98
Ingrese la tercera nota:
> 90
El promedio es: 91.66666666666667
El estudiante aprobó.
*** Ejecución Finalizada. ***
  
```

### Función resultado <- CalcularPromedio(n1, n2, n3)



## Algoritmo PromedioEstudiante



#### 4. Contraseña segura:

Crea una función 'ValidarPassword' que reciba una cadena y devuelva 'Verdadero' si la contraseña tiene más de 8 caracteres. El algoritmo principal debe mostrar si la contraseña es válida o no.

```

1 Función valida ← ValidaPassword(clave)
2   Definir valida Como Lógico
3   Si Longitud(clave) > 8 Entonces
4     valida ← Verdadero
5   Sino
6     valida ← Falso
7   FinSi
8 FinFunción

Algoritmo ValidaContraseña //Alba Sánchez
9   Definir password Como Cadena
10  Escribir "Ingrese una contraseña."
11  Leer password
12  Si ValidaPassword(password) Entonces
13    Escribir "La contraseña es válida (segura)."
14  Sino
15    Escribir "La contraseña NO es válida (debe tener más de 8 caracteres)."
16  FinSi
17 FinAlgoritmo
20
    
```

Ejecución:

```

*** Ejecución Iniciada. ***
Ingrese una contraseña:
> A65431k00PW
La contraseña es válida (segura).
*** Ejecución Finalizada. ***
    
```

La ejecución ha finalizado sin errores.

```

1 Función valida ← ValidaPassword(clave)
2   Definir valida Como Lógico
3   Si Longitud(clave) > 8 Entonces
4     valida ← Verdadero
5   Sino
6     valida ← Falso
7   FinSi
8 FinFunción

Algoritmo ValidaContraseña //Alba Sánchez
9   Definir password Como Cadena
10  Escribir "Ingrese una contraseña."
11  Leer password
12  Si ValidaPassword(password) Entonces
13    Escribir "La contraseña es válida (segura)."
14  Sino
15    Escribir "La contraseña NO es válida (debe tener más de 8 caracteres)."
16  FinSi
17 FinAlgoritmo
20
    
```

Ejecución:

```

*** Ejecución Iniciada. ***
Ingrese una contraseña:
> 1235
La contraseña NO es válida (debe tener más de 8 caracteres).
*** Ejecución Finalizada. ***
    
```

Selección de algoritmo/función para visualizar:

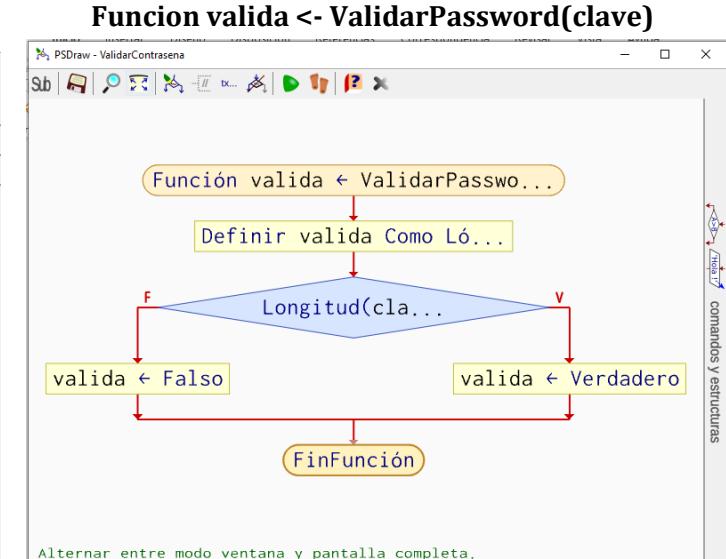
Función valida ← ValidaPassword(clave)

Algoritmo ValidaContraseña

Agregar Nueva Función

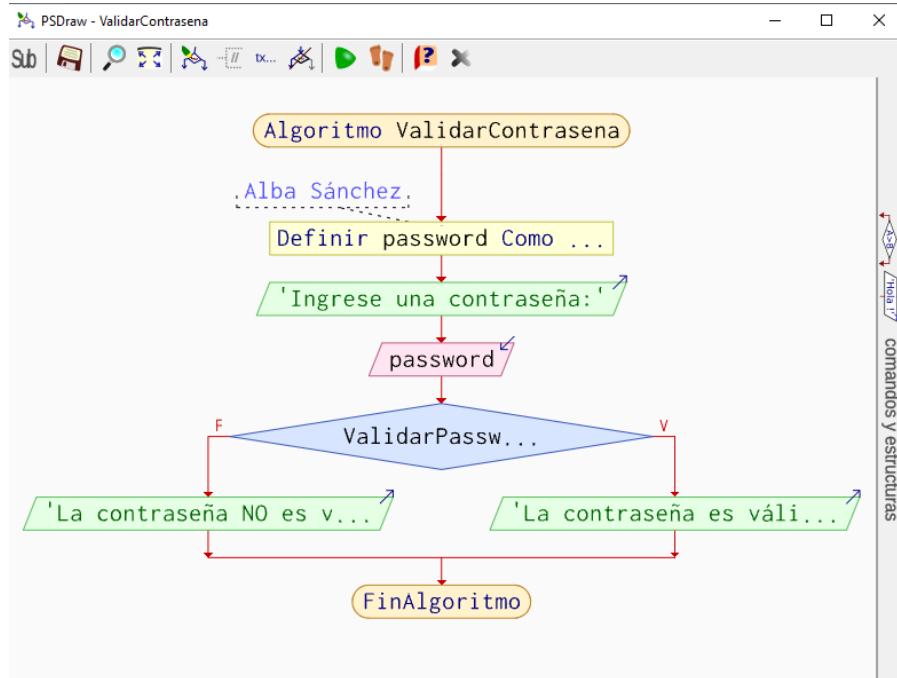
```

Función valida ← ValidaPassword(clave)
    Definir valida Como Lógico
    Si Longitud(clave) > 8 Entonces
        valida ← Verdadero
    Sino
        valida ← Falso
    FinSi
FinFunción
    
```



Alternar entre modo ventana y pantalla completa.

## Algoritmo ValidarContrasena



### 5. Mayor de tres números:

Crea una función 'MayorNumero' que reciba tres números y devuelva el mayor de ellos. Usa condicionales dentro de la función.

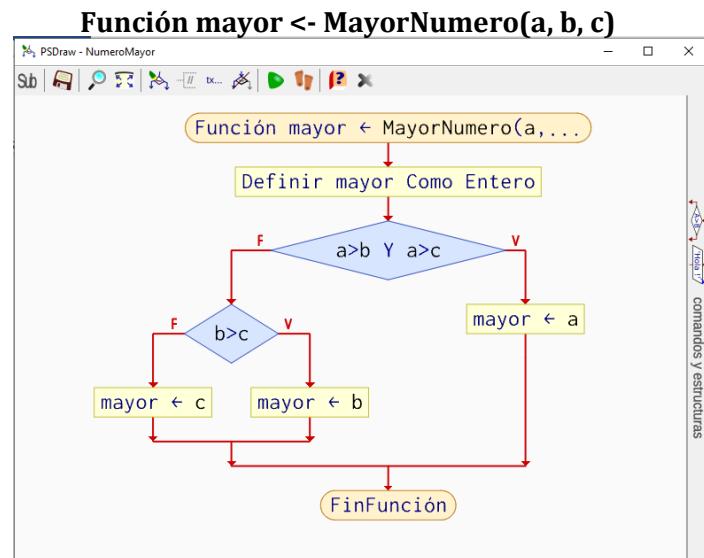
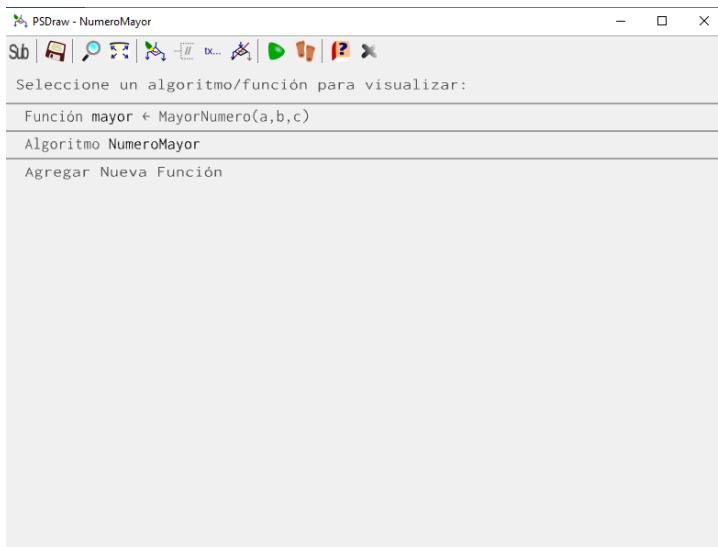
```

1 Función mayor ← MayorNumero(a, b, c)
2   Definir mayor Como Entero
3   Si a > b Y a > c Entonces
4     mayor ← a
5   Sino
6     Si b > c Entonces
7       mayor ← b
8     Sino
9       mayor ← c
10  FinSi
11 FinFunción
12
13
14 Algoritmo MayorNumero //Alba Sánchez
15   Definir n1, n2, n3, resultado Como Entero
16   Escribir "Ingrese el primer número:"
17   Leer n1
18   Escribir "Ingrese el segundo número:"
19   Leer n2
20   Escribir "Ingrese el tercer número:"
21   Leer n3
22   resultado ← MayorNumero(n1, n2, n3)
23   Escribir "El número mayor es: ", resultado
24 FinAlgoritmo
25

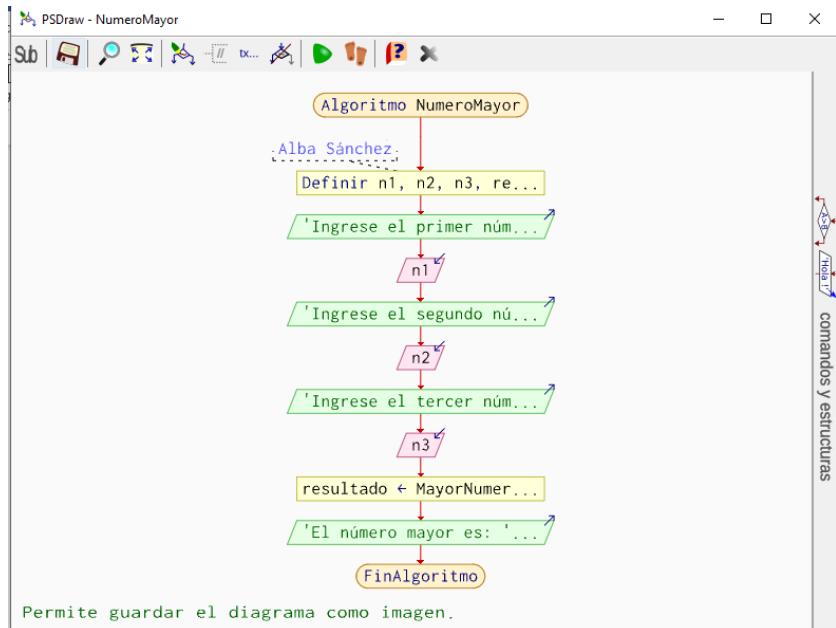
```

La ejecución ha finalizado sin errores.

\*\*\* Ejecución Iniciada. \*\*\*
Ingresar el primer número:
> 20
Ingresar el segundo número:
> 25
Ingresar el tercer número:
> 55
El número mayor es: 55
\*\*\* Ejecución Finalizada. \*\*\*



### Algoritmo NumeroMayor



## Ejercicios de Vectores

### 6. Cargar y mostrar elementos:

Declara un vector de 5 posiciones. Pide al usuario que ingrese 5 números y luego muestra todos los valores almacenados.

The screenshot shows the PSelint interface. On the left, the code is displayed:

```

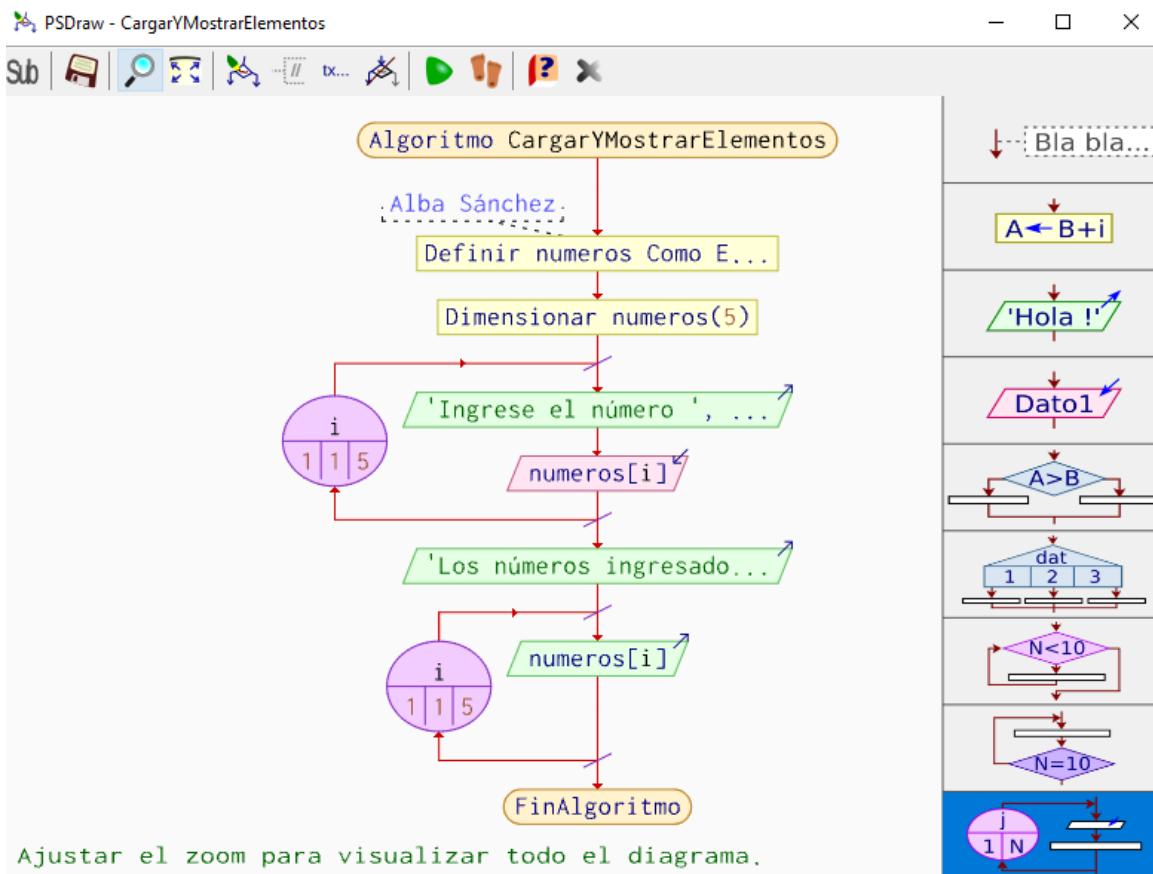
1 Algoritmo CargarYMostrarElementos //Alba Sánchez
2
3 Definir numeros Como Entero
4 Dimensionar numeros[5]
5
6 Para i ← 1 Hasta 5 Con Paso 1 Hacer
7   Escribir "Ingrese el número ", i, ": "
8   Leer numeros[i]
9 FinPara
10
11 Escribir "Los números ingresados son:"
12 Para i ← 1 Hasta 5 Con Paso 1 Hacer
13   Escribir numeros[i]
14 FinPara
15
16 FinAlgoritmo
17

```

In the center, the execution window shows the process: CARGARYMOSTRARELEMENTOS. It prompts for 5 numbers (25, 32, 43, 16, 14) and outputs them (25, 32, 43, 16, 14). The status bar at the bottom says "La ejecución ha finalizado sin errores."

The right panel shows a list of commands with their icons:

- Escribir
- Leer
- Asignar
- Si-Entonces
- Según
- Mientras
- Repetir
- Para
- Función



## 7. Suma de elementos del vector:

Crea un vector de 10 números y calcula la suma total de sus elementos.

The screenshot shows the PSeInt IDE interface. On the left, the code editor displays the pseudocode for the algorithm:

```

1 Algoritmo SumaElementosVector //Alba Sánchez
2
3 Definir numeros, suma Como Entero
4 Dimensionar numeros[10]
5 suma ← 0
6
7 Para i + 1 Hasta 10 Con Paso 1 Hacer
8   Escribir "Ingrese el número ", i, "; "
9   Leer numeros[i]
10  suma ← suma + numeros[i]
11 FinPara
12
13 Escribir "La suma total de los elementos es: ", suma
14
15 FinAlgoritmo
16

```

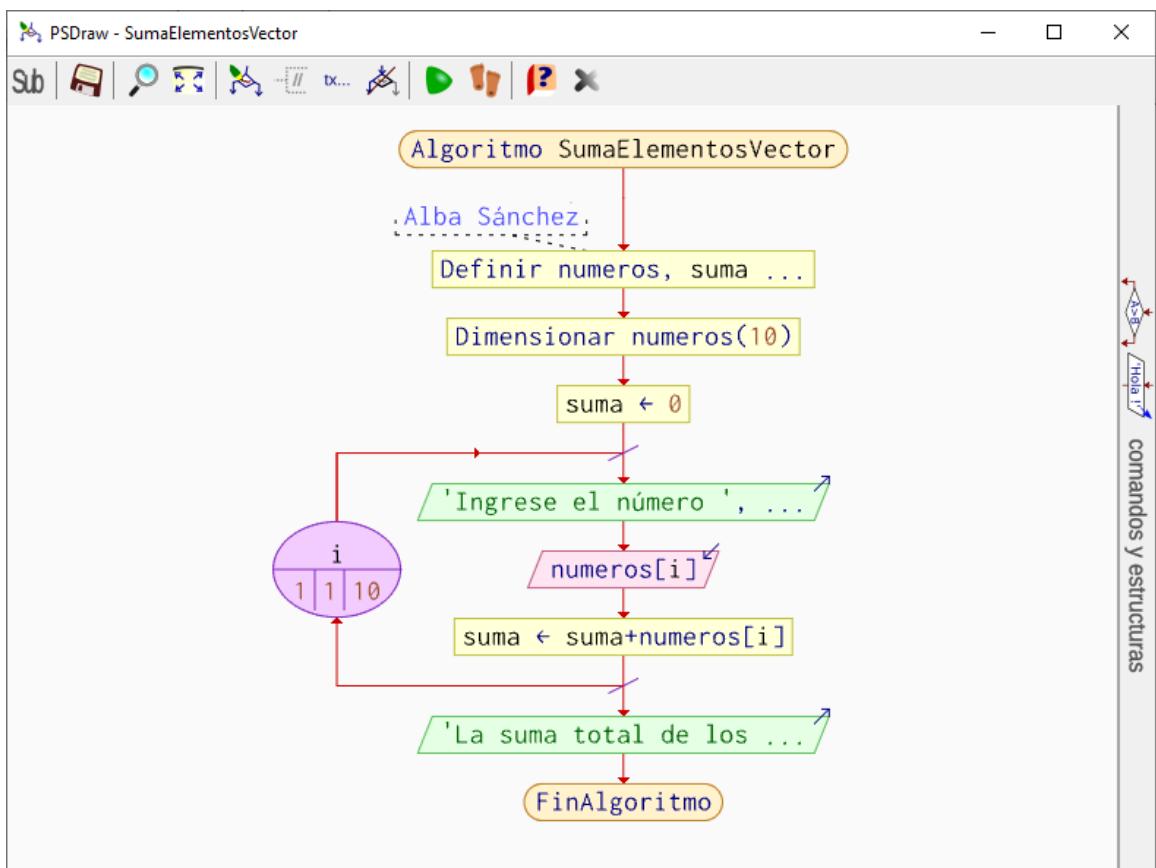
The right panel shows the execution window with the following output:

```

PSeInt - Ejecutando proceso SUMAELEMENTOSVECTOR
*** Ejecución Iniciada. ***
Ingrese el número 1:
> 1
Ingrese el número 2:
> 3
Ingrese el número 3:
> 25
Ingrese el número 4:
> 70
Ingrese el número 5:
> 10
Ingrese el número 6:
> 22
Ingrese el número 7:
> 14
Ingrese el número 8:
> 18
Ingrese el número 9:
> 54
Ingrese el número 10:
> 29
La suma total de los elementos es: 246
*** Ejecución Finalizada. ***

```

At the bottom of the execution window, it says: "La ejecución ha finalizado sin errores."



## 8. Buscar número en el vector:

Pide al usuario llenar un vector con 8 números y luego solicita un número a buscar. Muestra si el número se encuentra o no en el vector.

The screenshot shows the PSelnt IDE interface. On the left, the code editor displays an algorithm named 'Algoritmo BuscarNumeroEnVector' by 'Alba Sánchez'. The code defines a vector 'numeros[8]' and initializes 'encontrado <= 0'. It then loops from 1 to 8, prompting the user to enter a number at index 'i' and reading it into 'numeros[i]'. It asks for a search value 'buscar'. Inside the loop, it checks if 'numeros[i] = buscar'. If true, it sets 'encontrado <= 1' and exits the loop. Otherwise, it increments 'encontrado <= 1'. After the loop, it prints the result based on the value of 'encontrado'. On the right, the 'Comandos' window shows the execution process. It prompts for numbers 1 through 8, then asks for a search value. It finds that the number 35 is not in the vector and concludes the execution.

```

1 Algoritmo BuscarNumeroEnVector //Alba Sánchez
2
3 Definir numeros, buscar, encontrado Como Entero
4 Dimensionar numeros[8]
5 encontrado <= 0
6
7 Para i <= 1 Hasta 8 Con Paso 1 Hacer
8   Escribir "Ingrese el número ", i, ": "
9   Leer numeros[i]
10 FinPara
11
12 Escribir "Ingrese el número que desea buscar: "
13 Leer buscar
14
15 Para i <= 1 Hasta 8 Con Paso 1 Hacer
16   Si numeros[i] = buscar Entonces
17     | encontrado <= 1
18   FinSi
19 FinPara
20
21 Si encontrado = 1 Entonces
22   Escribir "El número ", buscar, " se encuentra en el vector."
23 SiNo
24   Escribir "El número ", buscar, " NO se encuentra en el vector."
25 FinSi
26
27 FinAlgoritmo
28

```

La ejecución ha finalizado sin errores.

## PSelnt - Ejecutando proceso BUSCARNUMEROENVECTOR

\*\*\* Ejecución Iniciada. \*\*\*

Ingrese el número 1:

> 1

Ingrese el número 2:

> 7

Ingrese el número 3:

> 8

Ingrese el número 4:

> 8

Ingrese el número 5:

> 0

Ingrese el número 6:

> 6

Ingrese el número 7:

> 4

Ingrese el número 8:

> 9

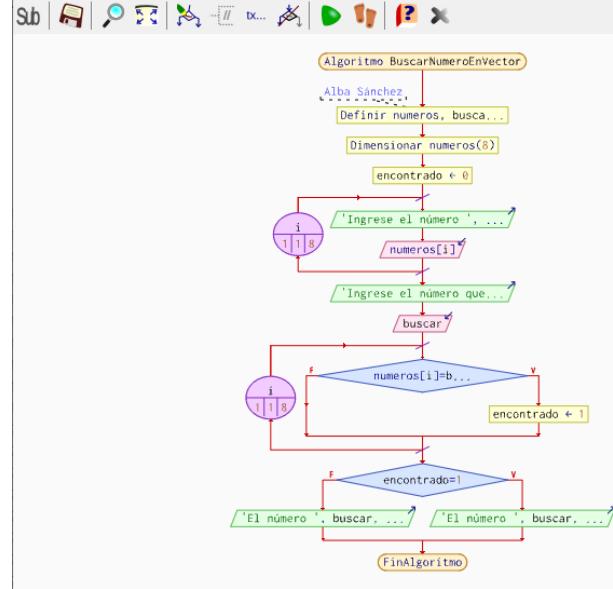
Ingrese el número que desea buscar:

> 6

El número 6 se encuentra en el vector.

\*\*\* Ejecución Finalizada. \*\*\*

## PSDraw - BuscarNumeroEnVector



## 9. Promedio de calificaciones:

Almacena las calificaciones de 5 estudiantes en un vector y calcula el promedio general del grupo.

The screenshot shows the PSelnt IDE interface. On the left, the code for 'PromedioCalificaciones' is displayed:

```

1 Algoritmo PromedioCalificaciones //Alba Sánchez
2
3 Definir calificaciones, suma, promedio Como Real
4 Dimensionar calificaciones[5]
5 suma ← 0
6
7 Para i ← 1 Hasta 5 Con Paso 1 Hacer
8   Escribir "Ingrese la calificación del estudiante ", i, ": "
9   Leer calificaciones[i]
10  suma ← suma + calificaciones[i]
11 FinPara
12
13 promedio ← suma / 5
14
15 Escribir "El promedio general del grupo es: ", promedio
16
17 FinAlgoritmo
18

```

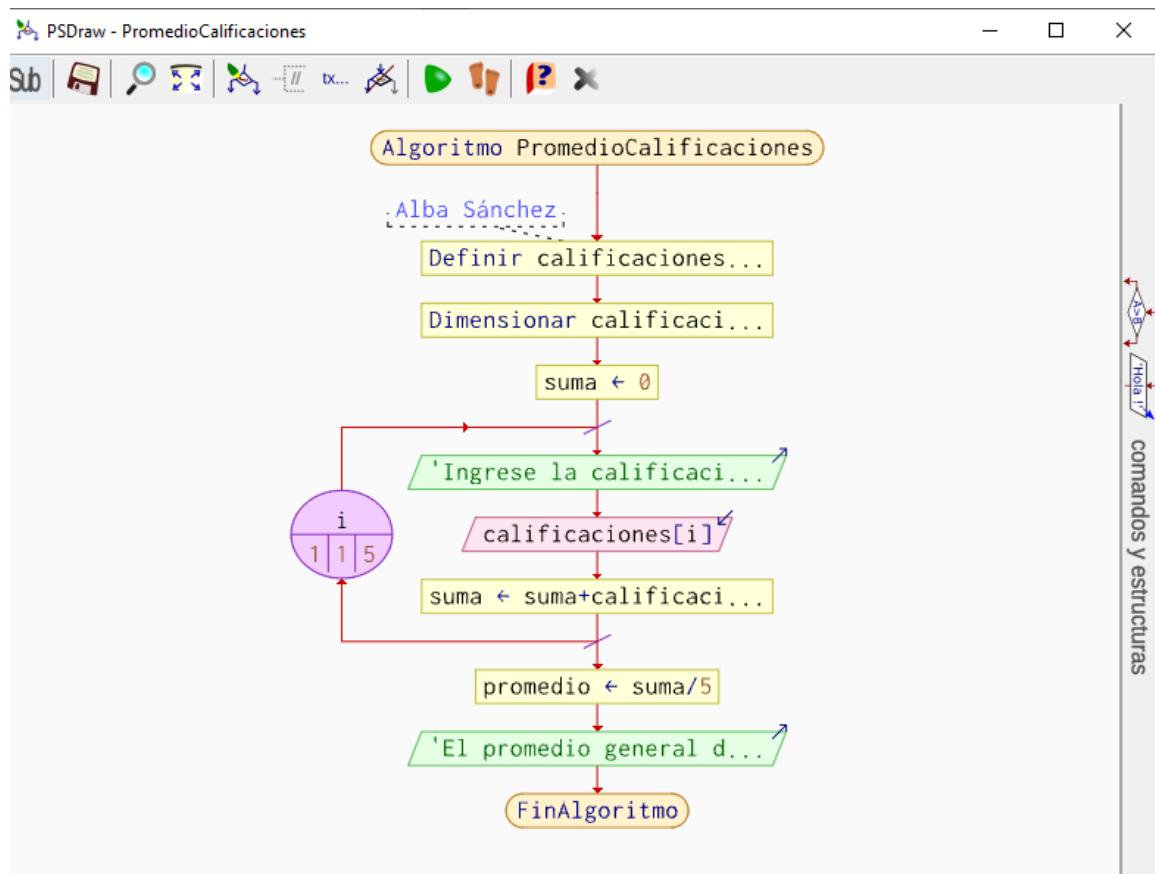
The right side shows the execution window titled 'PSelnt - Ejecutando proceso PROMEDIOCALIFICACIONES'. It displays the following interaction:

```

*** Ejecución Iniciada. ***
Ingrese la calificación del estudiante 1:
> 88
Ingrese la calificación del estudiante 2:
> 97
Ingrese la calificación del estudiante 3:
> 99
Ingrese la calificación del estudiante 4:
> 81
Ingrese la calificación del estudiante 5:
> 90
El promedio general del grupo es: 91
*** Ejecución Finalizada. ***

```

At the bottom, a message says 'La ejecución ha finalizado sin errores.'



## 10. Contar números pares:

Crea un vector con 10 números y usa un contador para determinar cuántos son pares. Muestra el total de números pares encontrados.

The screenshot shows the PSeInt IDE interface. On the left, the code editor displays the pseudocode for 'ContarNumerosPares' by Alba Sánchez. On the right, the 'Comandos' window shows the execution process, prompting for 10 numbers and outputting the result: 'La cantidad de números pares es: 4'. A vertical toolbar on the left includes buttons for 'Lista de Variables' and 'Operadores y Funciones'.

```

1  Algoritmo ContarNumerosPares //Alba Sánchez
2
3  Definir numeros, contador Como Entero
4  Dimensionar numeros[10]
5  contador ← 0
6
7  Para i ← 1 Hasta 10 Con Paso 1 Hacer
8      Escribir "Ingrese el número ", i, ": "
9      Leer numeros[i]
10     Si numeros[i] MOD 2 = 0 Entonces
11         contador ← contador + 1
12     FinSi
13     FinPara
14
15     Escribir "La cantidad de números pares es: ", contador
16
17 FinAlgoritmo
18

```

PSelnt - Ejecutando proceso CONTARNUMEROSPARES  
\*\*\* Ejecución Iniciada. \*\*\*  
Ingresar el número 1:  
> 33  
Ingresar el número 2:  
> 25  
Ingresar el número 3:  
> 28  
Ingresar el número 4:  
> 46  
Ingresar el número 5:  
> 65  
Ingresar el número 6:  
> 87  
Ingresar el número 7:  
> 23  
Ingresar el número 8:  
> 34  
Ingresar el número 9:  
> 67  
Ingresar el número 10:  
> 80  
La cantidad de números pares es: 4  
\*\*\* Ejecución Finalizada. \*\*\*

