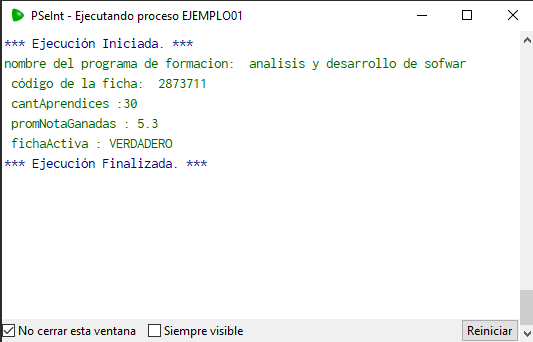


Ejemplos:

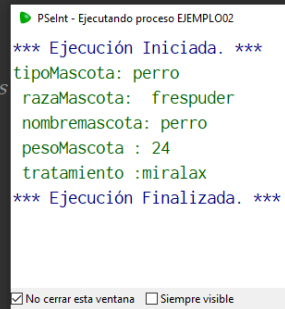
- 1) realizar un Algoritmo que permita guardar el nombre del programa de formaciones código de la ficha, cantidad de aprendices, promedio de notas ganadas y saber si la ficha esta activa en el sistema

```
1 Algoritmo ejemplo01
2   // ejemplo 01
3   // realizar un Algoritmo que permita guardar :
4   // el nombre del programa de formacion,el codigo
5   // de la ficha, cantidad de aprendices,promedio
6   // de notas ganadas y saber si la ficha esta activa
7   // en el sistema
8   Definir nombreProgramaFormacion como cadena
9   Definir codigoficha como cadena
10  Definir cantAprendices Como Entero
11  Definir promNotaGanadas Como Real
12  Definir fichaActivo Como Logico; //verdadero ó falso
13  // OPERACIONES Y ASIGNACIONES
14  nombreProgramaFormacion ← "analisis y desarrollo de sofwar";
15  codigoficha ← " 2873711"
16  cantAprendices ← 30;
17  promNotaGanadas ← 5.3;
18  fichaActiva ← Verdadero
19  // salida
20  Escribir "nombre del programa de formacion: ",nombreProgramaFormacion;
21  Escribir " código de la ficha: ",codigoficha;
22  Escribir " cantAprendices :",cantAprendices
23  Escribir " promNotaGanadas : " , promNotaGanadas;
24  Escribir " fichaActiva : " , fichaActiva
25
```



- 2) se requiere un Algoritmo para una veterinaria que permita tener la variable para el tipo de mascota nombre de la mascota, peso y tratamiento determinado por el veterinario. asignar las diferentes variables y Mostrar todos los datos

```
1 Algoritmo ejemplo02
2 // se requiere un Algoritmo para una veterinaria
3 // que permita tener la variable para el tipo de mascota
4 // nombre de la mascota, peso y tratamiento determinado
5 // por el veterinario. asignar las diferentes variables
6 // Mostrar todos los datos
7 Definir tipoMascota como cadena;
8 Definir razaMascota como cadena;
9 Definir nombreMascota como cadena;
10 Definir pesoMascota Como Real;
11 Definir tratamiento como cadena;
12
13 tipoMascota ← "perro";
14 razaMascota ← "frespuder"
15 nombreMascota ← "matilda"
16 pesoMascota ← 24;
17 tratamiento ← "miralax"
18 //salidas
19 Escribir "tipoMascota: ",tipoMascota;
20 Escribir " razaMascota: ",razaMascota;
21 Escribir " nombreMascota: ",tipoMascota;
22 Escribir " pesoMascota : ",pesoMascota;
23 Escribir " tratamiento : " , tratamiento
```

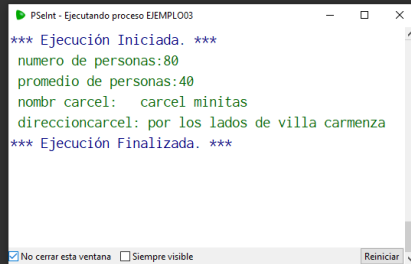


```
*** Ejecución Iniciada. ***
tipoMascota: perro
razaMascota: frespuder
nombreMascota: perro
pesoMascota : 24
tratamiento :miralax
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

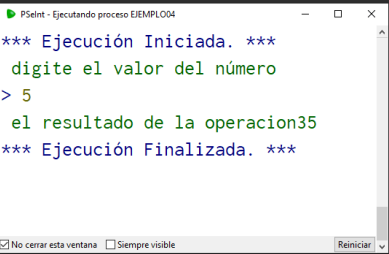
- 3) en una cárcel se requiere un Algoritmo que permita identificar cuantas personas privadas de la libertad tiene la cárcel, el promedio de años de condena del ppl nombre de la cárcel y la dirección de la cárcel de la cárcel mostrando datos

```
1 Algoritmo ejemplo03
2 //en una carcel se requiere un Algoritmo que permita identificar cuantas personas privadas
3 // de la libertad tiene la carcél, el promedio de años de condena
4 //de las ppl,el nombre de la carcel y la dirección de la carcel de la carcel
5 // mostrando datos
6
7 Definir numperson Como Entero;
8 Definir promA Como Real;
9 Definir nombrCarcel como cadena
10 Definir dirreccioncarcel como cadena
11 // procesos y asignaciones
12
13 numperson ← 80;
14 promA ← 40;
15 nombrCarcel ← " carcel minitas"
16 dirreccioncarcel ← " por los lados de villa carmenza"
17 // salida
18 Escribir " numero de personas:", numperson;
19 Escribir " promedio de personas:", promA;
20 Escribir " nombr carcel: " , nombrCarcel;
21 Escribir " direccioncarcel:" , dirreccioncarcel;
22
23
24
25 FinAlgoritmo
26
```



- 4) se requiere un algoritmo que permita solicitar un número al usuario y calcular la siguiente formula: $x \leftarrow 2A + A * 5$

```
1 Algoritmo ejemplo04
2 // se requiere un algoritmo que permita solicitar un número al usuario y calcular la siguiente formula :  $x \leftarrow 2A + A * 5$ 
3 // Declaracion/Definicion variables
4 Definir x, A Como Real
5 // entrada datos
6 Escribir " digite el valor del número";
7 Leer A ;
8
9 // PROCESO - Operaciones - formulas
10  $x \leftarrow 2 * A + A * 5$ ;
11 // SALIDA DATOS
12 Escribir " el resultado de la operacion",x;
13
14 FinAlgoritmo
```



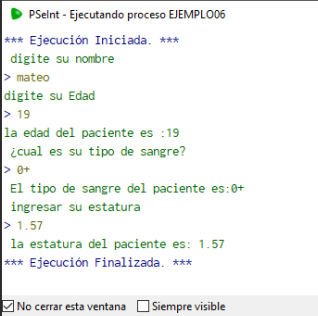
- 5) se requiere un algoritmo que pida 2 números y muestre la suma de ambos

```
1 Algoritmo EJEMPLO05
2 // se requiere un algoritmo que pida 2 numeros y muestre la suma de ambos
3
4 // definicion/ declaracion de variables
5 Definir num1, num2, resultado Como Real
6 // entrada de datos
7 Escribir "digite valor del número 1";
8 Leer num1 ;
9 Escribir " digite el valor del número 2 ";
10 leer num2;
11
12
13 // Proceso - operaciones - formulas
14 resultado  $\leftarrow$  num1 + num2
15 // salida de datos
16 Escribir " el resultado de seuma " , num1,"+ " , num2," es: " , resultado;
17
18 FinAlgoritmo
```



- 6) en un hospital quiere un Algoritmo que solicite datos: nombre del paciente, edad, tipo de sangre estatura y género. mostrar todos los datos al Final del algoritmo

```
1 Algoritmo ejemplo06
2 // en un hospital quiere un Algoritmo que solicite datos: nombre del paciente, edad ,tipo de sangre
3 // estatura y genero . mostrar todos los datos al FinAlgoritmo
4
5 // definir variables
6 Definir nombre, tipoDesangre, genero como cadena;
7 Definir Edad Como Entero;
8 Definir estatura Como Real;
9 // entrada de datos
10 Escribir " digite su nombre"
11 Leer nombre;
12
13 Escribir "digite su Edad";
14 Leer Edad;
15 Escribir "la edad del paciente es : " Edad;
16
17 Escribir " ¿cual es su tipo de sangre?"
18 Leer tipoDesangre
19 Escribir " El tipo de sangre del paciente es:" tipoDesangre;
20
21 Escribir " ingresar su estatura " ;
22 Leer estatura;
23 Escribir " la estatura del paciente es: " estatura;
24
```



8) condicional simple 1

```
1 Algoritmo EJEMPL008
2 //CONDICIONAL SIMPLE 1
3 // DEFINICION VARIABLE
4 Definir estadoClima como cadena;
5 //entrada
6 Escribir "ingrese el estado de clima";
7 Leer estadoClima
8 // Proceso - salida
9 si (estadoClima = "lluvias")Entonces
10     Escribir "sacar sombrilla";
11 FinSi
12 si((estadoClima="lluvias")o(estadoClima="llovioso"))Entonces
13     Escribir "colocar impermeable"
14
15
16 FinSi
17 Escribir " caminar por la calle";
18
19 FinAlgoritmo
20
```

PSeInt - Ejecutando proceso EJEMPL008

```
*** Ejecución Iniciada. ***
ingrese el estado de clima
> lluvias
sacar sombrilla
colocar impermeable
caminar por la calle
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

```
1 Algoritmo EJEMPL008
2 //CONDICIONAL SIMPLE 1
3 // DEFINICION VARIABLE
4 Definir estadoClima como cadena;
5 //entrada
6 Escribir "ingrese el estado de clima";
7 Leer estadoClima
8 // Proceso - salida
9 si (estadoClima = "lluvias")Entonces
10     Escribir "sacar sombrilla";
11 FinSi
12 si((estadoClima="lluvias")o(estadoClima="llovioso"))Entonces
13     Escribir "colocar impermeable"
14
15
16 FinSi
17 Escribir " caminar por la calle";
18
19 FinAlgoritmo
20
```

PSeInt - Ejecutando proceso EJEMPL008

```
*** Ejecución Iniciada. ***
ingrese el estado de clima
> llovioso
colocar impermeable
caminar por la calle
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

9) condicional simple 2

```
1 Algoritmo EJEMPL009
2 // CONDICIONAL SIMPLE 2
3 // DEFINICION/DEFINICION VBLES
4 Definir estrato Como Entero
5 //entrada
6 Escribir " digite su estrato";
7 Leer estrato
8 //Procesos - salida
9 Escribir "bienvenido al sena regional caldas";
10 si (estrato < 3 ) Entonces
11     Escribir " puede aplicar para apoyos de sostenimiento";
12 FinSi
13
14 FinAlgoritmo
15
```

PSeInt - Ejecutando proceso EJEMPL009

```
*** Ejecución Iniciada. ***
digite su estrato
> 3
bienvenido al sena regional caldas
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

```
1 Algoritmo EJEMPL009
2 // CONDICIONAL SIMPLE 2
3 // DEFINICION/DEFINICION VBLES
4 Definir estrato Como Entero
5 //entrada
6 Escribir " digite su estrato";
7 Leer estrato
8 //Procesos - salida
9 Escribir "bienvenido al sena regional caldas";
10 si (estrato < 3 ) Entonces
11     Escribir " puede aplicar para apoyos de sostenimiento";
12 FinSi
13
14 FinAlgoritmo
15
```

PSeInt - Ejecutando proceso EJEMPL009

```
*** Ejecución Iniciada. ***
digite su estrato
> 2
bienvenido al sena regional caldas
puede aplicar para apoyos de sostenimiento
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

10) Condicional DOBLE

```
Algoritmo ejemplo10
// CONDICIONAL DOBLE
// DEFINICION/DECLARACION VBLES
Definir estrato Como Entero
//entrada
Escribir " digite su estrato";
Leer estrato
//Procesos - salida
Escribir "bienvenido al sena regional caldas";
si (estrato ≤ 2 ) Entonces
    Escribir " puede aplicar para apoyos de sostenimiento";
SiNo
    Escribir " no puede acceder a la convocatoria";
FinSi
FinAlgoritmo
```

PSeInt - Ejecutando proceso EJEMPLO10

```
*** Ejecución Iniciada. ***
digite su estrato
> 3
bienvenido al sena regional caldas
no puede acceder a la convocatoria
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

```
<sin_titulo>* <sin_titulo>* EJEMPLO08.psc <sin_titulo>* <sin_titulo>* X
1 Algoritmo ejemplo10
2 // CONDICIONAL DOBLE
3 // DEFINICION/DECLARACION VBLES
4 Definir estrato Como Entero
5 //entrada
6 Escribir " digite su estrato";
7 Leer estrato
8 //Procesos - salida
9 Escribir "bienvenido al sena regional caldas";
10 si (estrato ≤ 2 ) Entonces
11     Escribir " puede aplicar para apoyos de sostenimiento";
12 SiNo
13     Escribir " no puede acceder a la convocatoria";
14 FinSi
15 FinAlgoritmo
16
```

PSeInt - Ejecutando proceso EJEMPLO10

```
*** Ejecución Iniciada. ***
digite su estrato
> 1
bienvenido al sena regional caldas
puede aplicar para apoyos de sostenimiento
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

11) condicional doble 2

```
1 Algoritmo EJEMPLO11
2 // CONDICIONAL DOBLE 2
3 // SE TIENE EL RESULTADO DE UN EXAMEN PARA UN
4 // UN APRENDIZ (SE DEBE PEDIR LA NOTA)
5 // SI LA NOTA ESTA POR DEBAJO 3, DECIR QUE LA
6 // PERDIO DE LO CONTRARIO MOSTRAR QUE SI LA GANO
7 // LA NOTA
8
9 //DEFINICION/DECLARACION VBLES
10 Definir NOTA Como Entero
11 //ENTRADA
12 Escribir "digite una nota entre 0 y 5";
13 Leer nota;
14 //PROCESOS - SALIDAS
15 si( (nota<0) o (nota>5))Entonces
16     Escribir "la nota esta en rango incorrecto";
17 SiNo
18     si (nota <3)Entonces
19         Escribir "usted desaprobo el examen";
20     SiNo
21         Escribir "felicitaciones gano el examen";
22     FinSi
23 FinSi
24 FinAlgoritmo
```

PSeInt - Ejecutando proceso EJEMPLO11

```
*** Ejecución Iniciada. ***
digite una nota entre 0 y 5
> 5
felicitaciones gano el examen
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

```
1 Algoritmo EJEMPLO11
2 // CONDICIONAL DOBLE 2
3 // SE TIENE EL RESULTADO DE UN EXAMEN PARA UN
4 // UN APRENDIZ (SE DEBE PEDIR LA NOTA)
5 // SI LA NOTA ESTA POR DEBAJO 3, DECIR QUE LA
6 // PERDIO DE LO CONTRARIO MOSTRAR QUE SI LA GANO
7 // LA NOTA
8
9 //DEFINICION/DECLARACION VBLES
10 Definir NOTA Como Entero
11 //ENTRADA
12 Escribir "digite una nota entre 0 y 5";
13 Leer nota;
14 //PROCESOS - SALIDAS
15 si( (nota<0) o (nota>5))Entonces
16     Escribir "la nota esta en rango incorrecto";
17 SiNo
18     si (nota <3)Entonces
19         Escribir "usted desaprobo el examen";
20     SiNo
21         Escribir "felicitaciones gano el examen";
22     FinSi
23 FinSi
24 FinAlgoritmo
```

PSeInt - Ejecutando proceso EJEMPLO11

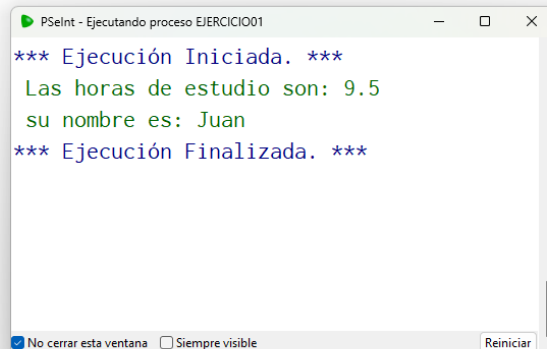
```
*** Ejecución Iniciada. ***
digite una nota entre 0 y 5
> -6
la nota esta en rango incorrecto
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

Ejercicios:

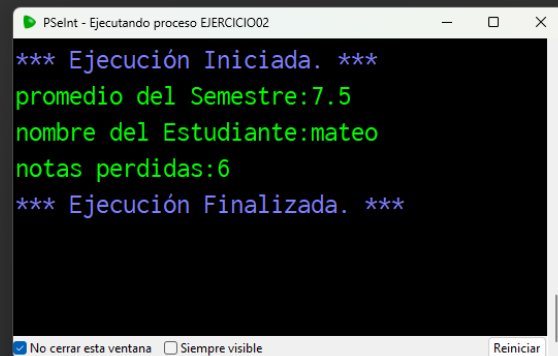
- 1) Hacer un algoritmo que declare una variable para guardar el número de horas de estudio, y otra para guardar el nombre. Escribir ambos datos

```
1 Algoritmo ejercicio01
2 // Hacer un algoritmo que declare una variable para guardar el número de horas de estudio,
3 //y otra para guardar el nombre. Escribir ambos datos.
4 Definir horasDEestudio Como Real;
5 Definir nombre como cadena;
6
7 horasDEestudio ← 9.5;
8 nombre ← "Juan";
9
10 Escribir " Las horas de estudio son: ",horasDEestudio;
11 Escribir " su nombre es: ",nombre;
12
13
14
15
16 FinAlgoritmo
17
```



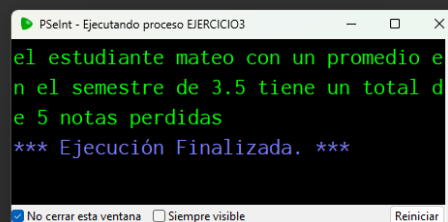
2. Hacer un algoritmo que lea dos números enteros A y B y muestre el doble de su suma

```
1 Algoritmo ejercicio02
2 // Hacer un algoritmo que lea dos números enteros A y B y muestre el doble de su suma
3
4 // Definir variables i
5
6 Definir promedioSemestre Como Real;
7 Definir nombreEstudiante como cadena;
8 Definir notasperdidas Como Entero;
9
10 //Proceso-operaciones y fórmulas
11 promedioSemestre ← 7.5;
12 nombreEstudiante ← "mateo";
13 notasperdidas ← 6;
14
15 //salidas
16 Escribir "promedio del Semestre:",promedioSemestre;
17 Escribir "nombre del Estudiante:" , nombreEstudiante;
18 Escribir "notas perdidas:" , notasperdidas
19
20 FinAlgoritmo
21
```



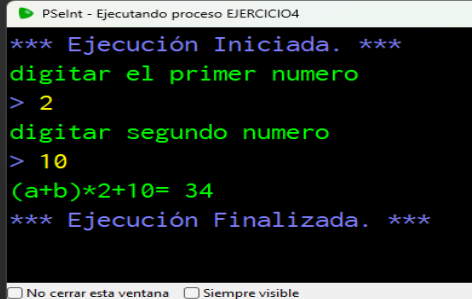
3. Hacer un algoritmo que declare una variable para guardar el promedio del semestre, otra para guardar el nombre de un estudiante y otra para guardar el número de notas perdidas

```
1 Algoritmo EJERCICIO3
2 // Hacer un algoritmo que declare una variable para guardar el promedio del semestre,
3 //otra para guardar el nombre de un estudiante y otra para guardar el número de notas perdidas
4
5 // ENTRDA DE DATOS
6
7 definir prom_semestre Como Real
8 Definir nombre como cadena
9 definir num_notas_perdidas Como Entero
10
11 //PROCESO
12
13 prom_semestre ← 3.5
14 nombre ← "mateo"
15 num_notas_perdidas ← 5
16
17 //SALIDA DE DATOS
18
19 Escribir "el estudiante " ,nombre " con un promedio en el semestre de " ,prom_semestre " tiene un total de " ,num_notas_perdidas " notas perdidas"
20
21 FinAlgoritmo
```



4) Hacer un algoritmo que lea dos números enteros A y B y muestre el resultado de realizar: $(A + B) * 2 + 10$

```
1 Algoritmo EJERCICIO4
2 // Hacer un algoritmo que lea dos números enteros A y B y
3 //muestre el resultado de realizar: (A + B) *2 +10
4
5 // DECLARAR VARIABLES
6
7 definir a, b, resultado Como Entero
8
9 // ENTRADA DE DATOS
10
11 Escribir "digitar el primer numero"
12 leer a
13 Escribir "digitar segundo numero"
14 Leer b
15
16 // PROCESO
17
18 resultado ← (a+b)*2+10
19
20 // SALIDA DE DATOS
21
22 Escribir "(a+b)*2+10= " , resultado
23
24 FinAlgoritmo
```



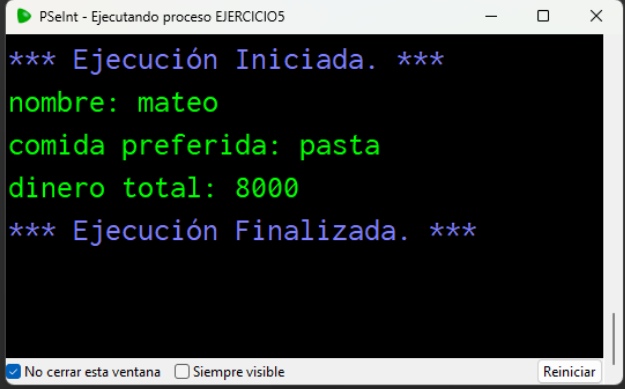
PSeInt - Ejecutando proceso EJERCICIO4

```
*** Ejecución Iniciada. ***
digitar el primer numero
> 2
digitar segundo numero
> 10
(a+b)*2+10= 34
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible

5) Hacer un algoritmo que declare una variable para guardar el nombre de una persona, otra para guardar la comida preferida y otra para guardar la cantidad de dinero que posee

```
1 Algoritmo EJERCICIO5
2
3 // ENTRADA DE DATOS
4
5 DEFINIR nombre, comida como cadena
6 definir cant_dinero como real
7
8 // PROCESO
9
10 nombre ← "mateo"
11 comida ← "pasta"
12 cant_dinero ← 8000
13
14 // SALIDA DE DATOS
15
16 Escribir "nombre: " ,nombre
17 Escribir "comida preferida: " ,comida
18 escribir "dinero total: " ,cant_dinero
19
20
21
22
23 FinAlgoritmo
```



PSeInt - Ejecutando proceso EJERCICIO5

```
*** Ejecución Iniciada. ***
nombre: mateo
comida preferida: pasta
dinero total: 8000
*** Ejecución Finalizada. ***
```

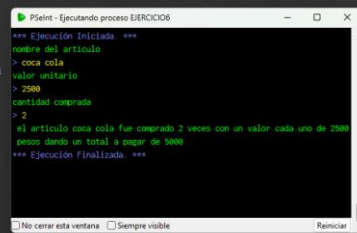
☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

6) Hacer un algoritmo que lea el nombre de un artículo, el valor unitario, la cantidad a comprar y muestre el nombre y el total a pagar

```

1 Algoritmo EJERCICIO6
2 // Hacer un algoritmo que lea el nombre de un artículo, el valor unitario, la cantidad a comprar y muestre
3 // el nombre y el total a pagar
4
5 //DECLARACION DE VARIABLES
6
7 definir nombre_articulo como cadena
8 definir valor_unit, total_pagar Como real
9 Definir cant_comprar como entero
10
11 // ENTRADA DE DATOS
12
13 Escribir "nombre del artículo"
14 Leer nombre_articulo
15 Escribir "valor unitario"
16 Leer valor_unit
17 Escribir "cantidad comprada"
18 Leer cant_comprar
19
20 //PROCESO
21
22 total_pagar = valor_unit * cant_comprar
23
24 //SALIDA DE DATOS
25
26 Escribir "el artículo " , nombre_articulo " fue comprado " , cant_comprar " veces con un valor cada uno de " , valor_unit " pesos dando un total a pagar de " , total_pagar ,

```

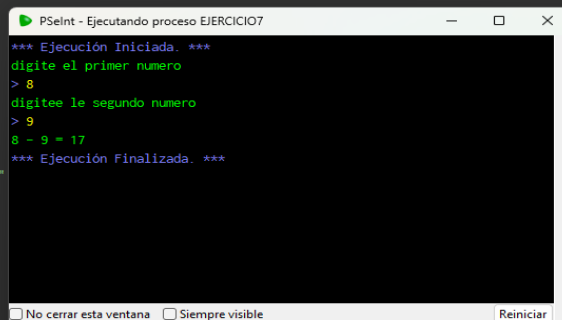


7) Hacer un algoritmo para sumar dos números, los cuales serán tecleados por el usuario
Mostrar el resultado

```

1 Algoritmo EJERCICIO7
2 // Hacer un algoritmo para sumar dos números, los cuales serán tecleados por el usuario Mostrar el resultado
3
4 // DECLARACION DE VARIABLES
5
6 Definir a, b, c Como Entero
7
8 // ENTRADA DE DATOS
9
10 Escribir "digite el primer numero"
11 Leer a
12 Escribir "digitee le segundo numero"
13 Leer b
14
15 //PROCESO
16
17 C ← a + b
18
19 // SALIDA DE DATOS
20
21 Escribir a, " + ", b, " = ", c
22
23 FinAlgoritmo

```

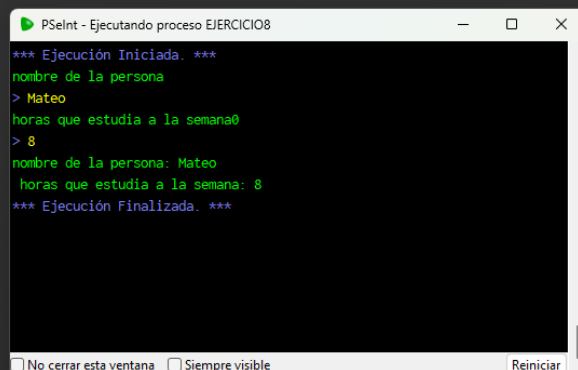


8) Hacer un algoritmo que lea el nombre de una persona y número de horas que estudia en la semana

```

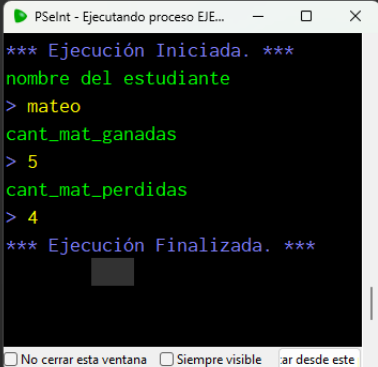
1 Algoritmo EJERCICIO8
2 // Hacer un algoritmo que lea el nombre de una persona y número de horas que estudia en la semana
3
4 // DECLARACION DE VARIABLES
5
6 definir nombre como cadena
7 definir horas Como Entero
8
9 // ENTRADA DE DATOS
10
11 Escribir "nombre de la persona" , nombre
12 Leer nombre
13 escribir "horas que estudia a la semana" , horas
14 Leer horas
15
16 // SALIDA DE DATOS
17
18 Escribir "nombre de la persona: " , nombre
19 escribir " horas que estudia a la semana: " , horas
20
21 FinAlgoritmo

```



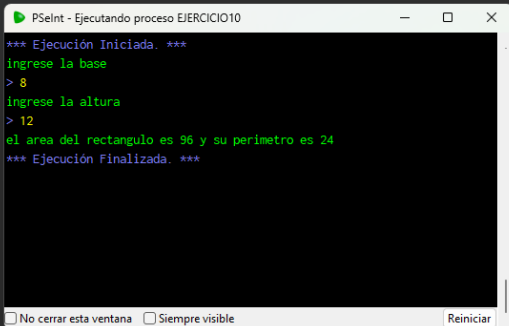
- 9) Hacer un algoritmo que lea el nombre de un estudiante, la cantidad de materias perdidas y la cantidad de materias ganadas

```
1 Algoritmo EJERCICIO9
2 // Hacer un algoritmo que lea el nombre de un estudiante, la cantidad de materias perdidas y
3 // la cantidad de materias ganadas
4
5 // DECLARACION DE VARIABLES
6
7 definir nombre como cadena
8 definir cant_mat_perdidas, cant_mat_ganadas como entero
9
10 // ENTRADA DE DATOS
11
12 Escribir "nombre del estudiante"
13 Leer nombre
14 Escribir "cant_mat_ganadas"
15 Leer cant_mat_ganadas
16 Escribir "cant_mat_perdidas"
17 Leer cant_mat_perdidas
18
19 FinAlgoritmo
20
```



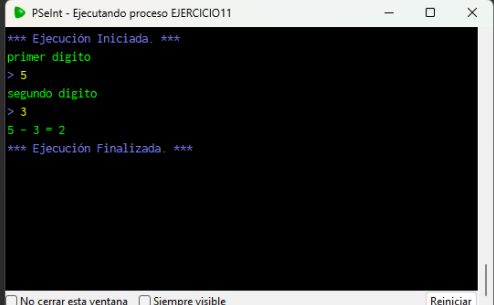
- 10) Hacer un algoritmo que lea el alto y el ancho de un rectángulo y muestre su área y su perímetro

```
1 Algoritmo EJERCICIO10
2 // Hacer un algoritmo que lea el alto y el ancho de un rectángulo y muestre su área y su perímetro.
3
4 // DECLARACION DE VARIABLES
5
6 definir b, h, area, peri como real
7
8 // ENTRADA DE DATOS
9
10 Escribir "ingrese la base"
11 leer b
12 Escribir "ingrese la altura"
13 leer h
14
15 // PROCESO
16
17 area ← b*h
18 peri ← (a*2)+h*2
19
20 // SALIDA DE DATOS
21
22 Escribir "el area del rectangulo es " ,area " y su perimetro es " ,peri
23
24 FinAlgoritmo
```



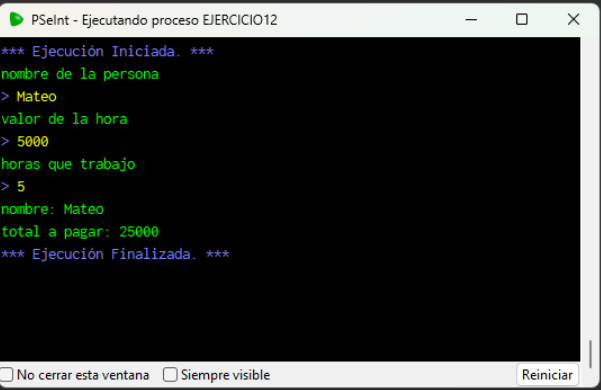
- 11) Hacer un algoritmo que lea dos números enteros A y B y muestre su diferencia

```
1 Algoritmo EJERCICIO11
2 // Hacer un algoritmo que lea dos números enteros A y B y muestre su diferencia
3
4 // DECLARAR VARIABLES
5
6 Definir a, b, c Como Entero
7
8 // ENTRADA DE DATOS
9
10 escribir "primer digito"
11 leer a
12 Escribir "segundo digito"
13 leer b
14
15 // PROCESO
16
17 c ← a-b
18
19 // SALIDA DE DATOS
20
21 Escribir a " - " ,b " = " , c
22
23 FinAlgoritmo
```



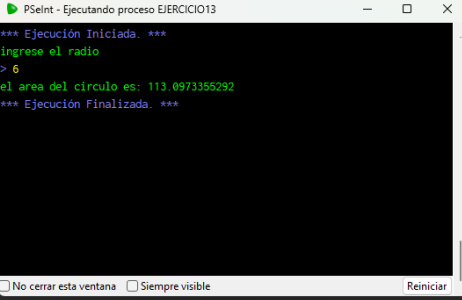
- 12) Hacer un algoritmo que lea el nombre de una persona, el valor de la hora trabajada y el número de horas que trabajó, Se debe mostrar el nombre y el pago de la persona

```
1  Algoritmo EJERCICIO12
2  // Hacer un algoritmo que lea el nombre de una persona, el valor de la hora trabajada y
3  // el número de horas que trabajó. Se debe mostrar el nombre y el pago de la persona
4
5  // DECLARACION DE VARIABLES
6
7  Definir nombre como cadena
8  Definir valorHora como real
9  definir horaTrabajada Como Entero
10 DEFINIR pago como real
11
12 // ENTRADA DE DATOS
13
14 Escribir "nombre de la persona"
15 leer nombre
16 Escribir "valor de la hora"
17 leer valorHora
18 Escribir "horas que trabajo"
19 leer horaTrabajada
20
21 // PROCESO
22
23 pago ← horaTrabajada*valorHora
24
25 // SALIDA DE DATOS
26
27 Escribir "nombre: " , nombre
28 Escribir "total a pagar: " , pago
29
30
31
32 FinAlgoritmo
```



- 13) Pedir el radio de un círculo y calcular su área. $A=PI*r^2$

```
1  Algoritmo EJERCICIO13
2  // Pedir el radio de un círculo y calcular su área.  $A=PI*r^2$ 
3
4  // DECLARACION DE VARIABLES
5
6  Definir r, A como real
7
8  // ENTRADA DE DATOS
9
10 Escribir "ingrese el radio"
11 Leer r
12
13 // PROCESO
14
15  $A \leftarrow PI*r^2$ 
16
17 // SALIDA DE DATOS
18
19 Escribir "el area del círculo es: " , A
20
21
22
23 FinAlgoritmo
```



14) Pedir el radio de una circunferencia y calcular su longitud

```
sin_titulo> EJERCICIO-14.psc x
1 Algoritmo EJERCICIO14
2 // Pedir el radio de una circunferencia y calcular su longitud
3
4 // DECLARACION DE VARIABLES
5
6 Definir r, L como real
7
8 // ENTRADA DE DATOS
9
10 Escribir "ingrese el radio"
11 Leer r
12
13 // PROCESO
14
15  $L \leftarrow r \cdot 2 \cdot \pi$ 
16
17 // SALIDA DE DATOS
18
19 Escribir "la longitud es: " ,L
20 Leer l
21
22
23 FinAlgoritmo
24
```

PSeInt - Ejecutando proceso EJERCICIO14

```
*** Ejecución Iniciada. ***
ingrese el radio
> 43
la longitud es: 270.1769682087
>
```

línea 20 instrucción 1

15) Pedir el lado de un cuadrado, mostrar su área y su perímetro

```
1 Algoritmo EJERCICIO15
2 // . Pedir el lado de un cuadrado, mostrar su área y su perímetro
3
4 //DECLARACION DE VARIABLES
5
6 definir lado, area, perimetro como real
7
8 // ENTRADA DE DATOS
9
10 Escribir "ingrese el lado del cuadrado"
11 leer lado
12
13 // PROCESO
14
15 area  $\leftarrow$  lado*lado
16 perimetro  $\leftarrow$  lado*4
17
18 // SALIDA DE DATOS
19
20 Escribir "area del cuadrado: " ,area
21 Escribir "perimetro del cuadrado: " ,perimetro
22
23
24 FinAlgoritmo
```

PSeInt - Ejecutando proceso EJERCICIO15

```
*** Ejecución Iniciada. ***
ingrese el lado del cuadrado
> 8
area del cuadrado: 64
perimetro del cuadrado: 32
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible Reiniciar

16) Calcular el área de un rectángulo de lados X e Y

```
1 Algoritmo EJERCICIO16
2   // Calcular el área de un rectángulo de lados X e Y
3
4   // DECLARACION DE VARIABLES
5
6   definir X, Z, area como real
7
8   // ENTRADA DE DATOS
9
10  Escribir "ingrese la base"
11  leer X
12  Escribir "ingrese la altura"
13  leer Z
14
15  // PROCESO
16
17  area ← X*Z
18
19  // SALIDA DE DATOS
20
21  Escribir "el area del rectangulo es " ,area
22
23
24
25 FinAlgoritmo
```

PSeInt - Ejecutando proceso EJERCICIO16

```
*** Ejecución Iniciada. ***
ingrese la base
> 4
ingrese la altura
> 9
el area del rectangulo es 36
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible Reiniciar

17) Pedir dos números y decir si son iguales o no

```
<sin_titulo>* <sin_titulo>* EJEMPLO08.psc EJEMPLO09.psc ejemplo10.psc EJEMPLO11.psc <sin_titulo>* X
```

```
1 Algoritmo EJERCIO17
2   // Pedir dos números y decir si son iguales o no
3
4   // DECLARACION/DEFINICION
5   Definir num1 , num2 Como Real;
6   //entrada
7   Escribir "digite el numero 1 y el numero 2";
8   Leer num1 , num2;
9   //Proceso -salida
10  si(num1 = num2) Entonces
11      Escribir "los numeros son iguales"
12  SiNo
13      Escribir "los numeros son diferentes"
14  FinSi
15
16
17
18 FinAlgoritmo
```

PSeInt - Ejecutando proceso EJERCIO17

```
*** Ejecución Iniciada. ***
digite el numero 1 y el numero 2
> 3
> 3
los numeros son iguales
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

```
<sin_titulo>* <sin_titulo>* EJEMPLO08.psc EJEMPLO09.psc ejemplo10.psc EJEMPLO11.psc <sin_titulo>* X
```

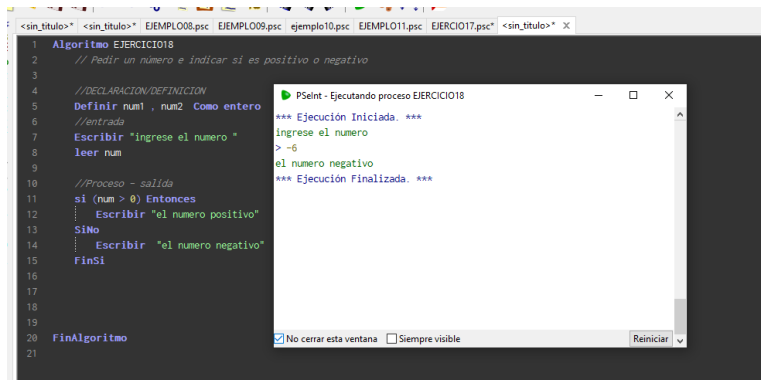
```
1 Algoritmo EJERCIO17
2   // Pedir dos números y decir si son iguales o
3
4   // DECLARACION/DEFINICION
5   Definir num1 , num2 Como Real;
6   //entrada
7   Escribir "digite el numero 1 y el numero 2";
8   Leer num1 , num2;
9   //Proceso -salida
10  si(num1 = num2) Entonces
11      Escribir "los numeros son iguales"
12  SiNo
13      Escribir "los numeros son diferentes"
14  FinSi
15
16
17
18 FinAlgoritmo
```

PSeInt - Ejecutando proceso EJERCIO17

```
*** Ejecución Iniciada. ***
digite el numero 1 y el numero 2
> 2
> 1
los numeros son diferentes
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

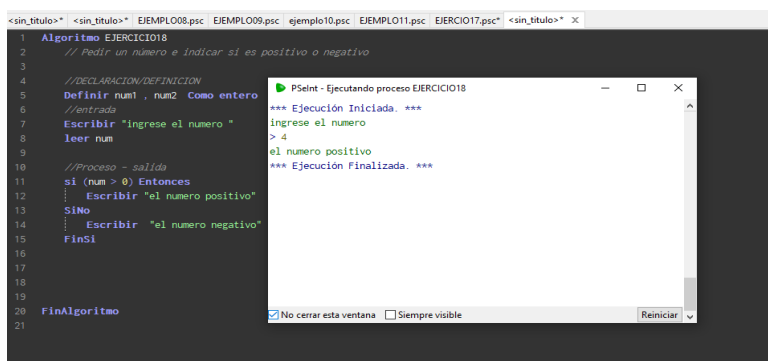
18) Pedir un número e indicar si es positivo o negativo



```
1 Algoritmo EJERCICIO18
2 // Pedir un número e indicar si es positivo o negativo
3
4 //DECLARACION/DEFINICION
5 Definir num1 , num2 Como entero
6 //entrada
7 Escribir "ingrese el numero "
8 leer num
9
10 //Proceso - salida
11 si (num > 0) Entonces
12     Escribir "el numero positivo"
13 SiNo
14     Escribir "el numero negativo"
15 FinSi
16
17
18
19
20 FinAlgoritmo
21
```

Console output:

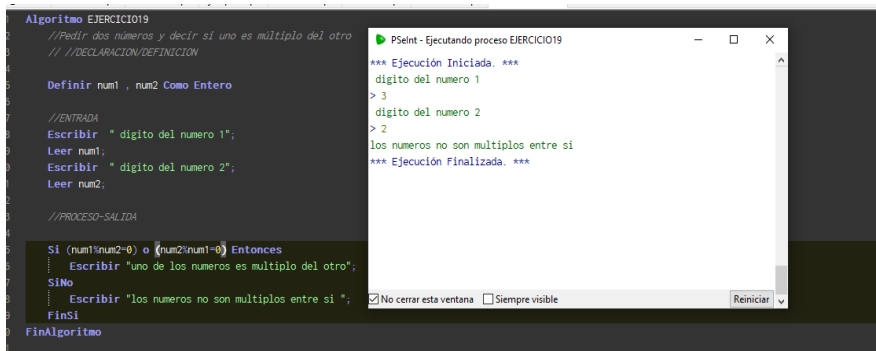
```
*** Ejecución Iniciada. ***
ingrese el numero
> 4
el numero positivo
*** Ejecución Finalizada. ***
```



Console output:

```
*** Ejecución Iniciada. ***
ingrese el numero
> 4
el numero positivo
*** Ejecución Finalizada. ***
```

19) Pedir dos números y decir si uno es múltiplo del otro



```
Algoritmo EJERCICIO19
//Pedir dos números y decir si uno es múltiplo del otro
//DECLARACION/DEFINICION

Definir num1 , num2 Como Entero

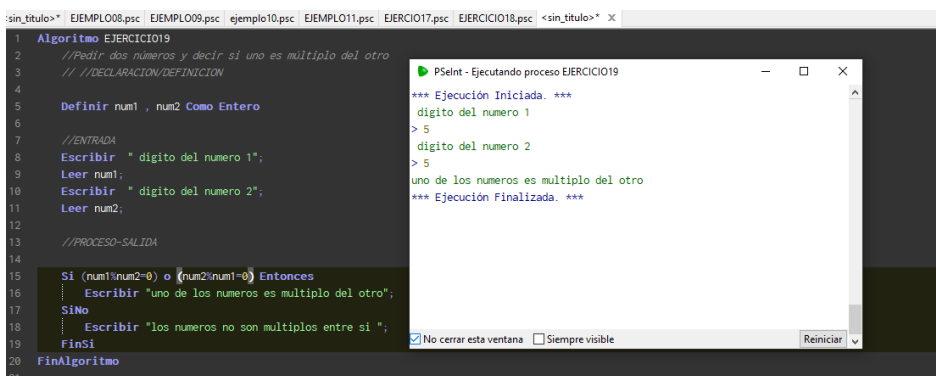
//ENTRADA
Escribir " digito del numero 1";
Leer num1;
Escribir " digito del numero 2";
Leer num2;

//PROCESO-SALIDA

Si (num1%num2=0) o (num2%num1=0) Entonces
    Escribir "uno de los numeros es multiplo del otro";
SiNo
    Escribir "los numeros no son multiplos entre si ";
FinSi
FinAlgoritmo
```

Console output:

```
*** Ejecución Iniciada. ***
digito del numero 1
> 3
digito del numero 2
> 2
los numeros no son multiplos entre si
*** Ejecución Finalizada. ***
```



Console output:

```
*** Ejecución Iniciada. ***
digito del numero 1
> 5
digito del numero 2
> 5
uno de los numeros es multiplo del otro
*** Ejecución Finalizada. ***
```

20) Pedir dos números y decir cuál es el mayor

```
1 Algoritmo EJERCIO20
2 // Pedir dos números y decir cuál es el mayor
3
4 // //DECLARACION/DEFINICION
5
6 Definir num1 , num2 Como Entero
7
8 //ENTRADA
9 Escribir " digito del numero 1";
10 Leer num1;
11 Escribir " digito del numero 2";
12 Leer num2;
13
14 //PROCESO-SALIDA
15 Si (num1 > num2) Entonces
16     Escribir " el numero 1 es mayor que el numero 2"
17 SiNo
18     Si (num2 > num1) Entonces
19         Escribir " el numero 2 es mayor que el numero 1"
20     FinSi
21 FinSi
22
23 FinAlgoritmo
```

PSeInt - Ejecutando proceso EJERCIO20

```
*** Ejecución Iniciada. ***
digito del numero 1
> 7
digito del numero 2
> 4
el numero 1 es mayor que el numero 2
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

```
<sin_titulo>* EJEMPLO08.psc EJEMPLO09.psc ejemplo10.psc EJEMPLO11.psc EJERCIO17.psc EJERCIO18.psc EJERCIO19.psc <sin_titulo>* X
1 Algoritmo EJERCIO20
2 // Pedir dos números y decir cuál es el mayor
3
4 // //DECLARACION/DEFINICION
5
6 Definir num1 , num2 Como Entero
7
8 //ENTRADA
9 Escribir " digito del numero 1";
10 Leer num1;
11 Escribir " digito del numero 2";
12 Leer num2;
13
14 //PROCESO-SALIDA
15 Si (num1 > num2) Entonces
16     Escribir " el numero 1 es mayor que el numero 2"
17 SiNo
18     Si (num2 > num1) Entonces
19         Escribir " el numero 2 es mayor que el numero 1"
20     FinSi
21 FinSi
22
23 FinAlgoritmo
```

PSeInt - Ejecutando proceso EJERCIO20

```
*** Ejecución Iniciada. ***
digito del numero 1
> 1
digito del numero 2
> 9
el numero 2 es mayor que el numero 1
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

21) Pedir dos números y decir cuál es el mayor o si son iguales.

```
<sin_titulo>* EJEMPLO08.psc EJEMPLO09.psc ejemplo10.psc EJEMPLO11.psc EJERCIO17.psc EJERCIO18.psc EJERCIO19.psc EJERCIO20.psc <sin_titulo>* X
1 Algoritmo EJERCIO21
2 // Pedir dos números y decir cuál es el mayor o si son iguales.
3
4 // //DECLARACION/DEFINICION
5
6 Definir num1 , num2 Como Entero
7
8 //ENTRADA
9 Escribir " digito del numero 1";
10 Leer num1;
11 Escribir " digito del numero 2";
12 Leer num2;
13
14 //PROCESO-SALIDA
15 Si (num1 > num2) Entonces
16     Escribir " el numero 1 es mayor que el numero 2"
17 SiNo
18     Si (num2 > num1) Entonces
19         Escribir " el numero 2 es mayor que el numero 1"
20     SiNo
21         Escribir "ambos numeros son iguales"
22     FinSi
23 FinSi
24
25 FinAlgoritmo
```

PSeInt - Ejecutando proceso EJERCIO21

```
*** Ejecución Iniciada. ***
digito del numero 1
> 4
digito del numero 2
> 4
ambos numeros son iguales
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

22) Pedir dos números y mostrarlos ordenados de mayor a menor

```
1 Algoritmo EJERCICIO22
2 // Pedir dos números y mostrarlos ordenados de mayor a menor
3
4 //DECLARACION/DEFINICION
5
6 Definir num1 , num2 Como Entero
7
8 //ENTRADA
9 Escribir " digito del numero 1";
10 Leer num1;
11 Escribir " digito del numero 2";
12 Leer num2;
13
14 //PROCESO-SALIDA
15 Si (num1 > num2) Entonces
16     Escribir "el orden de mayor a menor es " , num1," ",num2
17 SiNo
18     Escribir " el orden de mayor a menor es " , num2," ",num1
19 FinSi
20
21 FinAlgoritmo
```

Execution 1: Inputs 4 and 5. Output: el orden de mayor a menor es 5,4

Execution 2: Inputs 5 and 4. Output: el orden de mayor a menor es 5,4

23) Pedir tres números y mostrarlos ordenados de mayor a menor

```
15 //PROCESO - SALIDA
16 si (num1<num2) Entonces
17     si (num2<num3) Entonces
18         Escribir num3;
19         Escribir num2;
20         Escribir num1;
21     SiNo
22         Si (num1<num3) Entonces
23             Escribir num2;
24             Escribir num1;
25             Escribir num3;
26         SiNo
27             Escribir num2;
28             Escribir num1;
29             Escribir num3;
30         FinSi
31     FinSi
32 SiNo
33     Si (num1<num3) Entonces
34         Escribir num3;
35         Escribir num1;
36         Escribir num2;
37     SiNo
38         Si (num2<num3) Entonces
39             Escribir num1;
40             Escribir num3;
41             Escribir num2;
42         SiNo
43             Escribir num1;
44             Escribir num2;
45             Escribir num3;
46         FinSi
47     FinSi
48 FinSi
49 FinProceso
```

Execution 1: Inputs 9, 5, 7. Output: 9 7 5

Execution 2: Inputs 5, 7, 9. Output: 9 7 5


```
<sin_titulo>
4 //DECLARACIÓN-DEFINICIÓN
5 Definir num1, num2, num3 Como Real;
6
7 //ENTRADA
8 Escribir "Digite el número 1";
9 Leer num1;
10 Escribir "Digite el número 2";
11 Leer num2;
12 Escribir "Digite el número 3";
13 Leer num3;
14
15 //PROCESO - SALIDA
16 Si (num1<num2) Entonces
17   Si (num2<num3) Entonces
18     Escribir num3;
19     Escribir num2;
20     Escribir num1;
21   SiNo
22     Si (num1<num3) Entonces
23       Escribir num2;
24       Escribir num1;
25       Escribir num3;
26     SiNo
27       Escribir num2;
28       Escribir num1;
29       Escribir num3;
30     FinSi
31   FinSi
32 SiNo
33   Si (num1<num3) Entonces
34     Escribir num3;
35     Escribir num1;
36     Escribir num2;
37   SiNo
38     Si (num2<num3) Entonces
39       Escribir num1;
40       Escribir num3;
41       Escribir num2;
42     SiNo
43       Escribir num1;
44       Escribir num2;
45       Escribir num3;
46     FinSi
47   FinSi
48 FinSi
49 FinProceso
50
```

PSeInt - Ejecutando proceso EJERCICIO23

```
*** Ejecución Iniciada. ***
Digite el número 1
> 3
Digite el número 2
> 7
Digite el número 3
> 5
7
3
5
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible Reiniciar

24) Pedir un número entre 0 y 9.999 y decir cuántas cifras tiene

```
Algoritmo EJERCICO24
// Pedir un número entre 0 y 9.999 y decir cuántas cifras tiene

//DECLARACIÓN-DEFINICIÓN
Definir num Como entero;

//ENTRADA
Escribir "escriba un numero del 1 al 9.999"
Leer num1

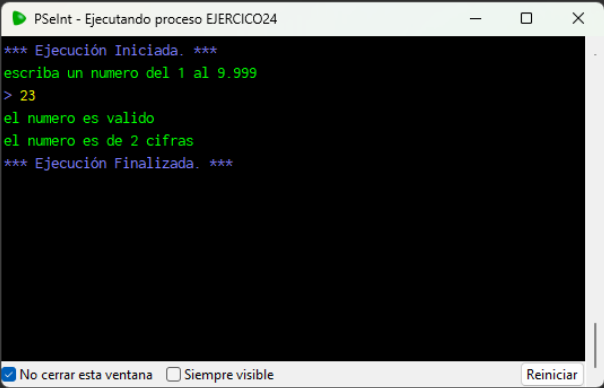
//PROCESO-SALIDA

si (num1<1)Entonces
    Escribir "el numero es invalido"
SiNo
    Escribir "el numero es valido"
FinSi
si (num1>9999) Entonces
    Escribir "el numero es invalido"
FinSi

si (num1>0)y(num1<10)Entonces
    Escribir "el numero es de 1 cifras"
FinSi

    si (num1≥10)y(num1<100) Entonces
        Escribir "el numero es de 2 cifras"
    FinSi

    si (num1>100)y(num1<1000)Entonces
        Escribir "el numero es de 3 cifras"
    SiNo
        si (num1≥1000)y(num1<9999)Entonces
            Escribir "el numero es de 4 cifras"
        FinSi
    FinSi
FinSi
```



```
Algoritmo EJERCICO24
// Pedir un número entre 0 y 9.999 y decir cuántas cifras tiene

//DECLARACIÓN-DEFINICIÓN
Definir num Como entero;

//ENTRADA
Escribir "escriba un numero del 1 al 9.999"
Leer num1

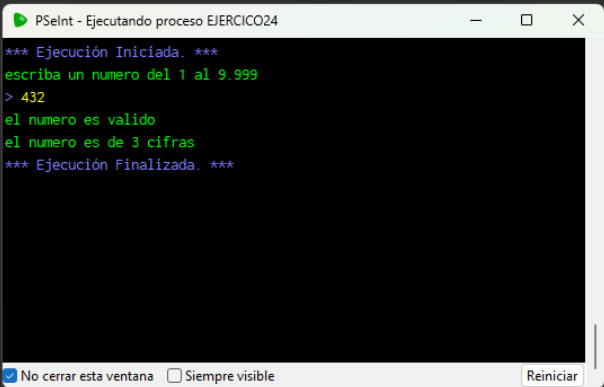
//PROCESO-SALIDA

si (num1<1)Entonces
    Escribir "el numero es invalido"
SiNo
    Escribir "el numero es valido"
FinSi
si (num1>9999) Entonces
    Escribir "el numero es invalido"
FinSi

si (num1>0)y(num1<10)Entonces
    Escribir "el numero es de 1 cifras"
FinSi

    si (num1≥10)y(num1<100) Entonces
        Escribir "el numero es de 2 cifras"
    FinSi

    si (num1>100)y(num1<1000)Entonces
        Escribir "el numero es de 3 cifras"
    SiNo
        si (num1≥1000)y(num1<9999)Entonces
            Escribir "el numero es de 4 cifras"
        FinSi
    FinSi
FinSi
```



25) Pedir una nota de 0 a 5 y mostrarla de la forma: Insuficiente (0 – 2,9), Suficiente (3 – 4,5) y Bien (4,6 – 5)

```
EJERCICIO23.psc ejercicio 24.psc <sin_titulo>* X
1  Algoritmo EJERCICIO25
2      //Pedir una nota de 0 a 5 y mostrarla de la forma: Insuficiente (0 -2,9), Suficiente (3 - 4,5) y Bien (4,6 - 5)
3
4      //DEFINIR VARIABLES
5      Definir NOTA Como Real;
6
7      //ENTRADA
8      Escribir "digite la nota de 0 a 5";
9      leer nota;
10
11      //salida/Proceso
12      si(nota<2.9)Entonces
13          Escribir "insuficiente";
14      SiNo
15          si(nota<3)y(nota<4.5) Entonces
16              Escribir "suficiente";
17          SiNo
18              si(nota<4.6)y(nota<5) Entonces
19                  Escribir "bien";
20              FinSi
21          FinSi
22      FinSi
23
24      FinSi
25
26      FinSi
```

PSeInt - Ejecutando proceso EJERCICIO25

```
*** Ejecución Iniciada. ***
digite la nota de 0 a 5
> 2
insuficiente
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible Reiniciar

```
EJERCICIO23.psc ejercicio 24.psc <sin_titulo>* X
1  Algoritmo EJERCICIO25
2      //Pedir una nota de 0 a 5 y mostrarla de la forma: Insuficiente (0 -2,9), Suficiente (3 - 4,5) y Bien (4,6 - 5)
3
4      //DEFINIR VARIABLES
5      Definir NOTA Como Real;
6
7      //ENTRADA
8      Escribir "digite la nota de 0 a 5";
9      leer nota;
10
11      //salida/Proceso
12      si(nota<2.9)Entonces
13          Escribir "insuficiente";
14      SiNo
15          si(nota<3)y(nota<4.5) Entonces
16              Escribir "suficiente";
17          SiNo
18              si(nota<4.6)y(nota<5) Entonces
19                  Escribir "bien";
20              FinSi
21          FinSi
22      FinSi
23
24      FinSi
25
26      FinSi
```

PSeInt - Ejecutando proceso EJERCICIO25

```
*** Ejecución Iniciada. ***
digite la nota de 0 a 5
> 5
bien
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible Reiniciar

```
1  Algoritmo EJERCICIO25
2      //Pedir una nota de 0 a 5 y mostrarla de la forma: Insuficiente (0 -2,9), Suficiente (3 - 4,5) y Bien (4,6 - 5)
3
4      //DEFINIR VARIABLES
5      Definir NOTA Como Real;
6
7      //ENTRADA
8      Escribir "digite la nota de 0 a 5";
9      leer nota;
10
11      //salida/Proceso
12      si(nota<2.9)Entonces
13          Escribir "insuficiente";
14      SiNo
15          si(nota<3)y(nota<4.5) Entonces
16              Escribir "suficiente";
17          SiNo
18              si(nota<4.6)y(nota<5) Entonces
19                  Escribir "bien";
20              FinSi
21          FinSi
22      FinSi
23
24      FinSi
25
26      FinSi
```

PSeInt - Ejecutando proceso EJERCICIO25

```
*** Ejecución Iniciada. ***
digite la nota de 0 a 5
> 4
suficiente
*** Ejecución Finalizada. ***
```

☒ No cerrar esta ventana ☐ Siempre visible Reiniciar

26) Pedir una nota numérica entera entre 0 y 10, y mostrar dicha nota de la forma: cero, uno, dos, tres etc.

```
EJERCICIO23.psc  ejercicio 24.psc  EJERCICIO 25.psc  <sin_titulo>* X

Algoritmo EJERCICIO26
    // Pedir una nota numérica entera entre 0 y 10, y mostrar dicha nota de la forma: cero, uno, dos, tres
    Definir nota Como Entero
    //Entrada
    Escribir "digite la nota";
    Leer nota;
    //salida/proceso
    si(nota=1)Entonces
        Escribir "uno"
    SiNo
        si(nota=2)Entonces
            Escribir "dos"
        SiNo
            si(nota=3)Entonces
                Escribir "tres"
            SiNo
                si(nota=4)Entonces
                    Escribir "cuatro"
                SiNo
                    si(nota=5)Entonces
                        Escribir "cinco"
                    SiNo
                        si(nota=6)Entonces
                            Escribir "seis"
                        SiNo
                            si(nota=7)Entonces
                                Escribir "siete"
                            SiNo
                                si(nota=8)Entonces
                                    Escribir "ocho"
                                SiNo
                                    si(nota=9)Entonces
                                        Escribir "nueve"
                                    SiNo
                                        si(nota=10)Entonces
                                            Escribir "diez"
                                        FinSi
                                    FinSi
                                FinSi
                            FinSi
                        FinSi
                    FinSi
                FinSi
            FinSi
        FinSi
    FinSi
FinSi
```

27) Pedir un número y decir si es par o impar

```
1 Algoritmo EJERCICIO27
2 // Pedir un número y decir si es par o impar
3
4 //Definir variables
5 Definir num Como Entero
6
7 //entrada
8 Escribir "Ingrese un número:"
9 Leer numero
10
11 //valida/procesa
12 resto ← numero MOD 2
13
14 Si resto = 0 Entonces
15     Escribir "El número ingresado es par."
16 Sino
17     Escribir "El número ingresado es impar."
18 FinSi
19
20 FinAlgoritmo
```

PSeInt - Ejecutando proceso EJERCICIO27

```
*** Ejecución Iniciada. ***
Ingrese un número:
> 11
El número ingresado es impar.
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible Reiniciar

```
1 Algoritmo EJERCICIO27
2     // Pedir un número y decir si es par o impar
3
4     //Definir variables
5     Definir num Como Entero
6
7     //entrada
8     Escribir "Ingrese un número:"
9     Leer numero
10
11     //salida/proceso
12     resto ← numero MOD 2
13
14     Si resto = 0 Entonces
15         Escribir "El número ingresado es par."
16     Sino
17         Escribir "El número ingresado es impar."
18     FinSi
19
20 FinAlgoritmo
```

PSeInt - Ejecutando proceso EJERCICIO27

*** Ejecución Iniciada. ***

Ingrese un número:

> 9

El número ingresado es impar.

*** Ejecución Finalizada. ***

☐ No cerrar esta ventana ☐ Siempre visible Reiniciar

- 28) Un trabajador recibe su pago, según la cantidad de horas trabajadas y su valor. Si la cantidad de horas trabajadas es mayor que 40, éstas se consideran horas extra, y tienen un incremento de \$10000 (diez mil) sobre el valor de la hora. Calcular y mostrar el salario (pago) del trabajador. Nota: leer horas trabajadas y valor de la hora

```
1  Algoritmo EJERCICIO28
2  // Un trabajador recibe su pago, según la cantidad de horas trabajadas y su valor. Si la cantidad de
3  // horas trabajadas es mayor que 40, éstas se consideran horas extra, y tienen un incremento de $10000
4  // diez mil sobre el valor de la hora. Calcular y mostrar el salario (pago) del trabajador. Nota: leer horas
5  // trabajadas y valor de la hora
6
7  //Definir variables
8  Definir horas_trabajadas, valor_hora, salario Como Real
9
10 //entrada
11 Escribir "Ingrese la cantidad de horas trabajadas:"
12 Leer horas_trabajadas
13
14 Escribir "Ingrese el valor de la hora de trabajo:"
15 Leer valor_hora
16
17 //salida/proceso
18 Si horas_trabajadas > 40 Entonces
19     salario ← 40 * valor_hora + (horas_trabajadas - 40) * (valor_hora + 10000)
20 Sino
21     salario ← horas_trabajadas * valor_hora
22 FinSi
23
24 Escribir "El salario del trabajador es: ", salario
25
26 FinAlgoritmo
```

PSeInt - Ejecutando proceso EJERCICIO28

```
*** Ejecución Iniciada. ***
Ingrese la cantidad de horas trabajadas:
> 24
Ingrese el valor de la hora de trabajo:
> 35000
El salario del trabajador es: 840000
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible

- 29) Dado un monto, calcular el descuento considerando que por encima de 100 el descuento es del 10% y por debajo de 100, el descuento es del 2%

```
1  Algoritmo EJERCICIO29
2  // Dado un monto, calcular el descuento considerando que por encima de 100 el descuento es del 10%
3  // y por debajo de 100, el descuento es del 2%.
4
5  //Definir variables
6  Definir monto, descuento Como Real
7
8  //entrada
9  Escribir "Ingrese el monto:"
10 Leer monto
11
12 //salida/proceso
13 Si monto > 100 Entonces
14     descuento ← monto * 0.10
15 Sino
16     descuento ← monto * 0.02
17 FinSi
18
19 Escribir "El descuento es: ", descuento
20
21
22
23 FinAlgoritmo
24
```

PSeInt - Ejecutando proceso EJERCICIO29

```
*** Ejecución Iniciada. ***
Ingrese el monto:
> 78
El descuento es: 1.56
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible

30) Leer dos números y calcular su división, teniendo en cuenta que el denominador no debe ser 0 (cero)

```
1  Algoritmo EJERCICIO30
2  // Leer dos números y calcular su división, teniendo en cuenta que el denominador no debe ser 0 (cero)
3
4  // DEFINIR VARIABLES
5  Definir numerador, denominador, resultado Como Real
6
7  //entrada
8  Escribir "Ingrese el numerador:"
9  Leer numerador
10
11  Escribir "Ingrese el denominador (distinto de cero):"
12  Leer denominador
13
14  //salida/proceso
15  Mientras denominador = 0 Hacer
16  |   Escribir "El denominador no puede ser cero. Ingrese un valor diferente:"
17  |   Leer denominador
18  Fin Mientras
19
20  resultado ← numerador / denominador
21
22  Escribir "El resultado de la división es: ", resultado
23 FinAlgoritmo
24
```

