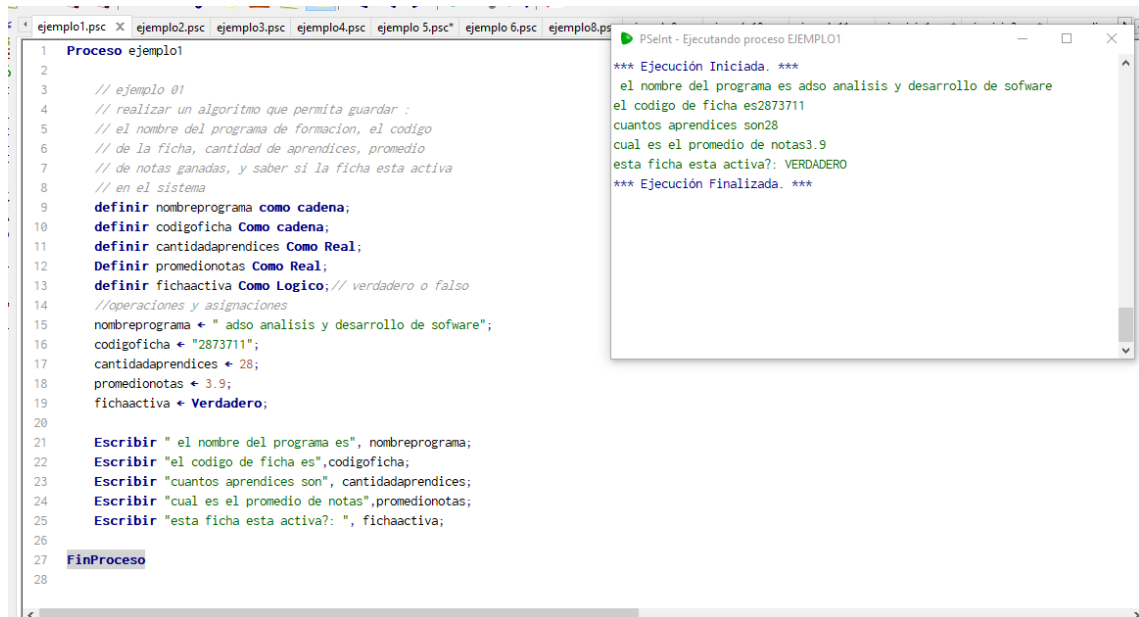


Evidencia algoritmos guía 1-introduccion a la programación

Ejemplos:

- 1) se requiere algoritmo que permita guardar: nombre programa de formación, cantidad de aprendices; código de ficha, promedio de notas, saber si la ficha esta activa



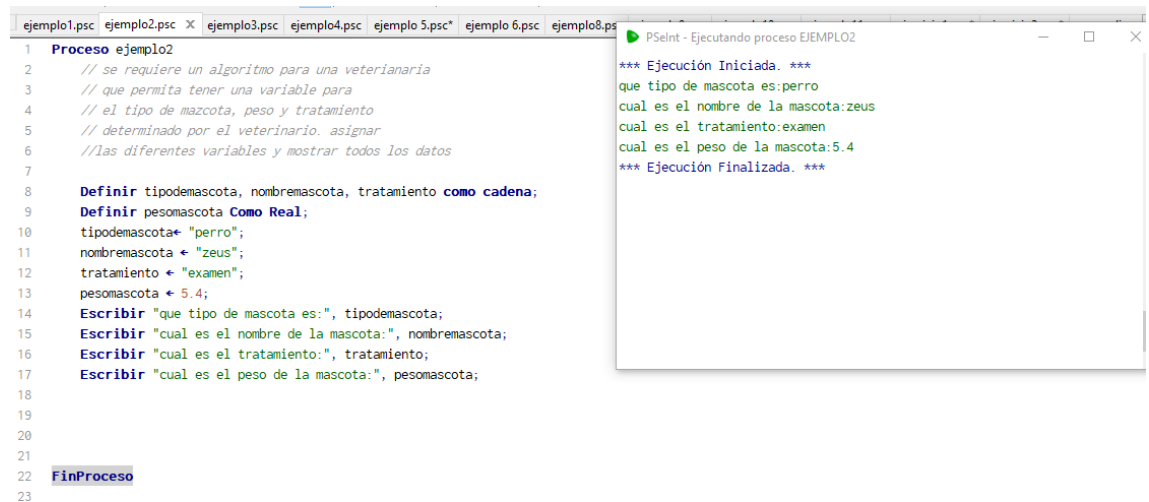
The screenshot shows the PSeInt IDE with a file named 'ejemplo1.psc' open. The code is written in Pascal and defines variables for program name, code, number of apprentices, average grade, and whether the file is active. It then assigns values to these variables and prints them out. A separate window titled 'PSeInt - Ejecutando proceso EJEMPLO1' shows the execution output, which matches the values assigned in the code.

```
1 Proceso ejemplo1
2
3     // ejemplo 01
4     // realizar un algoritmo que permita guardar :
5     // el nombre del programa de formacion, el codigo
6     // de la ficha, cantidad de aprendices, promedio
7     // de notas ganadas, y saber si la ficha esta activa
8     // en el sistema
9     definir nombreprograma como cadena;
10    definir codigoficha Como cadena;
11    definir cantidadaprendices Como Real;
12    Definir promedionotas Como Real;
13    definir fichaactiva Como Logico; // verdadero o falso
14    //operaciones y asignaciones
15    nombreprograma ← " adso analisis y desarrollo de software";
16    codigoficha ← "2873711";
17    cantidadaprendices ← 28;
18    promedionotas ← 3.9;
19    fichaactiva ← Verdadero;
20
21    Escribir " el nombre del programa es", nombreprograma;
22    Escribir "el codigo de ficha es",codigoficha;
23    Escribir "cuantos aprendices son", cantidadaprendices;
24    Escribir "cual es el promedio de notas",promedionotas;
25    Escribir "esta ficha esta activa?: ", fichaactiva;
26
27    FinProceso
28
```

*** Ejecución Iniciada. ***
el nombre del programa es adso analisis y desarrollo de software
el codigo de ficha es2873711
cuantos aprendices son28
cual es el promedio de notas3.9
esta ficha esta activa?: VERDADERO
*** Ejecución Finalizada. ***

- 2) Se requiere un algoritmo para una veterinaria, que permita tener una 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



The screenshot shows a code editor with multiple tabs. The active tab is 'ejemplo2.psc', which contains the following Pascal code:

```
1 Proceso ejemplo2
2   // se requiere un algoritmo para una veterinaria
3   // que permita tener una variable para
4   // el tipo de mascota, peso y tratamiento
5   // determinado por el veterinario. asignar
6   // las diferentes variables y mostrar todos los datos
7
8   Definir tipodemascota, nombremascota, tratamiento como cadena;
9   Definir pesomascota Como Real;
10  tipodemascota ← "perro";
11  nombremascota ← "zeus";
12  tratamiento ← "examen";
13  pesomascota ← 5.4;
14  Escribir "que tipo de mascota es:", tipodemascota;
15  Escribir "cual es el nombre de la mascota:", nombremascota;
16  Escribir "cual es el tratamiento:", tratamiento;
17  Escribir "cual es el peso de la mascota:", pesomascota;
18
19
20
21
22 FinProceso
23
```

To the right of the code editor, a small window titled 'PSeInt - Ejecutando proceso EJEMPLO2' displays the execution output:

```
*** Ejecución Iniciada. ***
que tipo de mascota es:perro
cual es el nombre de la mascota:zeus
cual es el tratamiento:examen
cual es el peso de la mascota:5.4
*** Ejecución Finalizada. ***
```

- 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 de las PPL, el nombre de la cárcel y la dirección de la cárcel. Mostrar todos los datos

```
1 Proceso ejemplo3
2 //en una carcel se requiere un Algoritmo
3 // que permita identificar
4 //cuantas personas privadas de la libertad tiene la carcel
5 // el promedio de años de condena de la ppl
6 // el nombre de la carcel y direccion de la carcel.mostrar todos los datos
7
8 Definir cantidadPPL Como Entero;
9 Definir proanosPPL Como real;
10 Definir nombrecarcel como cadena;
11 Definir direccion como cadena;
12 cantidadPPL ← 1200;
13 proanosPPL ← 8;
14 nombrecarcel ← "la blanca";
15 direccion ← "calle10c #32a33";
16 Escribir "la cantidad de ppl es", cantidadPPL;
17 Escribir "el promedio de los ppl es", proanosPPL;
18 Escribir "el nombre de la carcel es", nombrecarcel;
19 Escribir "cual es la diereccion de la carcel", direccion;
20
21
22
23 FinProceso
24
```

PSeInt - Ejecutando proceso EJEMPLO3

```
*** Ejecución Iniciada. ***
la cantidad de ppl es1200
el promedio de los ppl es8
el nombre de la carcel esla blanca
cual es la diereccion de la carcelcalle10c #32a33
*** Ejecución Finalizada. ***
```

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

```
1 Proceso ejemplo4
2 // declaracion/ definicion de variables
3 definir x,a Como Real;
4 // entrada de datos
5 Escribir " digite el valor del numero";
6 Leer a;
7
8 // proceso, operaciones o formulas
9 x ← -2*a + a * 5;
10
11 //salida de datos
12 Escribir "el resultado de la operacion", x;
13
14 FinProceso
15
```

PSeInt - Ejecutando proceso EJEMPLO4

```
*** Ejecución Iniciada. ***
digite el valor del numero
> 5
el resultado de la operacion35
*** Ejecución Finalizada. ***
```

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

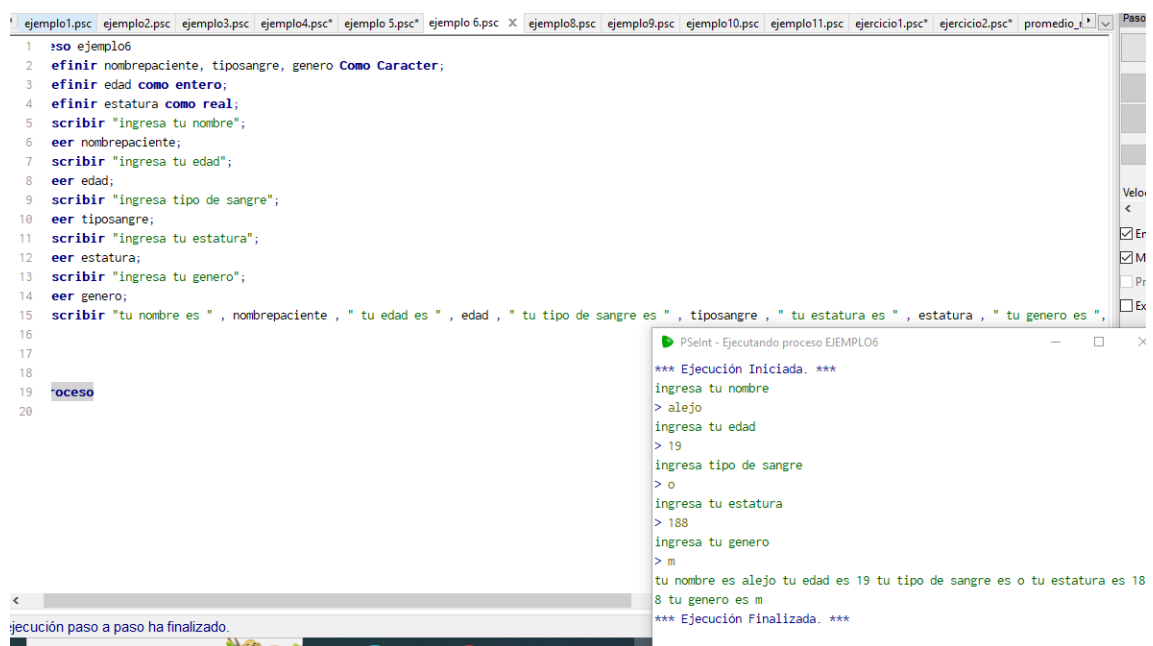
```
ejemplo1.psc ejemplo2.psc ejemplo3.psc ejemplo4.psc* ejemplo 5.psc* X ejemplo 6.psc ejemplo8.psc ejemplo9.psc ejemplo10.psc ejemplo11.psc ejercicio1.psc* ejercicio2.psc* promedio_1.psc* Paso a paso
5 Definir estudio Como Logico;
6 estudio ← Falso;
7 escribir "ingresa tu nombre";
8 Leer nombredelaprendiz;
9 Escribir "tipodedocumento";
10 Leer tipodedocumento;
11 Escribir " ingresa la direccion de la residencia";
12 leer direccionresidencia;
13 Escribir "ingresa cual es tu genero";
14 Leer genero;
15 escribir "cual es tu documento";
16 leer documento;
17 Escribir "cual es tu estrato";
18 Leer estrato;
19 Escribir "cual es tu peso";
20 Leer peso;
21 Escribir " el aprendiz ha estudiado en el sena";
22 Leer estudio;
23 Escribir " cual es tu nombre " , nombredelaprendiz , " cual es tu tipo de documento " , tipodedocumento , " cual es tu direccion de residencia " , direccionresidencia , " cual es tu genero " , genero , " cual es tu documento " , documento , " cual es tu estrato " , estrato , " cual es tu peso " , peso , " has estudiado en el sena " , estudio ;
24 escribir " cual es tu direccion de residencia " , direccionresidencia , " cual es tu genero " , genero , " cual es tu documento " , documento , " cual es tu estrato " , estrato , " cual es tu peso " , peso , " has estudiado en el sena " , estudio ;
25 escribir "cual es tu estrato" , estrato , " cual es tu peso" , peso , " has estudiado en el sena " , estudio ;
26
27
28
29
30
31 FinProceso
32
```

```
PSeInt - Ejecutando proceso EJEMPLOS
*** Ejecución Iniciada. ***
ingresa tu nombre
> alejo
tipodedocumento
> cc
ingresa la direccion de la residencia
> cl110b #35a14
ingresa cual es tu genero
> m
cual es tu documento
> 1055748158
cual es tu estrato
> 2
cual es tu peso
> 15
el aprendiz ha estudiado en el sena
> falso
cual es tu nombre alejo cual es tu tipo de documento cc
cual es tu direccion de residencia cl110b #35a14 cual es tu genero m cual
es tu documento1055748158
cual es tu estrato2 cual es tu peso15 has estudiado en el sena FALSO
*** Ejecución Finalizada. ***
```

6) En un hospital se quiere un algoritmo que solicite al usuario los siguientes datos:

nombre del paciente, edad, tipo de sangre, estatura y género

Mostrar todos los datos al final



The screenshot displays the PSeInt IDE interface. The main editor window shows a Pascal program named 'ejemplo6.psc' with the following code:

```
1  uso ejemplo6
2  efinir nombrepaciente, tiposangre, genero Como Caracter;
3  efinir edad como entero;
4  efinir estatura como real;
5  scribir "ingresa tu nombre";
6  eer nombrepaciente;
7  scribir "ingresa tu edad";
8  eer edad;
9  scribir "ingresa tipo de sangre";
10 eer tiposangre;
11 scribir "ingresa tu estatura";
12 eer estatura;
13 scribir "ingresa tu genero";
14 eer genero;
15 scribir "tu nombre es ", nombrepaciente, " tu edad es ", edad, " tu tipo de sangre es ", tiposangre, " tu estatura es ", estatura, " tu genero es ",
16
17
18
19 proceso
20
```

Below the editor, a console window titled 'PSeInt - Ejecutando proceso EJEMPLO6' shows the execution output:

```
*** Ejecución Iniciada. ***
ingresa tu nombre
> alejo
ingresa tu edad
> 19
ingresa tipo de sangre
> o
ingresa tu estatura
> 188
ingresa tu genero
> m
tu nombre es alejo tu edad es 19 tu tipo de sangre es o tu estatura es 188
tu genero es m
*** Ejecución Finalizada. ***
```

At the bottom of the IDE, a status bar indicates 'Ejecución paso a paso ha finalizado.'

```
ejemplo1.psc | ejemplo2.psc | ejemplo3.psc | ejemplo4.psc | ejemplo 5.psc | ejemplo 6.psc | ejemplo8.psc |
1 Proceso ejemplo8
2 //condicional simple 1
3 //definicion/declaracion de variable
4 Definir estadoclima como cadena;
5 estadoclima= "lluvias";
6 si (estadoclima= "lluvias") Entonces
7 | Escribir "sacar sombrilla";
8 |
9 FinSi
10 Escribir "caminar por la calle";
11
12 FinProceso
13
```

PSeInt - Ejecutando proceso EJEMPLO8

```
*** Ejecución Iniciada. ***
sacar sombrilla
caminar por la calle
*** Ejecución Finalizada. ***
```

Paso a paso

Velocidad: <

☒ Entrar en su

☒ Mostrar tra:

☐ Prueba de E

☐ Explicar en

Ayu

```
ejemplo1.psc | ejemplo2.psc | ejemplo3.psc | ejemplo4.psc | ejemplo 5.psc | ejemplo 6.psc | ejemplo8.psc |
1 Proceso ejemplo9
2 //condicional
3 //definicion
4 Definir estrato Como Entero;
5
6
7 //entrada
8 Escribir "digite su estrato";
9 Leer estrato;
10 //Procesos- salida
11 Escribir " bienvenido al sena regional caldas";
12 si (estrato < 3) Entonces
13 | Escribir "puede aplicar para apoyo de sostenimiento";
14 |
15 FinSi
16 //si ( estrato < 3) entonces
17 FinProceso
18
```

PSeInt - Ejecutando proceso EJEMPLO9

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

```
ejemplo1.psc | ejemplo2.psc | ejemplo3.psc | ejemplo4.psc | ejemplo 5.psc | ejemplo 6.psc | ejemplo8.psc |
1 Proceso ejemplo10
2 //definicion
3 Definir estrato Como Entero;
4 Escribir "digite su estrato";
5 Leer estrato;
6 // Proceso
7 Escribir "bienvenidos al sena caldas";
8 si(estrato ≤ 2) Entonces
9 | Escribir "puede aplicar para apoyos de sostenimiento";
10 |
11 SiNo
12 | Escribir "no puede acceder a la convocatoria";
13 |
14 FinSi
15 FinProceso
16
```

PSeInt - Ejecutando proceso EJEMPLO10

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

Velocidad: <

☒ En

☒ M

☐ Pr

☐ Ex

The screenshot shows the PSeInt IDE with a file explorer at the top displaying several files: ejemplo1.psc, ejemplo2.psc, ejemplo3.psc, ejemplo4.psc*, ejemplo 5.psc*, ejemplo 6.psc, and ejemplo8.psc. The main editor window contains the following Pascal code:

```
1 Proceso ejemplo1
2 // condicional doble 2
3 // se tiene el resultado de un examen para un
4 // aprendiz. si la nota es menor a 3, es decir que la
5 // perdio, de lo contrario mostrar que si gano
6 // la nota
7
8 //definir variables
9 definir nota Como Real;
10 Escribir "digite su nota";
11 leer nota;
12 //proceso-salida
13 Escribir "bienvenido al sena";
14 si(nota <3) Entonces
15 | Escribir "perdio la nota";
16 SiNo
17 | Escribir "gano la nota";
18 FinSi
19
20
21 FinProceso
22
```

Overlaid on the right is a console window titled "PSeInt - Ejecutando proceso EJEMPLO11". It displays the execution output:

```
*** Ejecución Iniciada. ***
digite su nota
> 5
bienvenido al sena
gano la nota
*** Ejecución Finalizada. ***
```

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.

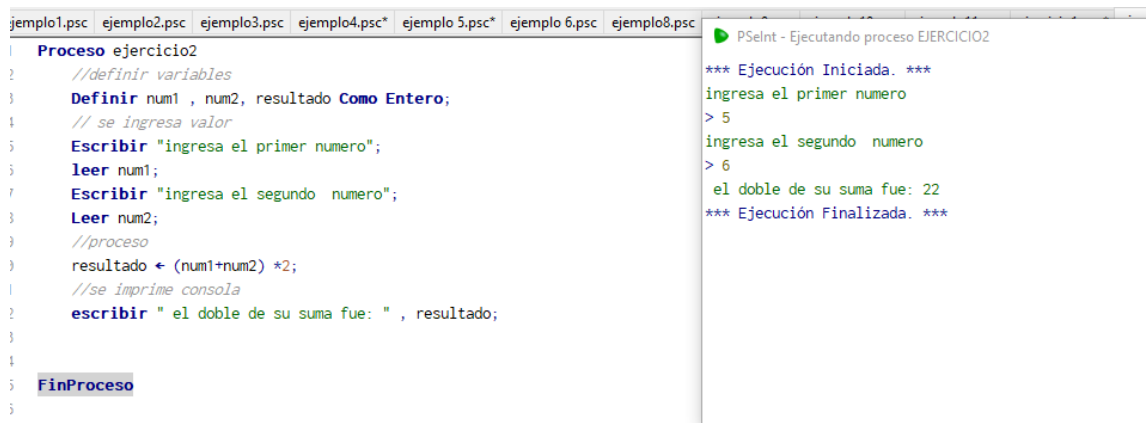
The screenshot shows the PSeInt IDE with a file explorer at the top displaying several files: ejemplo1.psc, ejemplo2.psc, ejemplo3.psc, ejemplo4.psc*, ejemplo 5.psc*, ejemplo 6.psc, and ejemplo8.psc. The main editor window contains the following Pascal code:

```
1 Proceso ejercicio1
2
3 //definir variables
4 definir horasestudio como real;
5 definir nombre como cadena;
6 horasestudio ← 9.5;
7 nombre ← "juan";
8
9 Escribir "las horas de estudio son:", horasestudio;
10 Escribir "su nombre es", nombre;
11 FinProceso
12
```

Overlaid on the right is a console window titled "PSeInt - Ejecutando proceso EJERCICIO1". It displays the execution output:

```
*** Ejecución Iniciada. ***
las horas de estudio son:9.5
su nombre esjuan
*** Ejecución Finalizada. ***
```

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



The screenshot shows the PSeInt IDE with a file explorer at the top containing files like 'ejemplo1.psc', 'ejemplo2.psc', etc. The main editor displays a Pascal program named 'Proceso ejercicio2'. The program defines variables 'num1', 'num2', and 'resultado' as integers. It prompts the user to enter two numbers, reads them, calculates the double of their sum, and prints the result. The execution window on the right shows the program running with inputs 5 and 6, resulting in an output of 22.

```
Proceso ejercicio2
2 //definir variables
3 Definir num1 , num2, resultado Como Entero;
4 // se ingresa valor
5 Escribir "ingresa el primer numero";
6 Leer num1;
7 Escribir "ingresa el segundo numero";
8 Leer num2;
9 //proceso
10 resultado ← (num1+num2) *2;
11 //se imprime consola
12 escribir " el doble de su suma fue: " , resultado;
13
14 FinProceso
```

PSeInt - Ejecutando proceso EJERCICIO2

```
*** Ejecución Iniciada. ***
ingresa el primer numero
> 5
ingresa el segundo numero
> 6
el doble de su suma fue: 22
*** Ejecución Finalizada. ***
```

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.


```
ejemplo3.psc | ejemplo4.psc* | ejemplo 5.psc* | ejemplo 6.psc | ejemplo8.psc | ejemplo9.psc | ejemplo10.psc
1 Proceso ejercicio3
2 //definir las variables
3 Definir promedio como real;
4 Definir nombre como caracter;
5 Definir notas_perdidas como entero;
6 //proceso
7 promedio ← 3.6;
8 nombre ← "alejo" ;
9 notas_perdidas ← 2;
10 //se imprime la consola
11 Escribir "su promedio de notas es: " , promedio;
12 Escribir "su nombre es: " , nombre;
13 Escribir "sus notas perdidas en total son: " , notas_perdidas;
14
15 FinProceso
16
```

PSeInt - Ejecutando proceso EJERCICIO3

```
*** Ejecución Iniciada. ***
su promedio de notas es: 3.6
su nombre es: alejo
sus notas perdidas en total son: 2
*** Ejecución Finalizada. ***
```

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

```
ejemplo3.psc | ejemplo4.psc* | promedio_nombre_notas.psc | ejemplo6.psc | Ejercicio9.psc | ejemplo10.psc | ejemplo11.psc | ejemplo12.psc | ejemplo13.psc | ejemplo14.psc | ejemplo15.psc | ejemplo16.psc | ejemplo17.psc | ejemplo18.psc | ejemplo19.psc | ejemplo20.psc | ejemplo21.psc | ejemplo22.psc | ejemplo23.psc | ejemplo24.psc | ejemplo25.psc | ejemplo26.psc | ejemplo27.psc | ejemplo28.psc | ejemplo29.psc | ejemplo30.psc | ejemplo31.psc | ejemplo32.psc | ejemplo33.psc | ejemplo34.psc | ejemplo35.psc | ejemplo36.psc | ejemplo37.psc | ejemplo38.psc | ejemplo39.psc | ejemplo40.psc | ejemplo41.psc | ejemplo42.psc | ejemplo43.psc | ejemplo44.psc | ejemplo45.psc | ejemplo46.psc | ejemplo47.psc | ejemplo48.psc | ejemplo49.psc | ejemplo50.psc | ejemplo51.psc | ejemplo52.psc | ejemplo53.psc | ejemplo54.psc | ejemplo55.psc | ejemplo56.psc | ejemplo57.psc | ejemplo58.psc | ejemplo59.psc | ejemplo60.psc | ejemplo61.psc | ejemplo62.psc | ejemplo63.psc | ejemplo64.psc | ejemplo65.psc | ejemplo66.psc | ejemplo67.psc | ejemplo68.psc | ejemplo69.psc | ejemplo70.psc | ejemplo71.psc | ejemplo72.psc | ejemplo73.psc | ejemplo74.psc | ejemplo75.psc | ejemplo76.psc | ejemplo77.psc | ejemplo78.psc | ejemplo79.psc | ejemplo80.psc | ejemplo81.psc | ejemplo82.psc | ejemplo83.psc | ejemplo84.psc | ejemplo85.psc | ejemplo86.psc | ejemplo87.psc | ejemplo88.psc | ejemplo89.psc | ejemplo90.psc | ejemplo91.psc | ejemplo92.psc | ejemplo93.psc | ejemplo94.psc | ejemplo95.psc | ejemplo96.psc | ejemplo97.psc | ejemplo98.psc | ejemplo99.psc | ejemplo100.psc
1 Proceso ejercicio9
2 //definir variables
3 Definir nombreestudiante Como Caracter;
4 Definir cantidadperdidas, cantidadganadas Como Entero;
5 //se ingresa valor
6 cantidadperdidas ← 3;
7 cantidadganadas ← 2 ;
8 //se pide datos al usuario
9 Escribir "ingresa el nombre del estudiante";
10 leer nombreestudiante;
11 //imprime consola
12 Escribir "tiene", "materias perdidas", cantidadperdidas, "materias ganadas", cantidadganadas;
13
14
15 FinProceso
16
```

PSeInt - Ejecutando proceso EJERCICIO9

```
*** Ejecución Iniciada. ***
ingresa el nombre del estudiante
> samuel
tienematerias peridas3materias ganadas2
*** Ejecución Finalizada. ***
```

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.

```
ejercicio1.psc* ejercicio2.psc* promedio_nombre_notas.psc* ejercico4.psc* EJERCICIO5.psc* X ejercicio6.psc ejercicio7.psc* ejercicio8.psc* ejercicio9.psc* ejercicio10.psc ejercicio11.psc*
1  Proceso EJERCICIO5
2      //definir variables
3      Definir nombredepersona, comidapreferida,cantidaddediner como caracter;
4      //proceso
5      nombredepersona← "alejo";
6      comidapreferida ← "lentejas" ;
7      cantidaddediner ← " 1 millon 900 mil ";
8      //se imprime consola
9      Escribir " su nombre es: " , nombredepersona;
10     Escribir " su comida favorita es: " , comidapreferida;
11     Escribir " el dinero que posee es: " , cantidaddediner;
12
13
14     FinProceso
15
```

```
PSInt - Ejecutando proceso EJERCICIO5
*** Ejecución Iniciada. ***
su nombre es; alejo
su comida favorita es: lentejas
el dinero que posee es:  1 millon 900 mil
*** Ejecución Finalizada. ***
```

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.

```

ejercicio1.psc* | ejercicio2.psc* | promedio_nombre_notas.psc* | ejercico4.psc* | EJERCICIO5.psc* | ejercicio6.psc*
1  Proceso ejercicio6
2  // se ponen las variables
3  definir nombrearticulo Como Cadena;
4  Definir cantidadacomprar Como Entero;
5  definir valorunitario, totalapagar Como Real;
6  // solicitar los datos al usuario
7  Escribir "cual es el articulo";
8  Leer nombrearticulo;
9  escribir "cuanto es el valor del aticulo";
10 leer valorunitario;
11 Escribir "cual es la cantidad a comprar";
12 leer cantidadacomprar;
13 Escribir "el nombre del articulo", nombrearticulo;
14 totalapagar ← valorunitario * cantidadacomprar;
15 escribir "el total a pagar es", totalapagar;
16
17
18 FinProceso
19

```

PSeInt - Ejecutando proceso EJERCICIO6

```

*** Ejecución Iniciada. ***
cual es el articulo
> doritos
cuanto es el valor del aticulo
> 2000
cual es la cantidad a comprar
> 3
el nombre del articulo doritos
el total a pagar es 6000
*** Ejecución Finalizada. ***

```

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

```

ejercicio1.psc* | ejercicio2.psc* | promedio_nombre_notas.psc* | ejercico4.psc* | EJERCICIO5.psc* | ejercicio6.psc* | ejercicio7.psc* | ejercicio8.psc* | ejercicio9.psc* | ejercicio10.psc | ejercicio11.psc*
Proceso ejercicio7
1  //definir variables
2  definir A, B, resultado como real;
3  //se ingresa valor
4  Escribir "ingresa el primer numero";
5  leer A;
6  Escribir "ingresa el segundo numero";
7  leer B;
8  //Proceso
9  resultado ← A + B;
10 // se imprime consola
11 Escribir " el resultado de sumar ambos numeros es: ", resultado;
12
13
14 FinProceso
15

```

PSeInt - Ejecutando proceso EJERCICIO7

```

*** Ejecución Iniciada. ***
ingresa el primer numero
> 6
ingresa el segundo numero
> 5
el resultado de sumar ambos numeros es: 11
*** Ejecución Finalizada. ***

```

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

ejercicio7.psc*

ejercicio8.psc*

ejercicio9.psc*

ejercicio10.psc*

ejercicio11.psc*

ejercicio12.psc* X

ejercicio13.psc*

ejercicio14.psc

ejercicio15.psc*

ejercicio16.psc*

ejercicio17.psc

ejercicio18.psc*

```
1 Proceso ejercicio12
2 //definir variables
3 Definir nombre_persona Como Caracter;
4 Definir valor_hora_trabajada, pago_del_trabajador Como Real;
5 Definir numero_horas_trabajo como Entero;
6 //se ingresa datos del usuario
7 Escribir "ingresa el nombre del trabajador";
8 Leer nombre_persona;
9 Escribir "¿Cual es el valor de la hora trabajada?";
10 Leer valor_hora_trabajada;
11 Escribir "Numero de horas que trabaja";
12 Leer numero_horas_trabajo;
13 //proceso
14 pago_del_trabajador ← valor_hora_trabajada * numero_horas_trabajo;
15 //se imprime consola
16 Escribir "El pago del trabajador", nombre_persona, "es:", pago_del_trabajador;
17 FinProceso
18
```

PSeInt - Ejecutando proceso EJERCICIO12

*** Ejecución Iniciada. ***
ingresa el nombre del trabajador
> miguel
¿Cual es el valor de la hora trabajada?
> 2000
Numero de horas que trabaja
> 4
El pago del trabajadormiguel es: 8000
*** Ejecución Finalizada. ***

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

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

```

1  Proceso ejercicio10
2
3      // definir variables
4  Definir alto Como Real;
5  Definir ancho Como Real;
6  Definir area, perimetro Como Real;
7
8      //entrada
9  Escribir "digite el alto del rectangulo";
10 Leer alto;
11 Escribir "digite el ancho del rectangulo";
12 Leer ancho;
13 //Proceso
14 perimetro ← 2 * alto + ancho;
15 area ← ancho * alto;
16
17 Escribir "el perimetro del rectangulo es;" perimetro;
18 Escribir "el area del rectangulo es: " area;
19
20 //salida
21 FinProceso
22

```

```

*** Ejecución Iniciada. ***
digite el alto del rectangulo
> 5
digite el ancho del rectangulo
> 6
el perimetro del rectangulo es;16
el area del rectangulo es: 30
*** Ejecución Finalizada. ***

```

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

```

ejercicio1.psc* ejercicio2.psc* promedio_nombre_notas.psc* ejercicio4.psc* EJERCICIO5.psc* ejercicio6.psc*
1  Proceso ejercicio11
2      //definir variables
3  Definir A, B, diferencia Como Entero;
4      //se ingresa valor
5  Escribir "Ingresa el primer numero";
6  Leer A;
7  Escribir "Ingresa el segundo numero";
8  Leer B;
9      //proceso
10 diferencia ← A - B;
11 //se imprime consola
12 Escribir "La diferencia entre ambos numeros es: ", diferencia;
13 FinProceso
14

```

```

PSeInt - Ejecutando proceso EJERCICIO11
*** Ejecución Iniciada. ***
Ingresa el primer numero
> 5
Ingresa el segundo numero
> 6
La diferencia entre ambos numeros es: -1
*** Ejecución Finalizada. ***

```

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.

The screenshot shows the PSeInt IDE with the code for Ejercicio 12. The code defines variables for a worker's name, hourly wage, and hours worked, then calculates the total payment. The execution output shows the user inputting 'miguel' for the name and '2000' for the hours, resulting in a payment of 8000.

```

1 Proceso ejercicio12
2   //definir variables
3   Definir nombre_persona Como Caracter;
4   Definir valor_hora_trabajada, pago_del_trabajador Como Real;
5   Definir numero_horas_trabajo Como Entero;
6   //se ingresa datos del usuario
7   Escribir "ingresa el nombre del trabajador";
8   Leer nombre_persona;
9   Escribir "¿Cual es el valor de la hora trabajada?";
10  Leer valor_hora_trabajada;
11  Escribir "Numero de horas que trabaja";
12  Leer numero_horas_trabajo;
13  //proceso
14  pago_del_trabajador ← valor_hora_trabajada * numero_horas_trabajo;
15  //se imprime consola
16  Escribir "El pago del trabajador", nombre_persona, "es:", pago_del_trabajador;
17 FinProceso
18

```

PSeInt - Ejecutando proceso EJERCICIO12

```

*** Ejecución Iniciada. ***
ingresa el nombre del trabajador
> miguel
¿Cual es el valor de la hora trabajada?
> 2000
Numero de horas que trabaja
> 4
El pago del trabajador miguel es: 8000
*** Ejecución Finalizada. ***

```

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

The screenshot shows the PSeInt IDE with the code for Ejercicio 13. The code defines variables for the radius and area of a circle, then calculates the area using the formula $A=PI*r^2$. The execution output shows the user inputting '5' for the radius, resulting in an area of 78.5398163397.

```

1 Proceso ejercicio13
2   //definir variables
3   Definir radio_circulo, area Como Real;
4   //ingresar valor
5   Escribir "Ingresa el radio del circulo: ";
6   Leer radio_circulo;
7   //proceso
8   area ← PI * radio_circulo^2;
9   //se imprime consola
10  Escribir "El area del circulo es: ", area;
11 FinProceso
12

```

PSeInt - Ejecutando proceso EJERCICIO13

```

*** Ejecución Iniciada. ***
Ingresa el radio del circulo:
> 5
El area del circulo es: 78.5398163397
*** Ejecución Finalizada. ***

```

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

The screenshot shows the PSeInt IDE with the code for Ejercicio 14. The code defines variables for the radius and circumference of a circle, then calculates the circumference using the formula $L=2*PI*r$. The execution output shows the user inputting '5' for the radius, resulting in a circumference of 31.4159265359.

```

1 Proceso ejercicio14
2   //definir variables
3   Definir radio_circunferencia, longitud_circunferencia Como Real;
4   // se ingresa el valor
5   Escribir "Ingresa el radio de la circunferencia";
6   Leer radio_circunferencia;
7   // proceso
8   longitud_circunferencia ← 2 * PI * radio_circunferencia;
9   // se imprime en consola
10  Escribir "La longitud de la circunferencia es: ", longitud_circunferencia;
11 FinProceso
12

```

PSeInt - Ejecutando proceso EJERCICIO14

```

*** Ejecución Iniciada. ***
Ingresa el radio de la circunferencia
> 5
La longitud de la circunferencia es: 31.4159265359
*** Ejecución Finalizada. ***

```

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

```
Proceso ejercicio15
//definir variables
Definir lado_cuadrado, area, perimetro Como Real;
//se ingresa valor
Escribir "Ingresa el lado del cuadrado";
Leer lado_cuadrado;
//Proceso
area ← lado_cuadrado * lado_cuadrado;
perimetro ← lado_cuadrado * 4;
//se imprime consola
Escribir "El area del cuadrado es: ", area;
Escribir "El perimetro del cuadrado es: ", perimetro;
FinProceso
```

PSeInt - Ejecutando proceso EJERCICIO15

```
*** Ejecución Iniciada. ***
Ingresa el lado del cuadrado
> 5
El area del cuadrado es: 25
El perimetro del cuadrado es: 20
*** Ejecución Finalizada. ***
```

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

```
Proceso ejercicio16
//definir variables
Definir area_rectangulo, largo, ancho Como Real;
//se ingresa valor
Escribir "Ingresa el largo del rectangulo";
Leer largo;
Escribir "Ingresa el ancho del rectangulo";
Leer ancho;
//Proceso
area_rectangulo ← largo * ancho;
//se imprime consola
Escribir "El area del rectangulo es: ", area_rectangulo;
FinProceso
```

PSeInt - Ejecutando proceso EJERCICIO16

```
*** Ejecución Iniciada. ***
Ingresa el largo del rectangulo
> 5
Ingresa el ancho del rectangulo
> 9
El area del rectangulo es: 45
*** Ejecución Finalizada. ***
```

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

```
1 Proceso ejercicio17
2 // pedir 2 numeros y decir si son iguales o no
3
4 //declaracion
5 Definir num1, num2 Como Entero;
6 //entrada
7 Escribir "digite el numero 1 y el numero 2";
8 Leer num1, num2;
9 //Proceso
10 si(num1 = num2) Entonces
11     Escribir "los numeros son iguales";
12 SiNo
13     Escribir "los numeros son diferentes";
14 FinSi
15 FinProceso
16
```

PSeInt - Ejecutando proceso EJERCICIO17

```
*** Ejecución Iniciada. ***
digite el numero 1 y el numero 2
> 1
> 9
los numeros son diferentes
*** Ejecución Finalizada. ***
```

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

```
ejercicio10.psc* ejercicio11.psc* ejercicio12.psc* ejercicio13.psc* ejercicio14.psc ejercicio15.psc* ejerc
Proceso ejercicio18
    //definimos variables
    Definir num1 Como Real;
    Escribir "inserta el primer numero";
    Leer num1;

    //Proceso-salida

    si (num1 > 0) Entonces
        Escribir "el numero es positivo";
    SiNo
        Escribir "el numero es negativo";
    FinSi

FinProceso
```

PSeInt - Ejecutando proceso EJERCICIO18

```
*** Ejecución Iniciada. ***
inserta el primer numero
> 7
el numero es positivo
*** Ejecución Finalizada. ***
```

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

```
ejercicio10.psc* ejercicio11.psc* ejercicio12.psc* ejercicio13.psc* ejercicio14.psc ejercicio15.psc* ejerc
Proceso ejercicio19
    // definicion de variables
    definir num1,num2, resto Como Entero;
    //copiar los valores
    Escribir "insertar los 2 numeros";
    Leer num1,num2;
    //proceso-salida
    resto ← num1 % num2;
    si(resto = 0) Entonces
        Escribir num1, " es multiplo de: ", num2;
    SiNo
        Escribir num1, " no es multiplo de ", num2;
    FinSi

FinProceso
```

PSeInt - Ejecutando proceso EJERCICIO19

```
*** Ejecución Iniciada. ***
insertar los 2 numeros
> 6
> 9
6 no es multiplo de 9
*** Ejecución Finalizada. ***
```

20. Pedir dos números y decir cuál es el mayor.

```
ejercicio10.psc* ejercicio11.psc* ejercicio12.psc* ejercicio13.psc* ejercicio14.psc ejercicio15.psc* ejerc
1 Proceso ejercicio20
2     //definir las variables
3     Definir num1,num2 Como real;
4     //solicitar numeros por consola
5     Escribir "copiar 2 numeros";
6     Leer num1,num2;
7     //entrada-salida
8     si ( num1 > num2 )Entonces
9         Escribir " el ", num1 , " es mayor que ", num2;
10    FinSi
11
12
13 FinProceso
14
```

PSeInt - Ejecutando proceso EJERCICIO20

```
*** Ejecución Iniciada. ***
copiar 2 numeros
> 6
> 4
el 6 es mayor que 4
*** Ejecución Finalizada. ***
```

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

The screenshot shows the PSeInt IDE with a file explorer at the top containing files from 'ejercicio11.psc' to 'ejercicio16.psc'. The main editor displays the code for 'Proceso ejercicio21'. The code defines two real variables, num1 and num2, and prompts the user to enter two numbers. It then uses conditional logic to compare the numbers and print the result. A console window on the right shows the execution output for 'EJERCICIO21', indicating that the first number (9) is greater than the second (5).

```
Proceso ejercicio21
//damos las variables
Definir num1,num2 Como Real;
//solicitar numeros
Escribir "ingresa los 2 numeros";
Leer num1,num2;
//entrada-salida
si(num1 > num2) Entonces
    Escribir "el ", num1, "es mayor que" , num2;
SiNo
    si (num1=num2) Entonces
        Escribir " los numeros son iguales ";
    SiNo
        Escribir " el ", num1, " es menor que el " , num2;
    FinSi
FinSi
FinProceso
```

PSeInt - Ejecutando proceso EJERCICIO21

```
*** Ejecución Iniciada. ***
ingresa los 2 numeros
> 9
> 5
el9es mayor que5
*** Ejecución Finalizada. ***
```

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

The screenshot shows the PSeInt IDE with a file explorer at the top containing files from 'ejercicio16.psc' to 'ejercicio21.psc'. The main editor displays the code for 'Proceso ejercicio22'. The code defines two real variables, num1 and num2, and prompts the user to insert two numbers. It then uses conditional logic to compare the numbers and print them in descending order. A console window on the right shows the execution output for 'EJERCICIO22', indicating that the first number (5) is less than the second (9), so they are printed as 'numero menor : 5 numero mayor : 9'.

```
Proceso ejercicio22
//definir variables
Definir num1, num2 Como Real;
//solicitar numeros
Escribir "insertar los 2 numeros";
Leer num1,num2;
//entrada-salida
si (num1>num2) Entonces
    Escribir "numero mayor: " , num1, "numero menor" , num2;
SiNo
    si(num1 < num2)Entonces
        Escribir " numero menor : " , num1 , "numero mayor : " , num2;
    SiNo
        si(num1=num2) Entonces
            Escribir "el numero: ", num1 , "es igual que" , num2;
        FinSi
    FinSi
FinSi
FinProceso
```

PSeInt - Ejecutando proceso EJERCICIO22

```
*** Ejecución Iniciada. ***
insertar los 2 numeros
> 5
> 9
numero menor : 5numero mayor : 9
*** Ejecución Finalizada. ***
```

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

```

ejercicio16.psc* | ejercicio17.psc | ejercicio18.psc | ejercicio19.psc | ejercicio20.psc | ejercicio21.psc | ejercicio22.psc | ejercicio23.psc | ejercicio24.psc | ejercicio25.psc | ejercicio26.psc | ejercicio27.psc
5  Escribir "Ingresa los tres numeros que desea organizar";
6  Leer num1, num2, num3;
7  // ENTRADA - SALIDA
8  Si (num1>num2 y num2>num3 )Entonces
9      Escribir "El orden de los numeros es: ", num1, num2, num3;
10 SiNo
11     Si (num2>num1 y num1>num3 )Entonces
12         Escribir "El orden de los numeros es: ", num2, num1, num3;
13     SiNo
14         Si (num3>num2 y num2>num1 )Entonces
15             Escribir "El orden de los numeros es: ", num3, num2, num1;
16         SiNo
17             Si (num1>num3 y num3>num2 )Entonces
18                 Escribir "El orden de los numeros es: ", num1, num3, num2;
19             SiNo
20                 Si (num2>num3 y num3>num1)Entonces
21                     Escribir "El orden de los numeros es: ", num2, num3, num1;
22                 SiNo
23                     Si (num3>num1 y num1>num2)Entonces
24                         Escribir "El orden de los numeros es: ", num3, num1, num2;
25                     FinSi
26                 FinSi
27             FinSi
28         FinSi
29     FinSi
30 FinSi
31 FinProceso
32
PSeInt - Ejecutando proceso EJERCICIO23
*** Ejecución Iniciada. ***
Ingresa los tres numeros que desea organizar
> 9
> 6
> 3
El orden de los numeros es: 963
*** Ejecución Finalizada. ***

```

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

```

ejercicio16.psc* | ejercicio17.psc | ejercicio18.psc | ejercicio19.psc | ejercicio20.psc | ejercicio21.psc | ejercicio22.psc | ejercicio23.psc | ejercicio24.psc | ejercicio25.psc | ejercicio26.psc | ejercicio27.psc
Algoritmo ejercicio24
    // Definicion/declaracion
    Definir num1, cifras como Entero;
    // datos por consola
    Escribir "Ingresa un número entre 0 y 9.999: ";
    Leer num1;
    // EENTRADA - SALIDA
    Si num1 ≥ 0 y num1 ≤ 9 Entonces
        cifras ← 1;
    FinSi

    Si num1 ≥ 10 y num1 ≤ 99 Entonces
        cifras ← 2;
    FinSi

    Si num1 ≥ 100 y num1 ≤ 999 Entonces
        cifras ← 3;
    FinSi

    Si num1 ≥ 1000 y num1 ≤ 9999 Entonces
        cifras ← 4;
    FinSi

    Escribir " tiene ", cifras, " cifras.";
FinAlgoritmo

PSeInt - Ejecutando proceso EJERCICIO24
*** Ejecución Iniciada. ***
Ingresa un número entre 0 y 9.999:
> 5
tiene 1 cifras.
*** Ejecución Finalizada. ***

```

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)

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.

```

ejercicio19.psc | ejercicio20.psc | ejercicio21.psc | ejercicio22.psc | ejercicio23.psc | ejercicio24.psc | ejercicio25.psc
1  .mo ejercicio28
2  //definicion/declaracion
3  inir hora_trabajo, valor_hora como Real;
4  //datos por consola
5  ribir "Ingrese las horas de trabajo";
6  r hora_trabajo;
7  ribir "Ingrese el valor de la hora trabajada";
8  r valor_hora;
9  //ENTRADA - SALIDA
10 (hora_trabajo > 40)Entonces
11     valor_hora ← hora_trabajo * valor_hora;
12     valor_hora ← valor_hora + 10000;
13     Escribir "El salario del trabajador es: ", valor_hora;
14 o
15 Si (hora_trabajo < 40)Entonces
16     valor_hora ← hora_trabajo * valor_hora;
17     Escribir "El salario del trabajador es: ", valor_hora;
18 FinSi
19 Si
20 ritmo
21

```

PSeInt - Ejecutando proceso EJERCICIO28

```

*** Ejecución Iniciada. ***
Ingrese las horas de trabajo
> 24
Ingrese el valor de la hora trabajada
> 2000
El salario del trabajador es: 48000
*** Ejecución Finalizada. ***

```

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%.

```

ejercicio19.psc | ejercicio20.psc | ejercicio21.psc | ejercicio22.psc | ejercicio23.psc | ejercicio24.psc | ejercicio25.psc
1  Algoritmo ejercicio29
2  // Definicion/declaracion
3  Definir monto, descuento, monto_con_descuento como Real;
4  // datos por consola
5  Escribir "Ingrese el monto:";
6  Leer monto;
7  // ENTRADA - SALIDA
8  Si monto > 100 Entonces
9      descuento = monto * 0.1;
10 Sino
11     Si (monto < 100)
12         descuento = monto * 0.02;
13     FinSi
14 FinSi
15 monto_con_descuento = monto - descuento;
16 Escribir "El monto con descuento es:", monto_con_descuento;
17 FinAlgoritmo
18

```

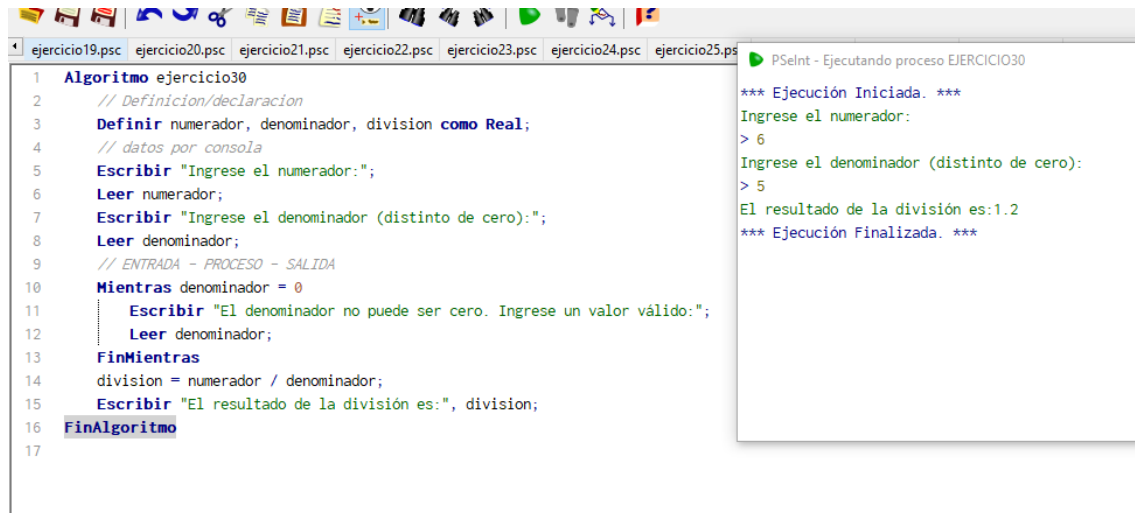
PSeInt - Ejecutando proceso EJERCICIO29

```

*** Ejecución Iniciada. ***
Ingrese el monto:
> 99
El monto con descuento es:97.02
*** Ejecución Finalizada. ***

```

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



The image shows a screenshot of the PSeInt IDE. The top toolbar contains various icons for file operations, editing, and execution. Below the toolbar, a series of tabs represent different exercise files: ejercicio19.psc, ejercicio20.psc, ejercicio21.psc, ejercicio22.psc, ejercicio23.psc, ejercicio24.psc, and ejercicio25.psc. The 'ejercicio23.psc' tab is active, displaying the following Pascal code:

```
1  Algoritmo ejercicio30
2      // Definicion/declaracion
3      Definir numerador, denominador, division como Real;
4      // datos por consola
5      Escribir "Ingrese el numerador:";
6      Leer numerador;
7      Escribir "Ingrese el denominador (distinto de cero):";
8      Leer denominador;
9      // ENTRADA - PROCESO - SALIDA
10     Mientras denominador = 0
11     .....
12         Escribir "El denominador no puede ser cero. Ingrese un valor válido:";
13         Leer denominador;
14     FinMientras
15     division = numerador / denominador;
16     Escribir "El resultado de la división es:", division;
17 FinAlgoritmo
```

To the right of the code editor, a console window titled 'PSeInt - Ejecutando proceso EJERCICIO30' shows the program's execution. The output is as follows:

```
*** Ejecución Iniciada. ***
Ingrese el numerador:
> 6
Ingrese el denominador (distinto de cero):
> 5
El resultado de la división es:1.2
*** Ejecución Finalizada. ***
```