

LISTA DE COTEJO: REPORTE DE PRACTICAS DE PSEINT

NOMBRE DEL DOCENTE: María de los Ángeles Pelayo Vaquero		FIRMA DEL DOCENTE	
DATOS GENERALES DEL PROCESO DE EVALUACIÓN			
NOMBRE DEL ALUMNO: RINCON PEDROZA OMAR YAEL			FIRMA DEL ALUMNO:
PRODUCTO: PRACTICA: ALGORITMOS EN PSEINT	UNIDAD: 2	FECHA: 31-03-2023	PERIODO ESCOLAR: FEBRERO - JULIO 2023

INSTRUCCIONES

Revisar los documentos o actividades que se solicitan y marque en los apartados “SI” cuando la evidencia a evaluar se cumple; en caso contrario marque “NO”. En la columna “OBSERVACIONES” ocúpela cuando tenga que hacer comentarios referentes a lo observado.

Valor del reactivo	Características a cumplir (Reactivo)	CUMPLE		VALOR %
		Si	NO	
5 %	¿Identifico los elementos principales de la práctica?	X		5 %
10 %	¿Identifico y utilizo las herramientas requeridas?	X		10 %
30 %	¿Elaboró y desarrolló un documento de problemas y soluciones que se presentaron durante el desarrollo de la práctica con lenguaje C?	XX		28 %
5 %	¿Identifico componentes teóricos de la unidad?	X		3%
50%	CALIFICACIÓN:	46%		

EVALUACION 50%

ANEXOS



**INSTITUTO TECNOLÓGICO SUPERIOR DE
SAN ANDRÉS TUXTLA**

CARRERA:

Ingeniería industrial

MATERIA:

ALGORITMOS Y PROGRAMACION

NOMBRE DEL ALUMNO:

Omar Yael Rincón Pedroza

46 %

GRUPO:

401B

DOCENTE:

MARIA DE LOS ANGELES PELAYO VAQUERO

SAN ANDRÉS TUXTLA. VER.

FECHA:

12/05/2023

1. Redactar el enunciado del algoritmo

Algoritmo SUMAA

Num1 <-0

Num2 <-0

Escribir "Escribir el numero 1";

Leer Num1;

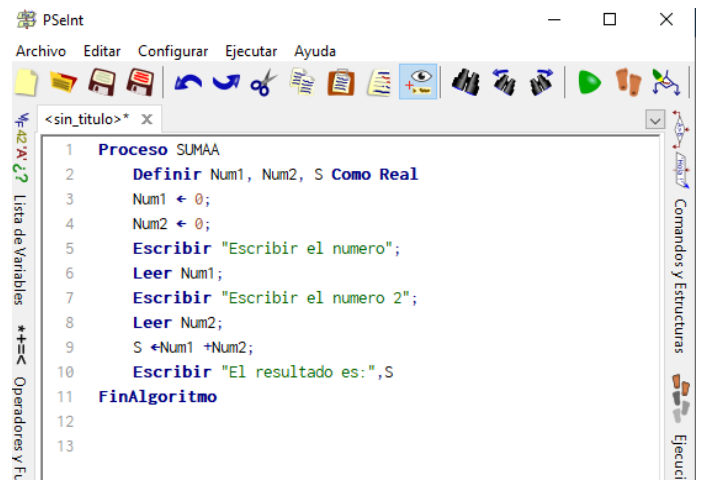
Escribir "Escribir el numero 2";

Leer Num2;

S <- Num1+Num2;

Escribir "El resultado es:" S;

FinAlgoritmo



```
1 Proceso SUMAA
2 Definir Num1, Num2, S Como Real
3 Num1 <- 0;
4 Num2 <- 0;
5 Escribir "Escribir el numero";
6 Leer Num1;
7 Escribir "Escribir el numero 2";
8 Leer Num2;
9 S <-Num1 +Num2;
10 Escribir "El resultado es:",S
11 FinAlgoritmo
12
13
```

PSelnt - Ejecutando proceso SUMAA

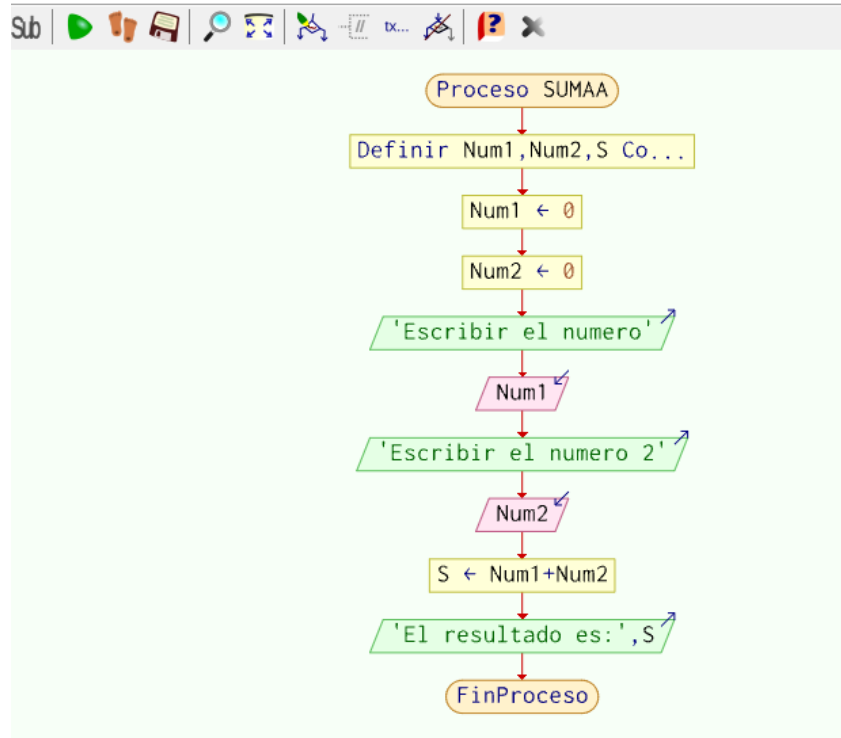
```
*** Ejecución Iniciada. ***
Escribir el numero
> -10
Escribir el numero 2
> 6
El resultado es:-4
*** Ejecución Finalizada. ***
```

No cerrar esta ventana Siempre visible

DESCRIPCION

- Primero se pone el nombre del algoritmo (SUMAA)
- Después se tiene que poner "definir" en las variables y a estas también se les agregara coma dependiendo cuantas sean
- Se especifica la variable "como real"
- Se coloca Num 1 y Num 2 seguido de <-
- Al momento de dar las indicaciones se pone "Escribir"
- Para que sea valido el registro se pondrá la palabra "leer y el nombre"
- Se vuelve a poner lo mismo pero ahora en la variable 2
- En otro espacio se coloca que la suma de ambos y también se coloca <- para indicar operación se utiliza + y sepone " S<- Num1+Num2"
- Seguido se coloca Escribir "El resultado es":S;"
- Para finalizar el algoritmo se coloca "FinAlgoritmo"

PSDraw - SUMAA



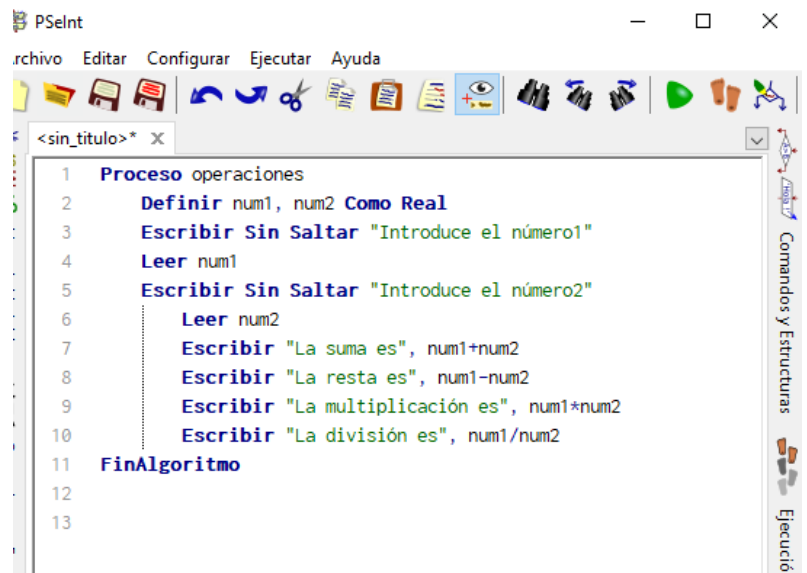
2. Redactar el enunciado del algoritmo

Algoritmo operaciones

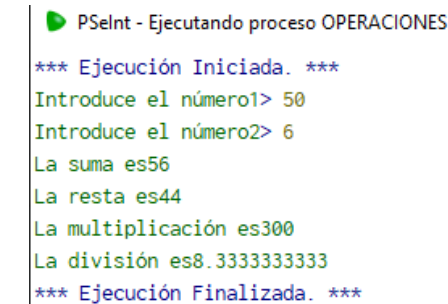
```
Definir num1, num2 Como Real;
Escribir Sin Saltar "Introduce el número1";
Leer num1;
Escribir Sin Saltar "Introduce el número2:";
Leer num2;
Escribir "La suma es", num1+num2;
Escribir "La resta es", num1-num2;
Escribir "La multiplicación es", num1*num2;
Escribir "La división es", num1/num2
FinAlgoritmo
```

DESCRIPCION

- Primero se pone el nombre de (operaciones) en el titulo
- Se colocan las variables poniéndoles como seguido de "como real"
- Se pone " Sin saltar" y también se coloca "introduce el numero 1
- Se coloca " Leer" num 1
- Se vuelve a repetir el paso para el numero 2
- Se escribe " la suma es", num1+num2
- Seguido se coloca "la resta es", num1-num2
- Se escribe "La multiplicacion es",num1*num2
- Se pone "La división es", num1/num2
- Y al finalizar se coloca FinAlgoritmo y se corre

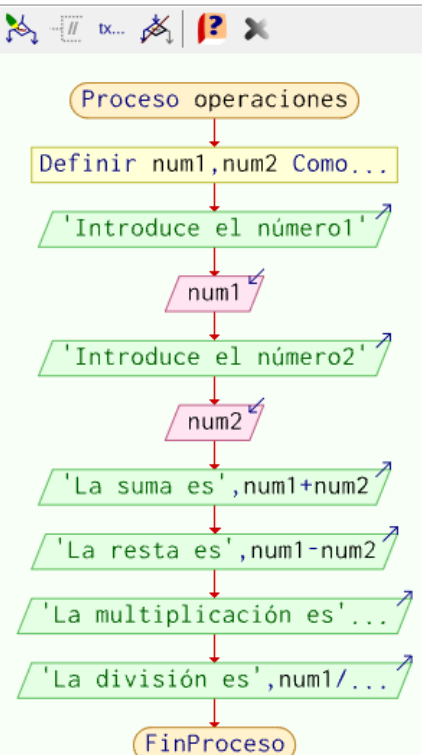


```
1 Proceso operaciones
2   Definir num1, num2 Como Real
3   Escribir Sin Saltar "Introduce el número1"
4   Leer num1
5   Escribir Sin Saltar "Introduce el número2"
6       Leer num2
7       Escribir "La suma es", num1+num2
8       Escribir "La resta es", num1-num2
9       Escribir "La multiplicación es", num1*num2
10      Escribir "La división es", num1/num2
11 FinAlgoritmo
12
13
```



```
PSelnt - Ejecutando proceso OPERACIONES
*** Ejecución Iniciada. ***
Introduce el número1> 50
Introduce el número2> 6
La suma es56
La resta es44
La multiplicación es300
La división es8.3333333333
*** Ejecución Finalizada. ***
```

PSDraw - operaciones



3. Una tienda ofrece un descuento del 15% sobre el total de la compra y un cliente desea saber cuánto deberá pagar finalmente por su compra

Algoritmo CalcularPrecio

Definir precio como Real;
 Escribir "Dime el precio:";
 Leer precio;
 Escribir "Precio final:", precio- precio*0.15;
 FinAlgoritmo

```

1  Algoritmo CalcularPrecio
2      Definir precio como Real;
3      Escribir "Dime el precio:";
4          Leer precio;
5          Escribir "Precio final:", precio- precio*0.15;
6  FinAlgoritmo
7
8
  
```

PSeInt - Ejecutando proceso CALCULARPRECIO

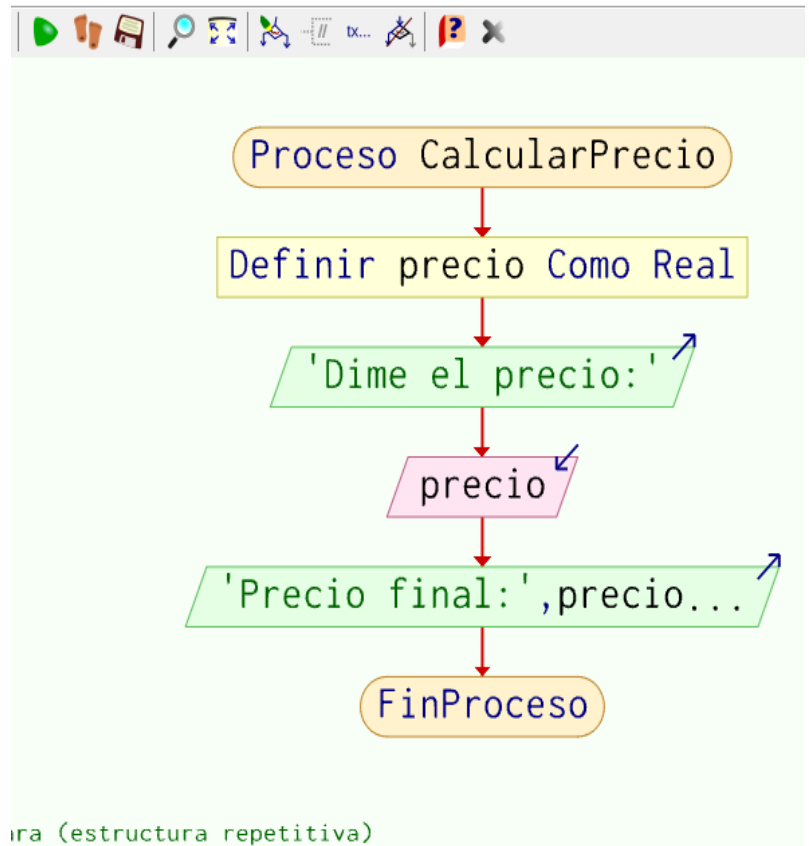
```

*** Ejecución Iniciada. ***
Dime el precio:
> 1600
Precio final:1360
*** Ejecución Finalizada. ***
  
```

DESCRIPCION

- Como primer paso se pone el nombre del algoritmo (CalcularPrecio)
- Después se define el precio "como real"
- Se escribe "dime el precio" y ahí es donde ira la variable
- Después se coloca "Leer precio" para su auto guardado
- Para casi finalizar se agrega en escribir " precio final", precio- precio*0.15; que esto quiere decir que al precio de venta se le restara el 15 por ciento
- Para terminar se agrega "Fin Algoritmo" y se corre

PSDraw - CalcularPrecio



ira (estructura repetitiva)

4. Redactar el enunciado del algoritmo

Crear un algoritmo que nos mida la distancia entre dos números y que sean positivos

Algoritmo CalcularDistancia

Definir num1, num2 como Entero;

Escribir "introducir el número1:";

Leer num1;

Escribir "introducir el número2:";

Leer num2;

Escribir "Distancia:", abs(num1-num2);

FinAlgoritmo

Describir funcionamiento con PSeint (pantallas)

Buscar en el manual PSeint para que se usa abs y explicar

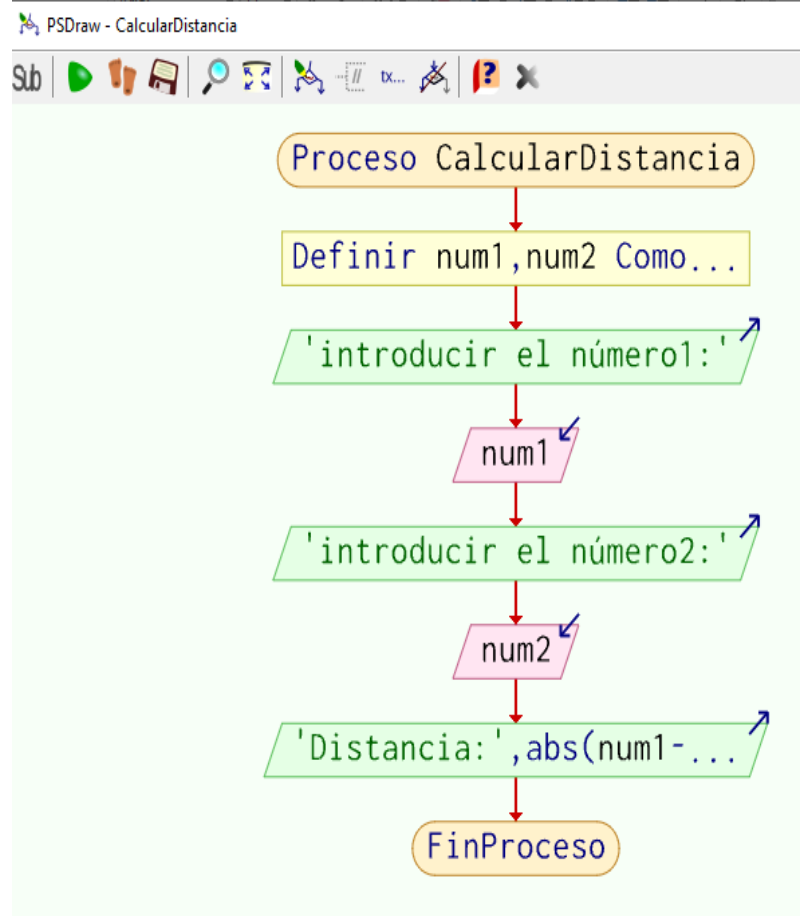
```
<sin_titulo>* X
1 Algoritmo CalcularDistancia
2 Definir num1, num2 como Entero;
3 Escribir "introducir el número1:";
4 Leer num1;
5 Escribir "introducir el número2:";
6 Leer num2;
7 Escribir "Distancia:", abs(num1-num2);
8 FinProceso
9
10
```

PSeint - Ejecutando proceso CALCULARDISTANCIA

```
*** Ejecución Iniciada. ***
introducir el número1:
> 40
introducir el número2:
> 28
Distancia:12
*** Ejecución Finalizada. ***
```

DESCRIPCION

- $ABS(X)$ = Valor Absoluto de X es la que nos dará los positivos después de la resta y obtener un valor absoluto.
- Primero ponemos el nombre (CalcularDistancia)
- Se plasman las variables num1, num2 con las comas
- Se coloca "Escribir" y después "introducir el numero 1!"
- Se le da la orde de leer el numero
- Se repite el proceso para la línea 2
- Para conocer el valor se le agrega a "distancia" el "abs(num1-num2"
- Se cierra con FinAlgoritmo y se corre.



5. Redactar el enunciado del algoritmo

Diseñar el algoritmo que nos dé una hipotenusa de un triángulo solo teniendo sus catetos.

Algoritmo CalcularHipotenusa

Definir cateto1,cateto2,hipotenusa Como Real

Escribir "Introduce el cateto 1:"

Leer cateto1

Escribir "Introduce la cateto 2:"

Leer cateto2

$hipotenusa \leftarrow raiz(cateto1^2 + cateto2^2)$

Escribir "La hipotenusa es" ,hipotenusa

FinAlgoritmo

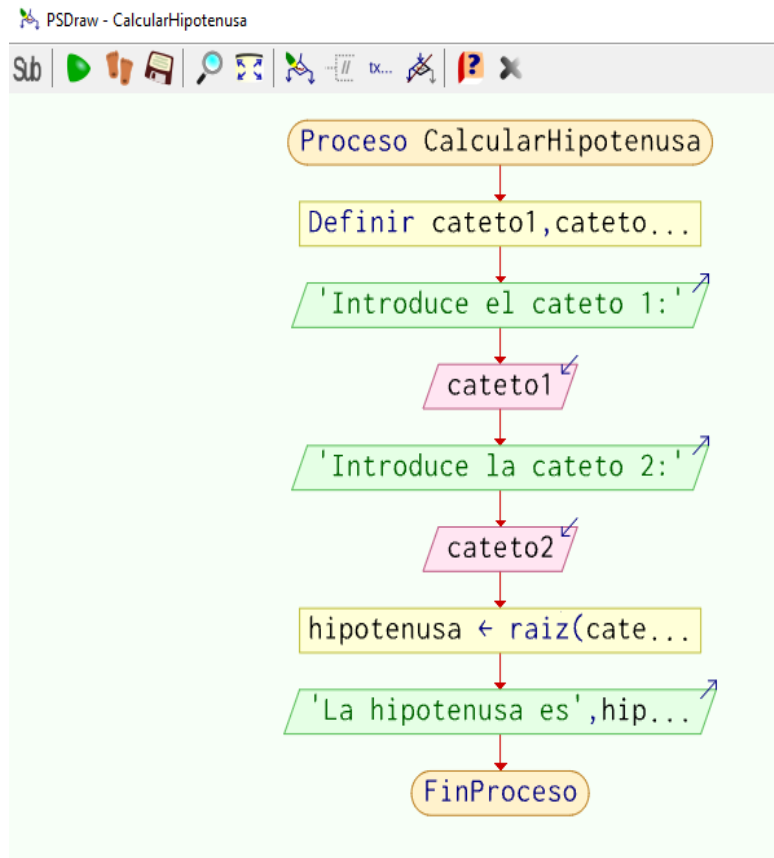
Describir funcionamiento con PSeint (pantallas)

```
PSeint
Archivo Editar Configurar Ejecutar Ayuda
<sin_titulo>* X
1 Algoritmo CalcularHipotenusa
2 Definir cateto1,cateto2,hipotenusa Como Real
3 Escribir "Introduce el cateto 1:"
4 Leer cateto1
5 Escribir "Introduce la cateto 2:"
6 Leer cateto2
7 hipotenusa ← raiz(cateto1 ↑ 2 + cateto2 ↑ 2)
8 Escribir "La hipotenusa es" ,hipotenusa
9 FinAlgoritmo
10
11
```

```
PSeint - Ejecutando proceso CALCULARHIPOTENUSA
*** Ejecución Iniciada. ***
Introduce el cateto 1:
> 50
Introduce la cateto 2:
> 30
La hipotenusa es58.3095189485
*** Ejecución Finalizada. ***
```

DESCRIPCION

- Primero se pone el nombre (CalcularHipotenusa)
- Escribimos definir cateto1,cateto2, agregando la hipotenusa como real
- Luego escribimos "cateto 1" para que lo guarde y después lo lea poniendo " Leer cateto 1"
- Se sigue el mismo paso para el cateto 2
- Luego se busca la hipotenusa poniendo al cuadrado los dos catetos para después sumarlos y sacarles la raíz cuadrada
- Para obtener el valor se escribe "La hipotenusa es",hipotenusa.
- Para cerrarlo se coloca FinAlgoritmo y listo



6. Redactar el enunciado del algoritmo:

A través de una variable auxiliar diseñaremos un pseudocódigo que nos deje intercambiar valores de dos variables.

Algoritmo **IntercambiarVariables**

Definir a,b,aux como Entero;

Escribir "Introduce valor de la variable A:";

Leer a;

Escribir "Introduce valor de la variable B:";

Leer b;

aux <- a;

a <- b;

b <- aux;

Escribir "Nuevo valor de A:", a;

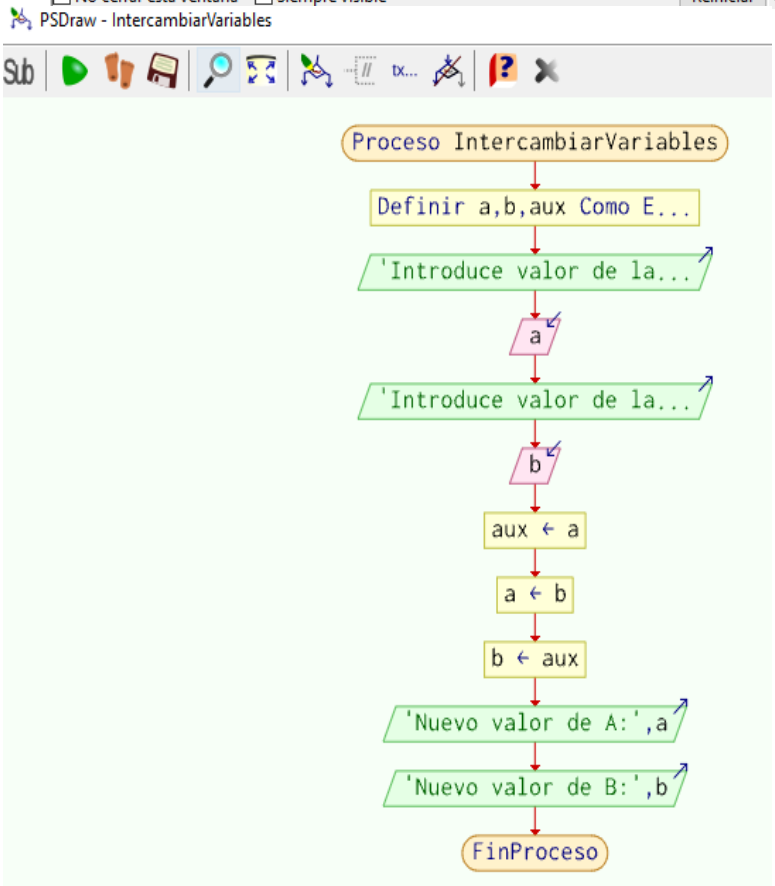
Escribir "Nuevo valor de B:", b;

FinAlgoritmo

Describir funcionamiento con PSeint (pantallas)

```
PSelnt
Archivo  Editar  Configurar  Ejecutar  Ayuda
<sin_titulo>* X
1  Proceso IntercambiarVariables
2  Definir a,b,aux como Entero;
3  Escribir "Introduce valor de la variable A:";
4  Leer a;
5  Escribir "Introduce valor de la variable B:";
6  Leer b;
7  aux <- a;
8  a <- b;
9  b <- aux;
10 Escribir "Nuevo valor de A:", a;
11 Escribir "Nuevo valor de B:", b;
12 FinProceso
13
```

```
PSelnt - Ejecutando proceso INTERCAMBIARVARIABLES
*** Ejecución Iniciada. ***
Introduce valor de la variable A:
> 32
Introduce valor de la variable B:
> 18
Nuevo valor de A:18
Nuevo valor de B:32
*** Ejecución Finalizada. ***
[ ] No cerrar esta ventana [ ] Siempre visible [Reiniciar]
```



DESCRIPCION

- Colocamos el nombre (IntercambiarVariables)
- Definimos las variables que conlleva y se toman como entero
- Escribimos "introducir el valor de la variable A" y siguiente le ordenamos que la Lea
- Se procede lo mismo con la B
- Para que se intercambien los valores "aux" ponemos que la variable a se cambiara con la b y la b con la a: aux=a, a=b y b=aux
- Se colocan los valores que son: escribir "Nuevo valor de A" y "Nuevo valor de B"
- Por último se agrega FinAlgoritmo y se corre

7. Un alumno desea saber cuál será su calificación final en la materia de Algoritmos y programación. Dicha calificación se compone de los siguientes porcentajes:

* 55% del promedio de sus tres calificaciones parciales.

* 30% de la calificación del examen final.

* 15% de la calificación de un trabajo final.

Algoritmo Calcularcalificaciones

Definir cal1, cal2, cal3, examen, trabajo, resultado como Real;

Escribir "Introducir la calificacion uno:";

Leer cal1;

Escribir "Introducir la calificacion dos:";

Leer cal2;

Escribir "Introducir la calificacion tres:";

Leer cal3;

Escribir "Introducir la calificacion del examen:";

Leer examen;

Escribir "Introducir la calificacion del trabajo:";

Leer trabajo;

Resultado $\leftarrow ((cal1 + cal2 + cal3)/3) * 0.55 + 0.3 * examen + 0.15 * trabajo;$

Escribir "Calificacion final:", resultado;

FinAlgoritmo

```

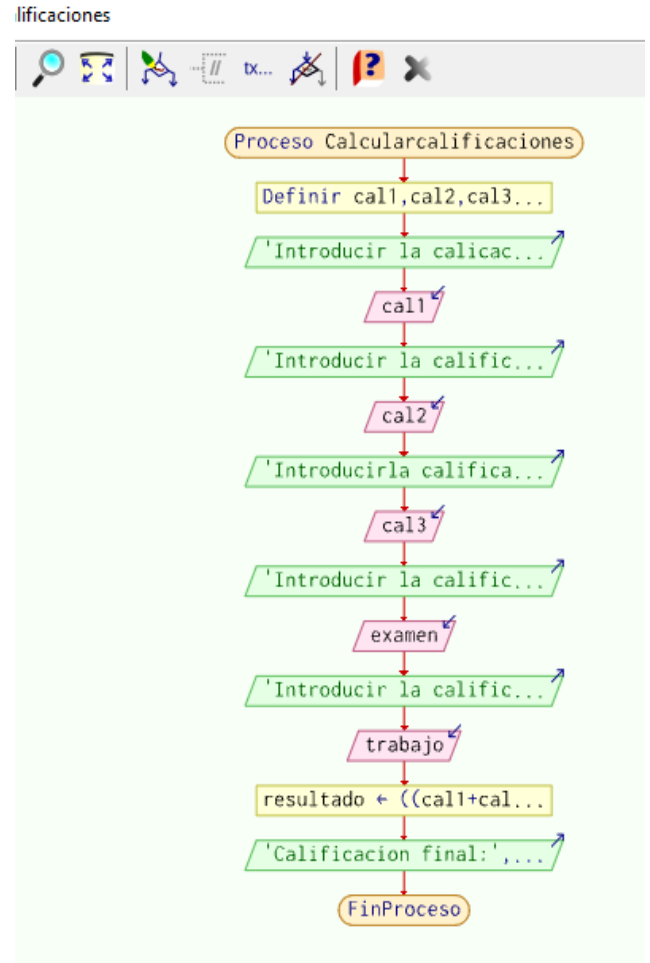
Archivo  Editar  Configurar  Ejecutar  Ayuda
<sin_titulo>* X
1  Proceso Calcularcalificaciones
2  Definir cal1, cal2, cal3, examen, trabajo, resultado como Real;
3  Escribir "Introducir la calificacion uno:";
4  Leer cal1;
5  Escribir "Introducir la calificacion dos:";
6  Leer cal2;
7  Escribir "Introducir la calificacion tres:";
8  Leer cal3;
9  Escribir "Introducir la calificacion del examen:";
10 Leer examen;
11 Escribir "Introducir la calificacion del trabajo:";
12 Leer trabajo;
13 Resultado  $\leftarrow ((cal1 + cal2 + cal3)/3) * 0.55 + 0.3 * examen + 0.15 * trabajo;$ 
14 Escribir "Calificacion final:", resultado;
15 FinProceso
16

```

```

PSeInt - Ejecutando proceso CALCULARCALIFICACIONES
*** Ejecución Iniciada. ***
Introducir la calificacion uno:
> 10
Introducir la calificacion dos:
> 6
Introducir la calificacion tres:
> 8
Introducir la calificacion del examen:
> 8
Introducir la calificacion del trabajo:
> 10
Calificacion final:8.3
*** Ejecución Finalizada. ***
 No cerrar esta ventana  Siempre visible 

```



DESCRIPCION 7

- Ponemos el nombre del algoritmo (Calcularcalificaciones)
- Definimos las variables las cuales son: cal1, cal2, cal3, examen, trabajo y lo definimos como real
- Se coloca el primer valor al escribir "Introducir la calificación uno" y se le ordena "Leer cal1"
- Se repite el proceso para la cal2 y cal3
- Después se pide "introducir la calificación del examen" y que la guarde al poner "Leer examen"
- También colocamos "introducir la calificación del trabajo" Y ordenamos "Leer trabajo"
- Para poder saber el resultado colocamos " Resultado<-
 $((cal1+cal2+cal3)/3)*0.55+0.3examen+0.15*trabajo$ "
- Escribimos " Calificacion final", resultado
- Y para cerrar el algoritmo colocamos FinAlgoritmo

8. Redactar el enunciado del algoritmo

Algoritmo CalcularRaices

Definir num como Entero;

Escribir "Dime el número:";

Leer num;

Escribir "Raíz cuadrada:", raiz(num);

Escribir "Raíz cúbica:", num $^{(1/3)}$;

FinAlgoritmo

Describir funcionamiento con PSeint (pantallas)

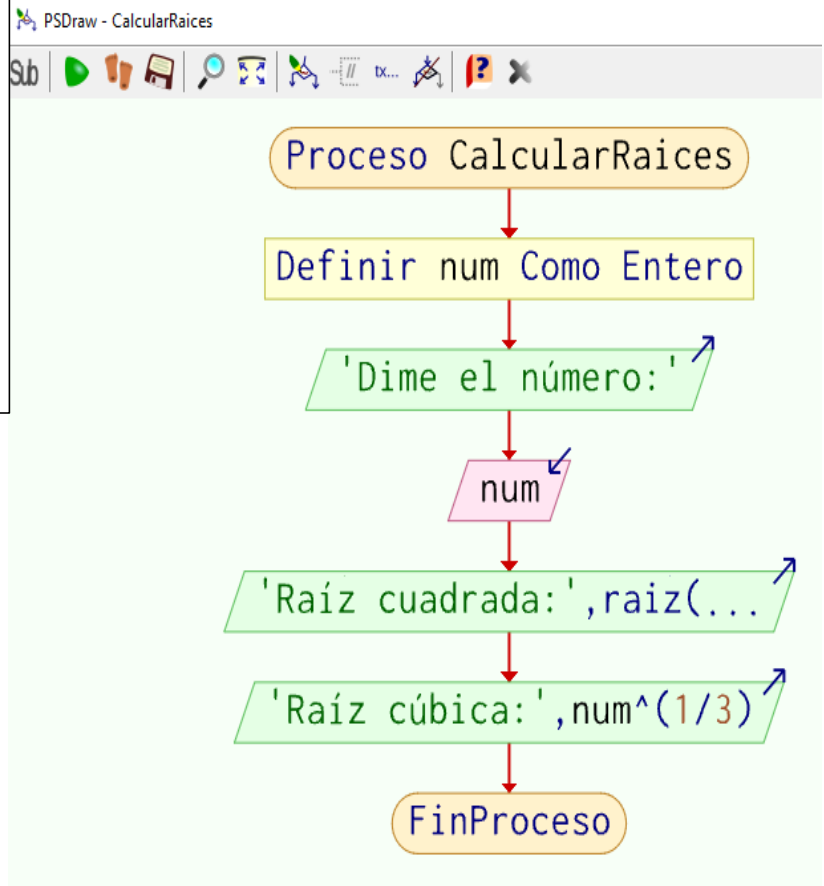
```
Archivo  Editar  Configurar  Ejecutar  Ayuda
<sin_titulo>* X
1  Proceso CalcularRaices
2      Definir num como Entero;
3      Escribir "Dime el número:";
4      Leer num;
5      Escribir "Raíz cuadrada:", raiz(num);
6      Escribir "Raíz cúbica:", num  $^{(1/3)}$ ;
7  FinAlgoritmo
8
9
```

PSelnt - Ejecutando proceso CALCULARRAICES

```
*** Ejecución Iniciada. ***
Dime el número:
> 92
Raíz cuadrada:9.5916630466
Raíz cúbica:4.5143574355
*** Ejecución Finalizada. ***
```

DESCRIPCION

- Primero se coloca el nombre del algoritmo(CalcularRaices)
- Definimos las variables como numero entero
- Ocupamos el primer valor por lo cual ponemos escribir "Dime el numero"
- Lo guardamos en "Leer num"
- Colocamos la "Raíz cuadrada",raíz(num) la cual da por automatico y se le agrega otra "Raíz cubica", $^{(1/3)}$ esto por que asi indican los índices.
- Para terminar con el algoritmo le ponemos "FinalAlgoritmo" y corremos



9. Crear un algoritmo que a partir de un número ingresado diga si el mismo es par o impar

Algoritmo Par_impNumeros

Escribir "Escribe un numero";

Leer nro;

Si $(nro \bmod 2) = 0$ entonces

Escribir "es par";

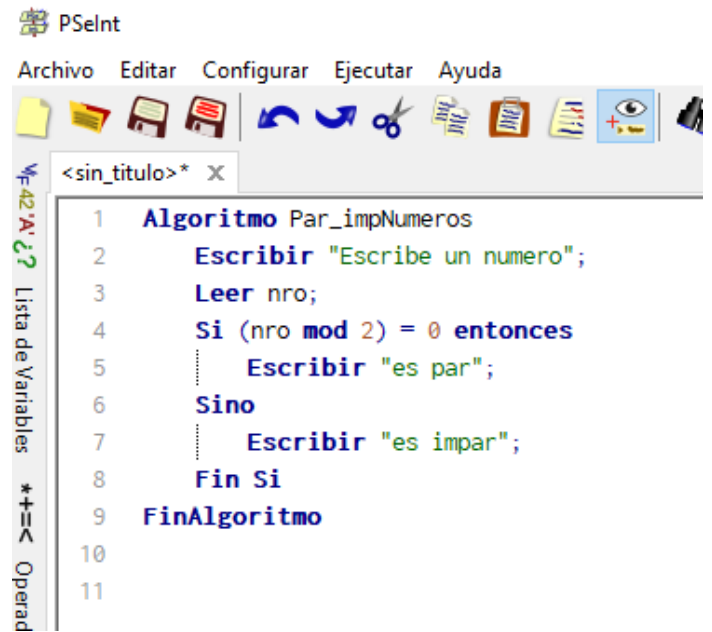
Sino

Escribir "es impar";

Fin Si

FinAlgoritmo

Describir funcionamiento con PSeint (pantallas)



```
1 Algoritmo Par_impNumeros
2   Escribir "Escribe un numero";
3   Leer nro;
4   Si (nro mod 2) = 0 entonces
5       Escribir "es par";
6   Sino
7       Escribir "es impar";
8   Fin Si
9 FinAlgoritmo
10
11
```

PSeint - Ejecutando proceso PAR_IMPNUMEROS

*** Ejecución Iniciada. ***

Escribe un numero

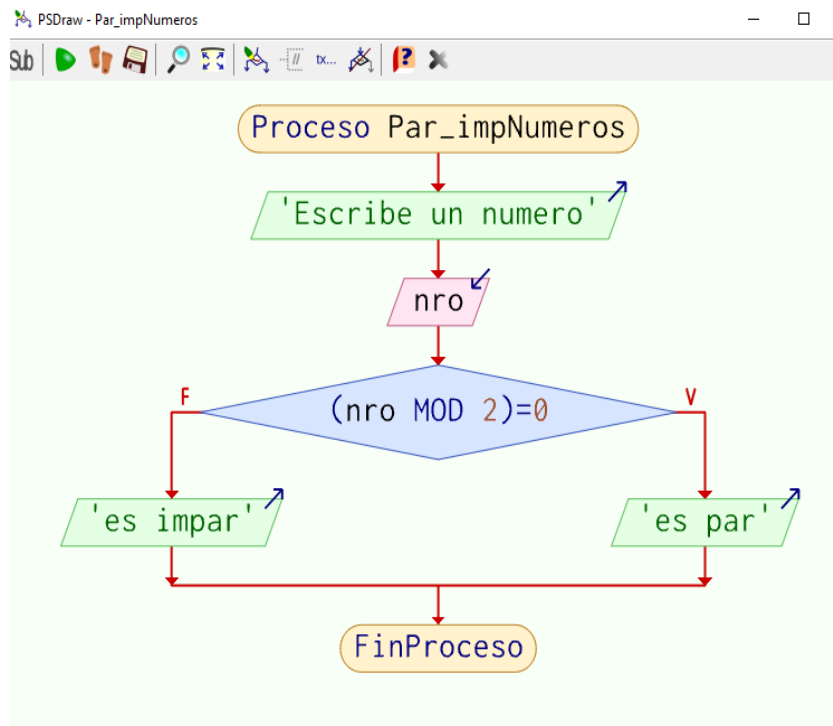
> 7

es impar

*** Ejecución Finalizada. ***

DESCRIPCION

- Se coloca el nombre del algoritmo(Par_impNumeros)
- Se escriben las variables la cual es "Escribe un numero"
- Lo guardamos en "Leer nro"
- Agregamos la indicación SI la condición que ponemos y al dividirlo entre 2 nos tiene que dar 0
- Aclaremos que el "mod" es para que el resultado de dividir se devuelva ejemplo "si $n \bmod 2 = 0$ "
- La condición se cierra con FinSi
- Y para terminar el algoritmo lo cerramos con FinAlgoritmo y corremos



10. Redactar el enunciado del algoritmo

Elaborar un algoritmo que que al momento de ingresar un numero nos diga si es mayor, menor o igual a 9

Algoritmo MayoresIgualesMenoresA9

N <- 0

Escribir "Escribir el numero"

Leer N

Si N Es Igual A 9 Entonces

Escribir "El numero es igual a 9"

Sino

Si N Es Mayor Que 9 Entonces

Escribir "El numero es mayor a 9"

Sino

Escribir "El numero es menor a 9"

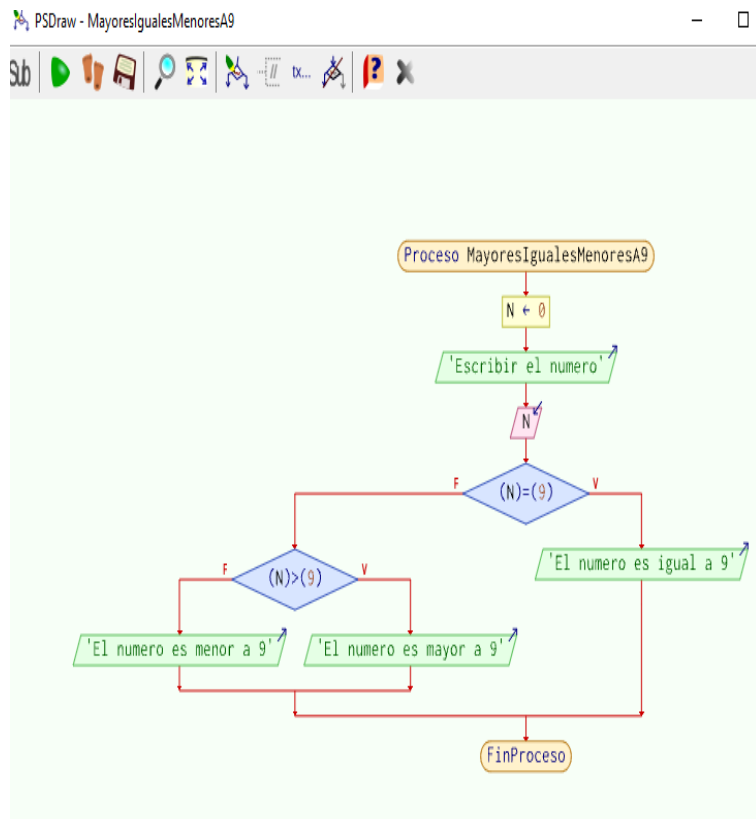
Fin Si

Fin Si

FinAlgoritmo

```
Archivo  Editar  Configurar  Ejecutar  Ayuda
<sin_titulo>* X
1  Algoritmo  MayoresIgualesMenoresA9
2  N <- 0
3  Escribir "Escribir el numero"
4  Leer N
5  Si N Es Igual A 9 Entonces
6  ..... Escribir "El numero es igual a 9"
7  Sino
8  ..... Si N Es Mayor Que 9 Entonces
9  ..... Escribir "El numero es mayor a 9"
10 ..... Sino
11 ..... Escribir "El numero es menor a 9"
12 ..... Fin Si
13 Fin Si
14 FinAlgoritmo
15
16
```

```
PSInt - Ejecutando proceso MAYORESIGUALESMENORES9
*** Ejecución Iniciada. ***
Escribir el numero
> 15
El numero es mayor a 9
*** Ejecución Finalizada. ***
```



DESCRIPCION

- Comenzamos con agregar el nombre del algoritmo (MayoresIgualesMenoresA9)
- Se expresan las variables (N)
- Se pide "Escribir el numero" y guardarlo en "Leer N"
- "Si N es igual a 9 entonces" pedimos "escribir el numero mayor a 9"
- Si no es asi pedimos "escribir el numero es menor a 9"
- Cerramos la condición con el FinSi
- Y terminamos el algoritmo con "FinAlgoritmo" Y se puede correr

11. Determinar si un alumno aprueba o reprueba un curso, sabiendo que aprobará si su promedio de tres calificaciones es mayor o igual a 70; reprueba en caso contrario. Deberá permitir ingresar las tres calificaciones y luego calcular su promedio.

Algoritmo aprueba_reprueba

Escribir "Ingrese calificación 1"

Leer Cal1

Escribir "Ingrese calificación 2"

Leer Cal2

Escribir "Ingrese calificación 3"

Leer Cal3

$Prom \leftarrow (Cal1 + Cal2 + Cal3) / 3$

Si $Prom \geq 70$ Entonces

Escribir "Aprueba"

Sino

Escribir "Reprueba"

Fin Si

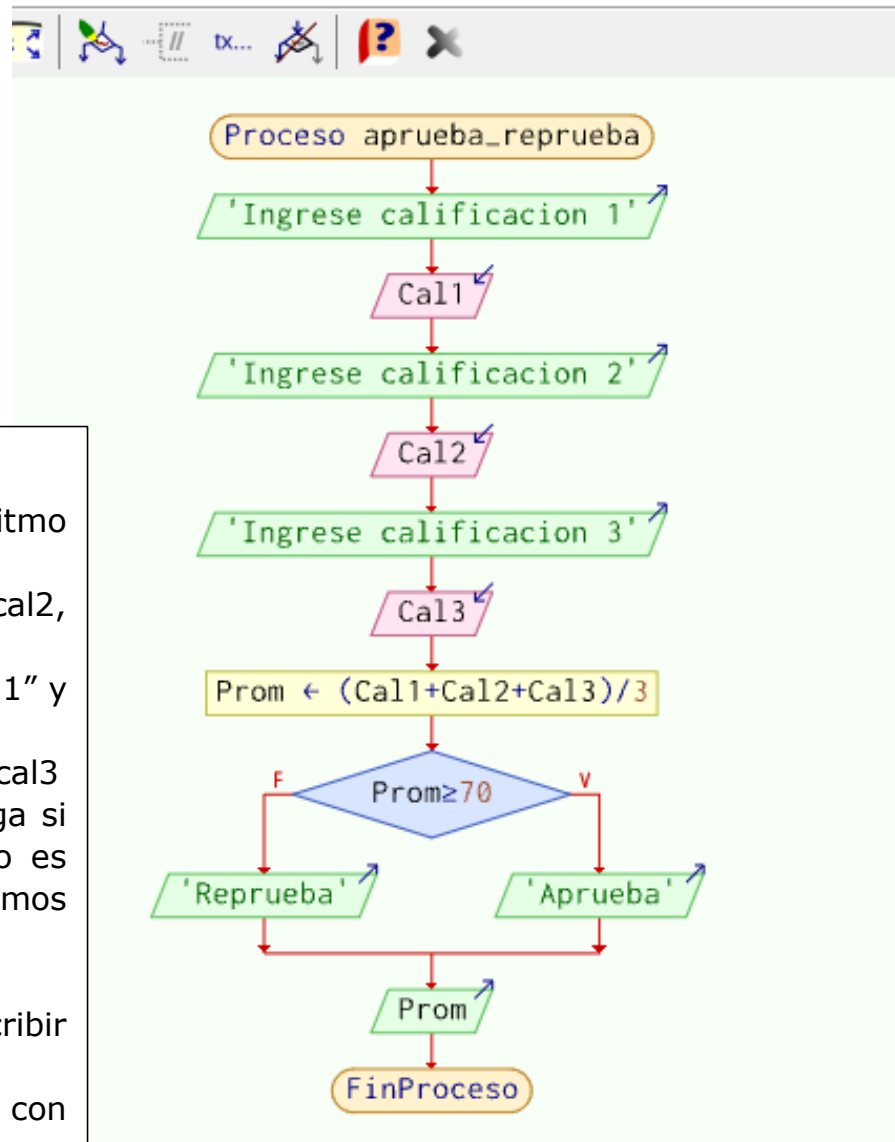
Escribir Prom

FinAlgoritmo

```

Archivo  Editar  Configurar  Ejecutar  Ayuda
<sin_titulo>* X
1  Algoritmo aprueba_reprueba
2  Escribir "Ingrese calificación 1"
3  Leer Cal1
4  Escribir "Ingrese calificación 2"
5  Leer Cal2
6  Escribir "Ingrese calificación 3"
7  Leer Cal3
8  Prom←(Cal1+Cal2+Cal3)/3
9  Si Prom≥70 Entonces
10     Escribir "Aprueba"
11  Sino
12     Escribir "Reprueba"
13  Fin Si
14  Escribir Prom
15  FinAlgoritmo
16
PSeInt - Ejecutando proceso APRUEBA_REPRUEBA
*** Ejecución Iniciada. ***
Ingrese calificación 1
> 92
Ingrese calificación 2
> 79
Ingrese calificación 3
> 84
Aprueba
85
*** Ejecución Finalizada. ***

```



DESCRIPCION

- Le asignamos el nombre al algoritmo (aprueba_reprueba)
- Se colocan la variables: cal1, cal2, cal3
- Se ordena "Escribir la calificación 1" y se guarda en "Leer cal1"
- Se repiten los pasos para la cal2, cal3
- Se busca una función que nos diga si tiene un promedio mayor a 70 o es igual a este, por lo que agregamos "Aprueba" o "Reprueba"
- La condición se cierra con FinSi
- Pedimos el promedio en "Escribir Prom"
- Y cerramos el algoritmo con FinAlgoritmo

12. Calcular el área de un triángulo, deberá desarrollar el algoritmo y ejecutarlo en PSeint, describiendo el funcionamiento

Algoritmo CalcularAreaTriangulo

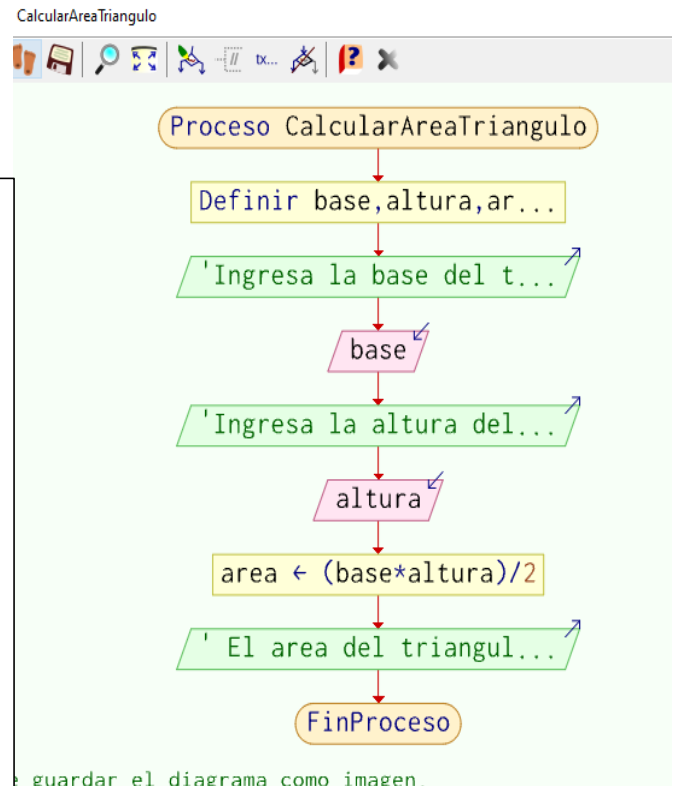
Definir base, altura, area como real
 Escribir "Ingresa la base del triangulo:"
 Leer base;
 Escribir "Ingresa la altura del triangulo:"
 Leer altura;
 $area \leftarrow (base * altura) / 2;$
 Escribir " El area del triangulo es:", área;
 FinAlgoritmo

```

Archivo  Editar  Configurar  Ejecutar  Ayuda
<sin_titulo>* X
1  Algoritmo  CalcularAreaTriangulo
2  Definir  base, altura, area como real
3  Escribir  "Ingresa la base del triangulo:"
4  Leer  base;
5  Escribir  "Ingresa la altura del triangulo:"
6  Leer  altura;
7  area ← (base*altura) / 2;
8  Escribir  " El area del triangulo es:", area;
9  FinAlgoritmo
10
11
  
```

```

PSeint - Ejecutando proceso CALCULARAREATRIANGULO
*** Ejecución Iniciada. ***
Ingresa la base del triangulo:
> 22
Ingresa la altura del triangulo:
> 15
El area del triangulo es:165
*** Ejecución Finalizada. ***
  
```



DESCRIPCION

- Como primer paso se agrega el nombre del algoritmo (CalcularAreaTriangulo)
- Colocamos las variables: base, altura, área como real
- Se requiere escribir "Ingrese la base del triangulo"
- Se guarda con "Leer base;"
- Se pide el valor de altura con Escribir " ingrese la altura del triangulo"
- Se guarda con "Leer altura;"
- Para sacar el área se debe colocar "área<- (base*altura) / 2 ya que es la formula para calcular área del triangulo
- Se pone " Escribir el área del triangulo es:" área;
- Y para cerrar el algoritmo le ponemos "FinAlgoritmo

EVALUACION UNIDAD 2.

44 %

MATERIA: ALGORITMOS Y LENGUAJE DE PROGRAMACIÓN INGENIERÍA INDUSTRIAL

Se ha registrado el correo del encuestado (211u0113@alumno.itssat.edu.mx) al enviar este formulario.

NOMBRE Y APELLIDOS *

OMAR Yael RINCON PEDROZA

GRUPO: *

401B

Diseño descendente (top-down): el problema se descompone en etapas o estructuras jerárquicas.

* 1 punto

V

F



Análisis del problema: esta fase requiere de una clara definición, donde se contemple exactamente lo que debe hacer el programa y el resultado o la solución deseada, entonces es necesario definir:

* 2 puntos

Los datos de entrada (tipo y cantidad).

Cuál es la salida deseada (tipo y cantidad).

Los métodos y las fórmulas que se necesitan para procesar los datos.

V



F

Diseño del código: en esta etapa se determina cómo se hace el programa. Este procedimiento es independiente del lenguaje de programación. Las herramientas son el diagrama de flujo y el pseudocódigo.

* 2 puntos

V



F

Recursos abstractos (simplicidad): consiste en descomponer las acciones complejas en otras más simples capaces de ser resueltas con mayor facilidad.

* 2 puntos

V

F



El diagrama de flujo son las instrucciones que se representan por medio de frases o proposiciones en español que facilitan tanto la escritura como la lectura de programas.

* 2 puntos

V



F

La definición del algoritmo es: una secuencia no ambigua, finita y ordenada de pasos para poder resolver un problema.

* 1 punto

V



F

Características de un buen algoritmo:

* 5 puntos

Debe tener un punto particular de inicio (programa principal). El módulo de nivel más alto que llama a (subprogramas) módulos de nivel más bajos.

Debe ser definido, no debe permitir dobles interpretaciones.

Debe ser general, es decir, soportar la mayoría de las variantes que puedan presentarse en la definición del problema.



Debe ser finito en tamaño y tiempo de ejecución.

V

F

Datos de entrada: un código tiene cero o más entradas, es decir cantidades que le son dadas antes de que el algoritmo comience, o dinámicamente mientras el algoritmo corre. * 2 puntos

V

F



Procesamiento de datos: aquí incluye operaciones aritmético-lógicas, selectivas y repetitivas; cuyo objetivo es obtener la solución del problema. * 2 puntos

V

F



Pseudocódigo: es la operación de escribir la solución del problema (de acuerdo a la lógica del diagrama de flujo o pseudocódigo), en una serie de instrucciones detalladas, en un código reconocible por la computadora. * 2 puntos

V

F



El diseño del algoritmo es independiente del lenguaje de programación, este puede ser usado para cualquier lenguaje de programación. * 2 puntos

V

F



El pseudocódigo está diseñado para facilitar su comprensión, en lugar de la lectura mediante la computadora.

* 2 puntos

También se utiliza en la planificación del desarrollo de programas informáticos, para esquematizar la estructura del programa antes de realizar la efectiva codificación

V

F



Un objeto es una variable cuando su valor puede modificarse y además posee un nombre que lo identifica y un tipo que describe su uso.

* 2 puntos

V

F



Cuando definimos una variable, creamos un identificador (nombre de la variable) que hace referencia a un lugar de la memoria donde se almacena un dato. La diferencia respecto de la definición de una constante, es que en el momento de su creación el valor del objeto es desconocido, mientras que para una constante no solo es conocido, sino que permanece inalterado durante la ejecución del procedimiento resolvente.

* 2 puntos

V

F



Los nombres o etiquetas de las variables y las constantes siempre deben empezar con una letra (mayúscula o minúscula) y no pueden contener espacios en blanco, si usamos más de un carácter para su identificación empezamos con la letra y luego podemos seguir con números o letras. * 2 puntos

V

F



Estructuras secuenciales: cada acción sigue a otra acción secuencialmente. La salida de una acción es la entrada de otra. * 2 puntos

V

F



Estructuras selectivas: en estas estructuras se evalúan las condiciones y en función de sus resultados se realizan unas acciones u otras. Se utilizan expresiones lógicas. * 2 puntos

V

F



DESARROLLAR EL ALGORITMO: Hallar el perímetro y el área de un rectángulo ingresando sus lados

* 5 puntos

Algoritmo CalcularElPerimetroAreaRectangulo

Definir ladoA, ladoB, perimetro, area como Real;

Escribir "Ingrese la longitud del lado a: "

Leer ladoA

Escribir "Ingrese la longitud del lado b: "

Leer ladoB

perimetro <- (2 * ladoA) + (2 * ladoB)

area <- ladoA * ladoB

Escribir "El perímetro del rectángulo es: ", perimetro

Escribir "El área del rectángulo es: ", area

FinAlgoritmo



En PSeInt se hace referencia a una variable mediante un identificador (el nombre de la variable). Un identificador debe comenzar con letras, y puede contener solo letras, números y el guión bajo. No puede contener ni espacios ni operadores, ni coincidir con una palabra reservada o función del lenguaje, para no generar ambigüedad. Ejemplos de identificadores válidos son: A, B, C, Lado1, Total, Nombre_y_Apellido, DireccionCorreo, ... En la mayoría de los lenguajes reales los nombres de variables no pueden contener acentos, ni diéresis, ni eñes. En PSeInt, esto se permite, dependiendo del perfil de lenguaje seleccionado.

* 2 puntos

V

F

Existen tres tipos de datos básicos:

* 2 puntos

Numérico: números, tanto enteros como reales. Para separar decimales se utiliza el punto. Ejemplos: 12 23 0 -2.3 3.14

Lógico: solo puede tomar dos valores: VERDADERO o FALSO.

Carácter: caracteres o cadenas de caracteres encerrados entre comillas (pueden ser dobles o simples). Ejemplos 'hola' "hola mundo" '123' 'FALSO' 'etc'

Los tipos de datos simples se determinan automáticamente cuando se crean las variables. Las dos acciones que pueden crear una variable son la lectura(LEER) y la asignación(<-). Por ejemplo, la asignación "A<-0;" está indicando implícitamente que la variable A será una variable numérica. Una vez determinado el tipo de dato, deberá permanecer constante durante toda la ejecución del proceso; en caso contrario el proceso será interrumpido.

V

F

La instrucción definir permite explicitar el tipo de una o más variables. Esta definición puede ser opcional u obligatoria dependiendo de la configuración del lenguaje. La sintaxis es:

* 2 puntos

Definir <var1> , <var2> , ... , <varN> Como [REAL/ENTERO/LOGICO/CARACTER];

Una variable debe definirse antes de ser utilizada por primera vez. Los arreglos, se definen utilizando su identificador (sin subíndices ni dimensiones) antes o después de dimensionarlos, y el tipo aplica para todos los elementos del mismo (ya que se trata de una estructura de datos homogénea).

Los tipos posibles son NUMERO, NUMERICO, REAL, ENTERO, LOGICO, CARACTER, TEXTO, CADENA

V

F



Todo algoritmo en pseudocódigo tiene la siguiente estructura general:

* 2 puntos

Algoritmo SinTitulo

acción 1;

acción 1;

.

.

.

acción n;

FinAlgoritmo



Comienza con la palabra clave Algoritmo(o alternatively Proceso, son sinónimos) seguida del nombre del programa, luego le sigue una secuencia de instrucciones y finaliza con la palabra Algoritmo(o Proceso). Una secuencia de instrucciones es una lista de una o más instrucciones y/o estructuras de control.

V

F

La instrucción Escribir permite mostrar valores al ambiente.

* 2 puntos

Escribir <expr1> , <expr2> , ... , <exprN> ;

Esta instrucción informa al ambiente (en este caso escribiendo en pantalla) los valores obtenidos de evaluar N expresiones. Dado que puede incluir una o más expresiones, mostrará uno o más valores. Si hay más de una expresión, se escriben una a continuación de la otra sin separación, por lo que el algoritmo debe explicitar los espacios necesarios para diferenciar dos resultados si así lo requiere.



Si en algún punto de la línea se encuentran las palabras clave "SIN SALTAR" o "SIN BAJAR" los valores se muestran en la pantalla, pero no se avanza a la línea siguiente, de modo que la próxima acción de lectura o escritura continuará en la misma línea. En caso contrario, se añade un salto de línea luego de las expresiones mostradas.

Escribir Sin Saltar <expr1> , ... , <exprN>;

Escribir <expr1> , ... , <exprN> Sin Saltar;

Puede utilizarse indistintamente las palabras Imprimir y Mostrar en lugar de Escribir si su perfil de lenguaje permite sintaxis flexible. Además, en este caso se permite opcionalmente separar las expresiones a mostrar simplemente con espacios en lugar de comas.

V

F

Este formulario se creó en INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA.

Google Formularios