

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: CHAPOL PONCIANO ROSA ADELA			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.

<i>Valor del reactivo</i>	<i>Características a cumplir (Reactivo)</i>	CUMPLE		VALOR %
		<i>Si</i>	<i>NO</i>	
5 %	¿Identifico los elementos principales de la práctica?	X		5 %
10 %	¿Identifico y utilizo las herramientas requeridas?	X		9 %
30 %	¿Elaboró y desarrolló un documento de problemas y soluciones que se presentaron durante el desarrollo de la práctica con lenguaje C?	XX		25 %
5 %	¿Identifico componentes teóricos de la unidad?	X		3%
50%	CALIFICACIÓN:	43%		

EVALUACION 50%

ANEXOS

The background features a white page with three blue circles of varying sizes arranged vertically. Each circle is composed of three concentric layers: a dark blue inner circle, a medium blue middle ring, and a light blue outer ring. Two thin, light blue lines intersect at the top center, forming a 'V' shape that frames the circles. The text is centered horizontally and positioned between the middle and bottom circles.

REPORTE DE PRACTICA PSeInt



INSTITUTO TECNOLÓGICO SUPERIOR DE SAN ANDRÉS TUXTLA

INGENIERÍA INDUSTRIAL



MATERIA:

Algoritmos y Lenguaje de Programación

43%

ACTIVIDAD:

Practica PSeInt

ALUMNA:

Rosa Isela Chapol Ponciano

DOCENTE:

María de los Ángeles Pelayo Vaquero

GRUPO:

401° "A"

San Andrés Tuxtla Ver. A 9 de Mayo del 2023

PRACTICA 1

Un vendedor recibe un sueldo base más un 10% extra por comisión de sus ventas, el vendedor desea saber cuánto dinero obtendrá por concepto de comisiones por las tres ventas que realiza en el mes y el total que recibirá en el mes tomando en cuenta su sueldo base y comisiones.

The screenshot displays three windows from a PSeInt IDE:

- Code Editor:** Shows the algorithm for calculating the salary. It defines variables for base salary, three sales prices, and commission. It prompts the user for each value, calculates the commission as 10% of the sum of sales, and outputs the total salary.
- Terminal:** Shows the execution process. It displays the prompts and user inputs: base salary (2500), sales prices (7000, 8750.50, 9798.78), and the resulting commission (2554.928) and total salary (5054.928).
- Flowchart:** A vertical flowchart titled 'Algoritmo CalcularSueldo'. It starts with 'Definir sueldo_base, ve...', followed by three input steps for sales prices, three assignment steps for the sales values, a calculation step for 'comision', an output step for 'Comisión por ventas', another output step for 'Sueldo total', and ends with 'FinAlgoritmo'.

At the bottom of the screen, the Windows taskbar shows the time as 04:00 p. m. on 08/05/2023.

```
Algoritmo CalcularSueldo
Definir sueldo_base, venta1, venta2,
venta3, comision como Real;
Escribir "Dime el sueldo base:";
Leer sueldo_base;
Escribir "Dime precio de la venta 1:";
Leer venta1;
Escribir "Dime precio de la venta 2:";
Leer venta2;
Escribir "Dime precio de la venta 3:";
Leer venta3; comision<-
venta1*0.1+venta2*0.1+venta3*0.1;
Escribir "Comisión por ventas:",comision;
Escribir "Sueldo total:";
```

En este caso nuestro algoritmo como dato de entrada es el Calcular sueldo y este se almacena en la variable 1 "sueldo base", para después pasar a la venta 1, 2 y 3 y al final de esto se suman las ventas y se multiplica por 0.1 que son las comisiones y es así como nos da el resultado final de este algoritmo.

En el diagrama que se muestra en este ejercicio nos indica cómo se van realizando cada paso de acuerdo al sueldo que se va a calcular más las comisiones, hasta el sueldo total y al final obtener el resultado

PRACTICA 2

Diseñar un algoritmo que pida por teclado tres números; si el primero es negativo, debe imprimir la multiplicación de los tres y si no lo es, imprimirá la suma

The screenshot displays a Windows desktop with three windows. The left window, titled 'PSeInt', shows the source code for an algorithm named 'tresnumeros'. The code prompts for three numbers, reads them into variables Num1, Num2, and Num3, and then uses a conditional statement to calculate the result: multiplication if Num1 is negative, and addition otherwise. The middle window shows the program's execution with the following input and output:

```
*** Ejecución Iniciada. ***
Ingrese numero 1
> -10
Ingrese numero 2
> 9
Ingrese numero 3
> 5
*** Ejecución Finalizada. ***
```

The right window, titled 'PSDraw - tresnumeros', contains a flowchart that visualizes the program's logic. It starts with an oval 'Algoritmo tresnumeros', followed by three parallelogram input boxes for 'Ingrese numero 1', 'Ingrese numero 2', and 'Ingrese numero 3'. These lead to three pink parallelogram assignment boxes for 'Num1', 'Num2', and 'Num3'. A blue diamond decision box 'Num1 < 0' follows. The 'F' (False) path leads to a yellow parallelogram output box 'Resul ← Num1+Num2+Num3'. The 'V' (True) path leads to a yellow parallelogram output box 'Resul ← Num1*Num2*Num3'. Both paths merge and lead to a green parallelogram output box 'Resul', which finally leads to an oval 'FinAlgoritmo'.

```
Algoritmo tresnumeros
Escribir "Ingrese numero 1"
Leer Num1
Escribir "Ingrese numero 2"
Leer Num2
Escribir "Ingrese numero 3"
Leer Num3
Si Num1<0 Entonces
  Resul<-Num1 * Num2 * Num3
Sino
  Resul<-Num1+Num2+Num3
Fin Si
Escribir Resul
FinAlgoritmo
```

En este caso nuestro algoritmo como dato de entrada es el escribir tres números y este se almacena en la variable 1 "numero 1" para después pasar al numero 2 y 3 y al final de esto se suman esos números y es así como nos da la suma como el resultado final de este algoritmo.

En el diagrama que se muestra en este ejercicio nos indica cómo se van realizando cada paso de acuerdo a los 3 números que se van a sumar hasta el resultado total y al final obtener el resultado

PRACTICA 3

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.

The screenshot displays three windows from a software development environment:

- PSelnt (Left):** Shows the pseudocode for the algorithm. It starts with 'Algoritmo aprueba_reprueba', followed by prompts for three grades (Cal1, Cal2, Cal3), calculation of the average (Prom), and a conditional statement: 'Si Prom >= 70 Entonces Escribir "Aprueba" Sino Escribir "Reprueba" Fin Si Escribir Prom FinAlgoritmo'.
- PSelnt - Ejecutando proceso APRUEBA_REPRUEBA (Middle):** Shows the execution output. It displays the prompts and user inputs: 'Ingrese calificacion 1' (50), 'Ingrese calificacion 2' (60), 'Ingrese calificacion 3' (70). The output shows 'Reprueba' and '60' (likely the average), followed by '*** Ejecución Finalizada. ***'.
- PSDraw - aprueba_reprueba (Right):** Shows a flowchart that visualizes the pseudocode. It starts with 'Algoritmo aprueba_reprueba', followed by three input boxes for 'Ingrese calificacion 1', 'Ingrese calificacion 2', and 'Ingrese calificacion 3'. These lead to variables 'Cal1', 'Cal2', and 'Cal3'. The flowchart then calculates 'Prom ← (Cal1+Cal2+Cal3)/3', followed by a decision diamond 'Prom >= 70'. If 'F' (False), it outputs 'Reprueba'; if 'V' (True), it outputs 'Aprueba'. Both paths lead to 'Prom' and finally 'FinAlgoritmo'.

```
Algoritmo aprueba_reprueba
Escribir "Ingrese calificacion 1"
Leer Cal1
Escribir "Ingrese calificacion 2"
Leer Cal2
Escribir "Ingrese calificacion 3"
Leer Cal3
Prom ← (Cal1+Cal2+Cal3)/3
Si Prom >= 70 Entonces
Escribir "Aprueba"
Sino
Escribir "Reprueba"
Fin Si
Escribir Prom
FinAlgoritmo
```

Nuestro algoritmo principal y como dato de entrada es el escribir "Aprueba o reprueba" y este se almacena en la variable 1 que es la primer "calificación 1" para después pasar a la calificación 2 y 3 y al final de esto se suman esas 3 calificaciones y poder dividirlas entre esas 3 y así obtener nuestro promedio para ver si el alumno aprueba o reprueba.

En el diagrama que se muestra en este ejercicio nos indica cómo se van realizando cada paso de acuerdo a los 3 calificaciones y el promedio que se van a obtener como el resultado total y al final obtener el resultado.

PRACTICA 4

Un alumno de 6° grado de primaria desea saber cuanto es que obtuvo como calificación final del primer bloque, en este caso su maestra solo le da los resultados de sus trabajos y exámenes, pero el necesita sacar su promedio final. A continuación se realiza el siguiente algoritmo:

The screenshot shows two windows. The left window is a pseudocode editor titled 'Calcularcalfificaciones 2.psc'. It contains the following code:

```
1 Algoritmo Calcularcalfificaciones
2 Definir cal1, cal2, cal3, examen, trabajo, resultado como Real;
3 Escribir "Dime la calificacion uno:";
4 Leer cal1;
5 Escribir "Dime la calificacion dos:";
6 Leer cal2;
7 Escribir "Dime la calificacion tres:";
8 Leer cal3;
9 Escribir "Dime la calificacion del examen:";
10 Leer examen;
11 Escribir "Dime la calificacion del trabajo:";
12 Leer trabajo;
13 resultado ← ((cal1 + cal2 + cal3)/3)*0.55 + 0.3*examen + 0.15*trabajo;
14 Escribir "Calificacion final:", resultado;
15 FinAlgoritmo
```

The right window is a flowchart titled 'Algoritmo Calcularcalfificaciones'. It starts with 'Definir cal1, cal2, cal3...' and proceeds through a series of input and output steps: 'Dime la calificacion uno:', 'cal1', 'Dime la calificacion dos:', 'cal2', 'Dime la calificacion tres:', 'cal3', 'Dime la calificacion del examen:', 'examen', 'Dime la calificacion del trabajo:', 'trabajo', 'resultado ← ((cal1+cal2+cal3)/3)*0.55 + 0.3*examen + 0.15*trabajo', and finally 'Calificacion final:', resultado. The flowchart ends with 'FinAlgoritmo'.

```
Algoritmo Calcularcalfificaciones
Definir cal1, cal2, cal3, examen, trabajo,
resultado como Real;
Escribir "Dime la calificacion uno:";
Leer cal1;
Escribir "Dime la calificacion dos:";
Leer cal2;
Escribir "Dime la calificacion tres:";
Leer cal3;
Escribir "Dime la calificacion del
examen:";
Leer examen;
Escribir "Dime la calificacion del trabajo:";
Leer trabajo;
resultado ← ((cal1 + cal2 + cal3)/3)*0.55 +
0.3*examen + 0.15*trabajo;
Escribir "Calificacion final:", resultado;
FinAlgoritmo
```

Nuestro algoritmo principal y como dato de entrada es el escribir "Calcular calificacion" y este se almacena en la variable 1 que es la primer "calificación 1, 2 y 3" para después pasar a escribir las 3 primeras calificaciones y sumarlas para al final solo multiplicar por la calificación de examen y trabajos y así mismo obtener su promedio final del alumno.

En el diagrama que se muestra en este ejercicio es capaz de interpretar los pseudocódigos y transformarlos a diagrama de flujo, para eso dispone de un visualizador y editor de diagramas de flujo. Esto es útil si queremos analizar el pseudocódigo desde un punto de vista gráfico. Se accede pulsando el ícono de la barra de tareas.

PRACTICA 5

Escribir un programa que a partir de un número ingresado diga si es mayor, menor o igual a 9.

The screenshot displays three windows from a Windows operating system. The top-left window, titled 'PSeInt', shows a pseudocode editor for 'NUMERO.psc' with the following code:

```
1 Algoritmo NUMERO
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
```

The top-right window, titled 'PSDraw - NUMERO', shows a flowchart that visually represents the pseudocode. It starts with 'Algoritmo NUMERO', followed by 'N ← 0', 'Escribir el numero', and a decision diamond '(N)=9'. If true (V), it outputs 'El numero es igual a 9'. If false (F), it goes to another decision diamond '(N)>9'. If true (V), it outputs 'El numero es mayor a 9'. If false (F), it outputs 'El numero es menor a 9'. Both paths eventually lead to 'FinAlgoritmo'.

The bottom window, titled 'PSeInt - Ejecutando proceso NUMERO', shows the execution output:

```
*** Ejecución Iniciada. ***
Escribir el numero
> 5
El numero es menor a 9
*** Ejecución Finalizada. ***
```

The Windows taskbar at the bottom shows the system tray with the time 06:37 p.m. on 08/05/2023.

Algoritmo NUMERO

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

Nuestro algoritmo principal y como dato de entrada es el escribir "Numero" y este se almacena en la variable 1 que es el primer numero, después damos enter para saber si este numero es igual, menor o mayor a 9 . y es así como obtenemos la indicación al final si nuestro numero es mayor o menor

En el diagrama que se muestra en este ejercicio es capaz de interpretar los pseudocódigos y transformarlos a diagrama de flujo, para eso dispone de un visualizador y editor de diagramas de flujo. Esto es útil si queremos analizar el pseudocódigo desde un punto de vista gráfico. Se accede pulsando el ícono de la barra de tareas.

PRACTICA 6

una ama de casa va a una tienda de abarrotes pero solo lleva \$500 para poder comprar lo basi, necesita ir sumando los productos para saber si lo que lleva le alcanzara. Asi que empieza sumando cada producto que vaya agarrando.

The screenshot displays a Windows desktop environment. On the left, a pseudocode editor window titled 'SUMAA.psc' contains the following code:

```
1 Algoritmo SUMAA
2 Num1←0
3 Num2←0
4 Escribir "Escribir el numero 1"
5 Leer Num1
6 Escribir "Escribir el numero 2"
7 Leer Num2
8 S←Num1+Num2
9 Escribir "El resultado es:" S
10 FinAlgoritmo
```

In the center, a terminal window titled 'PSeInt - Ejecutando proceso SUMAA' shows the execution output:

```
*** Ejecución Iniciada. ***
Escribir el numero 1
> 165
Escribir el numero 2
> 450
El resultado es:615
*** Ejecución Finalizada. ***
```

On the right, a flowchart editor window titled 'PSDraw - SUMAA' visualizes the algorithm as a flowchart. The steps are: 'Algoritmo SUMAA' (start), 'Num1 ← 0', 'Num2 ← 0', 'Escribir el numero 1', 'Leer Num1', 'Escribir el numero 2', 'Leer Num2', 'S ← Num1+Num2', 'Escribir "El resultado es:", S', and 'FinAlgoritmo' (end).

```
Algoritmo SUMA
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
```

Nuestro algoritmo principal y como dato de entrada es el escribir "Suma" y este se almacena en la variable 1 que es el primer numero para poder sumar, después damos enter para saber el siguiente numero, y al final volver a dar enter para obtener la suma total de la compra que realizo la señora.

En el diagrama que se muestra en este ejercicio es capaz de interpretar los pseudocódigos y transformarlos a diagrama de flujo, para eso dispone de un visualizador y editor de diagramas de flujo. Esto es útil si queremos analizar el pseudocódigo desde un punto de vista gráfico. Se accede pulsando el ícono de la barra de tareas.

PRACTICA 7

Juan Perez va a una tienda de aparatos electrónicos y necesita calcular el precio de dichos artículos para ello ocupa el siguiente algoritmo.

The screenshot displays a Windows desktop environment. In the foreground, a window titled 'PSeInt - Ejecutando proceso CALC...' shows the execution output of a program. The output reads:
*** Ejecución Iniciada. ***
Dime el precio:
> 1800
Precio final:1530
*** Ejecución Finalizada. ***
Below the terminal, a message states 'La ejecución ha finalizado sin errores.' To the right, a flowchart titled 'Algoritmo CalcularPrecio' illustrates the process: it starts with an oval 'Algoritmo CalcularPrecio', followed by a rectangle 'Definir precio Como Real', a parallelogram 'Dime el precio:', a parallelogram 'precio', a parallelogram ''Precio final:', precio...', and ends with an oval 'FinAlgoritmo'. In the background, a window titled 'PSeInt' shows the pseudocode:
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

Algoritmo CalcularPrecio

Definir precio como Real; Escribir "Dime el precio:";

Leer precio; Escribir "Precio final:", precio-precio*0.15;

FinAlgoritmo

Nuestro algoritmo principal y como dato de entrada es el escribir "Calcular precio" y este se almacena en la única variable así que ponemos en este caso el precio que quiere calcular Juan Perez al final damos enter y para poder obtener el resultado del precio de ese aparato electrónico.

En el diagrama que se muestra en este ejercicio es capaz de interpretar los pseudocódigos y transformarlos a diagrama de flujo, para eso dispone de un visualizador y editor de diagramas de flujo. Esto es útil si queremos analizar el pseudocódigo desde un punto de vista gráfico. Se accede pulsando el ícono de la barra de tareas.

Conclusión

Pseudocódigo es una herramienta que permite describir los pasos de un algoritmo de manera clara y sencilla. Los componentes y la sintaxis del pseudocódigo son similares a los de un lenguaje de programación. Los componentes del pseudocódigo son las estructuras de control, los operadores y las variables.

Finalmente gracias a PSeInt que es un programa informático que nos ayudó a realizar en la construcción de algoritmos mediante el uso de pseudocódigo. Es una herramienta útil y valiosa para aquellos que deseemos aprender programación.

42%

EVALUACION UNIDAD 2.

MATERIA: ALGORITMOS Y LENGUAJE DE PROGRAMACIÓN INGENIERÍA INDUSTRIAL

Se ha registrado el correo del encuestado (211u0075@alumno.itssat.edu.mx) al enviar este formulario.

NOMBRE Y APELLIDOS *

ROSA ISELA CHAPOL PONCIANO

GRUPO: *

401 A

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

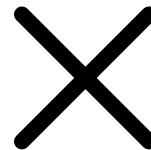
F



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 Un_Rectángulo

Definir área, perímetro como real

Escribir "ingrese la altura del rectangulo"

leer altura

Escribir "ingrese la base del rectangulo"

Leer base

area <- altura*base;

perimetro <- altura+base+altura+base;

Escribir "Valor de area: ", area;

Escribir "Valor de perimetro: ", perimetro;

FinProceso

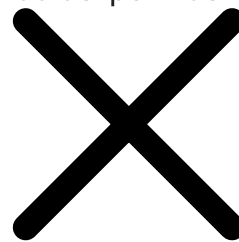


En PSeInt se hace referencia a una variable mediante un identificador (el nombre de la variable). Un identificador debe comenzar con letras, y puede contener solo letras, números y el guión bajo. No puede contener ni espacios ni operadores, ni coincidir con una palabra reservada o función del lenguaje, 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;

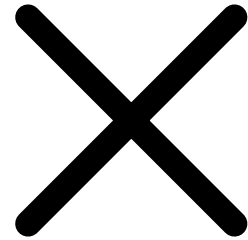
.

.

.

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