

# Guía rápida PSeInt

## 1. Detalle de los Comandos

### ¿Qué es un comando?

Un comando es una orden que se le da a un programa de computadora que actúa como intérprete del mismo, para así realizar una tarea específica.

### Definición de variables

En todo pseudocódigo partimos declarando el tipo de variable que vamos a utilizar.

**Definir:** La instrucción Definir nos permite explicitar el tipo de una o más variables que se utilizarán en el programa. Antes de usar una variable en el proceso, es necesaria indicar que va a existir, esto es definirla y a la vez es necesario indicar de que tipo va a guardar o almacenar, para ello luego de los nombres de las variables, se coloca el indicador **como**, y luego el tipo de variable

Una variable debe definirse antes de ser utilizada por primera vez.

-Los tipos posibles son **REAL**, **ENTERO**, **LÓGICO**, **CARÁCTER**

- **REAL** son sinónimos para el tipo de datos numérico básico, que puede almacenar tanto números reales como enteros. El tipo **ENTERO** es una especialización que sólo permite almacenar valores enteros; cualquier valor no entero que se lea o asigne en una variable de este tipo será truncado.

-Una variable de tipo **LOGICO** sólo puede tomar los valores **VERDADERO** y **FALSO**.

-**CARACTER** Estas pueden contener cero, uno o más caracteres arbitrarios y no tienen una longitud máxima. Si se declara una variable de este tipo y en una lectura el usuario ingresa un número o un valor lógico, se asignará una cadena que contiene el texto ingresado (ejemplo: "1", "VERDADERO", etc).

**Definir** num1, num2 **Como Entero**  
**Definir** nombreAlumno **como Cadena**

### Salida de texto o información

**Escribir:** La instrucción Escribir permite mostrar valores o textos por pantalla.

La instrucción **Escribir** <expr1> , <expr2> , ... , <exprN> nos permite mostrar en pantalla un valor, ya sea el contenido de una variable, el resultado directo de una expresión o bien un texto.

# Guía rápida PSeInt

El comando **podría ser:**

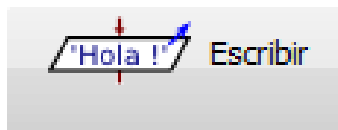
**Escribir <variable>**  
**Escribir <expresión>**  
**Escribir "TEXTO"**

O una combinación de estas separadas por **coma ( , )** donde se evalúa cada una de las expresiones y muestra en pantalla el contenido. En el caso de mostrar un texto siempre debe estar entre comillas dobles.

Ejemplo:

```
Definir edad como ENTERO
Escribir "Ingresa tu edad: "
Leer edad
Escribir "La Variable ", edad, " es un entero"
```

En los comandos, **Escribir** se puede ver un Paralelogramo con una flechita hacia afuera.



## Entrada de datos

**Leer:** La instrucción leer permite ingresar información desde el teclado.

**Leer <variable>**

```
Leer num1
```



## Comentarios

Es una aclaración para comprender mejor el algoritmo, pero no forma parte del mismo, es decir no se procesa.

```
//Definimos la variable edad para ingresar la edad del alumno
Definir edad como ENTERO
```

## 2. Procesos de los algoritmos

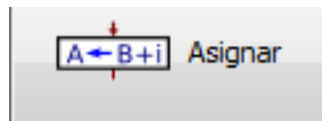
### Asignación

Esta instrucción permite almacenar el valor en una variable, ya sea resultado de una operación o bien el valor de otra variable

El comando es **<variable> = <expresión>** primero evalúa la expresión de la derecha y luego asigna el resultado a la variable de la izquierda. Es importante que el tipo de variable que involucra a la expresión coincida con el tipo de variable de asignación.

`C = A + B` //almacena la suma de las variables A y B en la variable C

`J = 8` //almacena el valor 8 en la variable J



### Operadores

Este pseudolenguaje dispone de un conjunto básico de operadores que pueden ser utilizados para la construcción de expresiones más o menos complejas. La siguiente tabla exhibe la totalidad de los operadores de este lenguaje reducido:

Operador	Significado	Ejemplo
<b>Relacionales</b>		
>	Mayor que	3>2
<	Menor que	'ABC'<'abc'
=	Igual que	4=3
<=	Menor o igual que	'a'<='b'
>=	Mayor o igual que	4>=5
<>	Distinto que	'a'<>'b'
<b>Lógicos</b>		
& ó Y	Conjunción (y).	(7>4) & (2=1) //falso
ó O	Disyunción (o).	(1=1   2=1) //verdadero
~ ó NO	Negación (no).	~(2<5) //falso
<b>Algebraicos</b>		
+	Suma	total <- cant1 + cant2
-	Resta	stock <- disp - venta
*	Multiplicación	area <- base * altura
/	División	porc <- 100 * parte / total
^	Potenciación	sup <- 3.41 * radio ^ 2
% ó MOD	Módulo (resto de la división entera)	resto <- num MOD div

# Guía rápida PSeInt

La jerarquía de los operadores matemáticos es igual a la del álgebra, aunque puede alterarse mediante el uso de paréntesis.

## Borrado de Pantalla

Se puede utilizar una instrucción para limpiar la pantalla en la que se ejecuta el código y posiciona al cursor en la esquina superior izquierda.

La sintaxis es **Borrar Pantalla** y es útil cuando se necesita refrescar la pantalla.

## Espera o Pausa

Sirve para pausar el programa, continuando con el ingreso de cualquier tecla apretada por el usuario, o bien se puede colocar la instrucción de esperar tantos segundos o milisegundos. Las sintaxis se pueden observar en el siguiente ejemplo.

Ejemplo:

**Esperar** Tecla

**Esperar** 4 Segundos

**Esperar** 115 Milisegundos

## 3. Condicionales

### Condición Si-Entonces (if)

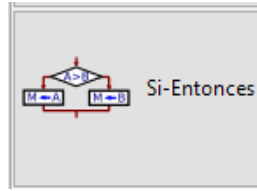
La ejecución de estas sentencias depende del valor de la condición lógica. Al evaluar la expresión lógica, devuelve un Verdadero o un Falso. Si la expresión lógica es Verdadera se realizan las instrucciones que están bajo la indicación. La sintaxis sería:

**Si** <expresión> **Entonces**

Ejemplo:

```
Algoritmo  if
    Definir a,b como entero ;
    Escribir "Ingresa un numero: "
    leer a
    Escribir "Ingresa otro numero: "
    leer b
    Si a>b Entonces
        Escribir "El primer numero ingresado ", a, " es el mayor"
    FinSi
FinAlgoritmo
```

# Guía rápida PSeInt



## Condición Si-Entonces Sino (if - else)

Existe una alternativa que es que el **Si** aparte de tener una acción por el Verdadero, TAMBIEN, tenga una alternativa por el Falso, (PERO SIEMPRE debe estar al menos una por el verdadero).

Si la condición es Falsa, se ejecutan las instrucciones que están bajo la indicación **SiNo**, esta indicación puede no estar, en este caso no se ejecutará nada como en el ejemplo anterior.

Ejemplo:

```
Algoritmo  if
    Definir a,b como entero ;
    Escribir "Ingresa un numero: "
    leer a
    Escribir "Ingresa otro numero: "
    leer b
    Si a>b Entonces
        Escribir "El primer numero ingresado ", a, " es el mayor"
    SiNo
        Escribir "El segundo numero ingresado ", b, " es el mayor"
    FinSi
FinAlgoritmo
```

## Selección Múltiple (switch)

La ejecución de esta sentencia depende del valor de una variable, que va seguido del comando **Segun**, esto evalúa la variable numérica y finaliza con **FinSegun**.

Esta instrucción permite ejecutar opcionalmente varias acciones posibles, al ejecutarse, se evalúa el contenido de la variable y se ejecuta la secuencia de instrucciones asociada con dicho valor que se detallan luego del comando **Hacer**.

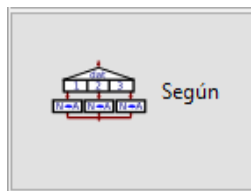
Cada opción está formada por uno o más números separados por comas, dos puntos y una secuencia de instrucciones.

Opcionalmente se puede agregar una opción final denominada **De Otro Modo**, cuya secuencia de instrucciones asociada se ejecutará sólo si el valor almacenado en la variable no coincide con ninguna de las opciones

# Guía rápida PSeInt

Ejemplo:

```
Algoritmo  if
    Definir opc como entero ;
    Escribir "Ingresa una opción: "
    leer opc
    Segun opc Hacer
        1:
            Escribir "Seleccionaste la opcion 1"
        2:
            Escribir "Seleccionaste la opcion 2"
        3:
            Escribir "Seleccionaste la opcion 3"
    De Otro Modo:
        Escribir "No es un valor entre 1 y 3 inclusive"
    Fin Segun
FinAlgoritmo
```



## 4. Ciclos

### Para (for)

La instrucción **Para** ejecuta una secuencia de instrucciones un número determinado de veces que está definido por el valor seguido del comando **Hasta**.

Al ingresar al bloque, la variable de control recibe el valor inicial y se ejecuta la secuencia de instrucciones que forma el cuerpo bucle. Luego se incrementa la variable de control en un paso de unidades determinado y se evalúa la condición de fin. Se repite el ciclo mientras hasta que de verdadera la condición.

Ejemplo:

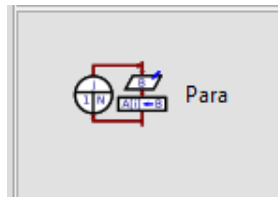
## Guía rápida PSeInt

```
Algoritmo sin_titulo
    Definir num como ENTERO
    Escribir "Los primeros 10 números pares son: "
    Para num = 0 Hasta 10 Con Paso 2 Hacer
        Escribir num
    FinPara
FinAlgoritmo
```

Definimos la variable num como Entero

En **Para**, a num le damos un valor inicial en 0

Repetirá el ciclo mientras num no supere el valor establecido en **Hasta** e irá incrementando el valor de num de 2 en 2



### Mientras (while):

La instrucción **Mientras** ejecuta una secuencia de instrucciones seguidas del comando **Hacer** mientras una condición sea **verdadera**. Al ejecutarse esta instrucción, primero se evalúa la condición. Si la condición resulta **verdadera**, se ejecuta una vez la secuencia de instrucciones que forman el cuerpo del ciclo.

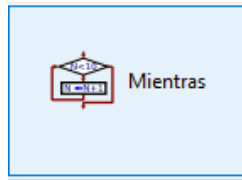
Luego se vuelve a evaluar la condición y, si es verdadera, la ejecución se repite. Caso contrario, que sea **falsa**, **sale del ciclo** y el programa continúa abajo del comando **FinMientras**.

Tener en cuenta **que las instrucciones del cuerpo del ciclo pueden no ejecutarse nunca**, si al evaluar por primera vez la condición resulta ser **falsa**. Por otro lado, si la condición siempre es **verdadera**, al ejecutar esta instrucción se produce un ciclo infinito. A fin de evitarlo, las instrucciones del cuerpo del bucle deben contener alguna instrucción que modifique la o las variables involucradas en la condición, de modo que ésta permita ser **falsa** en algún momento y así finalice la ejecución del ciclo.

Ejemplo:

# Guía rápida PSeInt

```
Algoritmo while
  Definir numIngresar, numSumados como ENTERO
  numSumados = 0
  Mientras numSumados < 250 Hacer
    Leer numIngresar
    numSumados = numSumados + numIngresar
    Escribir "La suma de los numeros ingresados hasta el momento es: ", numSumados
  FinMientras
  Escribir "Se alcanzó o se superó el valor 250"
FinAlgoritmo
```



## Repetir Mientras (do- while)

La sentencia **Repetir** entra a un bucle, donde se ejecuta las instrucciones que le proceden, y luego evalúa una condición determinada por el comando **Mientras Que <condición>** si esta condición es **verdadera** el ciclo sigue ejecutándose, **sale** del mismo una vez que la condición evaluada es **falsa**.

Se debe tener en cuenta que **las instrucciones del cuerpo del ciclo se ejecuta al menos una vez**, ya que **primero ejecuta** las instrucciones **y luego evalúa** la condición de fin. Al igual que el **Mientras**, hay que modificar en el cuerpo del bucle la o las variables que intervienen en la condición, de manera tal que en algún momento obtengamos un falso, y no quedemos en un bucle infinito.

Por lo general este ciclo se utiliza para realizar validaciones.

Ejemplo:

```
Algoritmo do_while
  Definir notaAlumno Como Entero
  Repetir
    Escribir "Ingresa la nota del alumno: "
    Leer notaAlumno
  Mientras Que notaAlumno < 0 o notaAlumno > 10
FinAlgoritmo
```



## Guía rápida PSeInt



# Guía rápida PSeInt

## 5. Herramientas para usar en Pseint

### *Función de los botones*



1. Abre un documento nuevo.
2. Recuperar un algoritmo guardado.
3. Guardar un algoritmo.
4. Guardar como (permite modificar el nombre al algoritmo guardado).



1. Deshacer.
2. Rehacer.
3. Cortar.
4. Copiar.
5. Pegar
6. Corregir el Pseudocódigo.



1. Buscar.
2. Buscar anterior.
3. Buscar siguiente.



1. Ejecutar el algoritmo.
2. Ejecutar paso a paso.
3. Dibujar el diagrama de flujo.



1. Ayuda

## Guía rápida PSeInt