

Pseudocódigo	Descripción
ALGORITMO <nombreAlgoritmo> FINALGORITMO	Indica inicio y fin del algoritmo. Ejemplo: ALGORITMO <i>Factorial</i> FINALGORITMO (el nombre del algoritmo es opcional).
//	La doble barra indica que la línea es un comentario y no será procesada.
;	Fin de línea.
DEFINIR a COMO <tipo>;	Definición de variable de un tipo específico. Los tipos básicos pueden ser: <b>Real:</b> para datos numérico enteros o con punto decimal. <b>Entero:</b> para datos numéricos enteros únicamente. <b>Caracter:</b> para almacenar un caracter. Por ejemplo letra o número. <b>Cadena:</b> para cadenas de caracteres. <b>Logico:</b> para datos Verdadero y Falso únicamente.
DEFINIR a,b,c COMO <tipo>;	Definición de múltiples variables de un mismo tipo. Por ejemplo: DEFINIR dividendo, divisor, resto COMO Entero; DEFINIR nombre, apellido COMO Cadena; DEFINIR p, q COMO Logico;
DEFINIR a COMO Entero; a <- 1;	Definición de variable e inicialización
DEFINIR cad1, cad2, cad3 COMO Cadena; cad1 <- "Ingreso"; cad2 <- "2020"; cad3 <- cad1 + " " + cad2;	Concatenación de variables de tipo cadena usando símbolo "+".
DEFINIR cad1, cad2 COMO Cadena; DEFINIR valor COMO Entero; cad1 <- "Valor: "; ent1 <- 10; cad2 <- cad1, valor;	Concatenación de variables de tipo cadena con variables de tipo numéricas (Entero o Real) usando símbolo ",".
LEER a;	El valor de a es asignado por una interface entrada de datos (ej. teclado)
ESCRIBIR "Valor de a: ", a;	Mostrar una salida. (ej. pantalla)
a <- b;	Asigna a la variable a el valor de b
cont <- cont + 1;	Incrementa a la variable cont en 1 (contador)
acum <- acum+valor;	Incrementa acum en valor (acumulador)
SI (condición) ..... FINSI	Si cumple la condición realizar instrucciones hasta FINSI
SI (condición)	Si cumple la condición realizar instrucciones hasta SINO

..... SINO ..... FINSI	Si no cumple la condición realizar las instrucciones desde SINO hasta FINSI
MIENTRAS (condición) ..... FINMIENTRAS	Mientras cumpla la condición realizar las instrucciones hasta FINMIENTRAS
HACER ..... HASTA QUE (condicion)	Ejecuta las instrucciones hasta que se cumpla la condición.
PARA i<-0 HASTA 9 CON PASO 1 ..... FINPARA	Realizar las instrucciones 10 veces, ya que $0 \leq i \leq 9$ , CON PASO 1 indica que i se incrementara en 1
(a=b)	Retorna un valor Lógico (Verdadero o Falso), según las variables a y b sean iguales o no.
(a != b) o también (a <> b)	Se lee “a distinto de b”, retorna un valor Lógico (Verdadero o Falso), según las variables a y b sean distintas o no
(a > b)	Se lee “a mayor que b”, retorna un valor Lógico (Verdadero o Falso), según las variables a sea mayor que b
(a < b)	Se lee “a menor que b”, retorna un valor Lógico (Verdadero o Falso), según las variables a sea menor que b
(a >= b)	Se lee “a mayor o igual que b”, retorna un valor Lógico (Verdadero o Falso), según las variables a sea mayor o igual que b
(a <= b)	Se lee “a menor igual que b”, retorna un valor Lógico (Verdadero o Falso), según las variables a sea mayor o igual que b
(p Y q)	Retorna un valor Lógico (Verdadero o Falso). Retorna verdadero si p y q son verdadero de lo contrario falso.
(p O q)	Retorna un valor Lógico (Verdadero o Falso). Retorna falso si p y q son falsos de lo contrario verdadero
(NO p)	Retorna un valor Lógico (Verdadero o Falso). Retorna verdadero si p es falso y si falso si p es verdadero.
dividendo MOD divisor	Función congruencia módulo retorna el resto de la división entre <i>dividendo</i> y <i>divisor</i> Ejemplos: (8 MOD 3) retorna 2 (2 es el resto de la división de 8 entre 3); (16 MOD 2) retorna 0 (ya que 0 es el resto de la división de 16 entre 2)
base^exponente	Retorna el resultado de elevar la base al exponente. Ejemplos $2^3$ retorna 8; $9^{0.5}$ retorna 3.
Si $x \geq 0$ entonces RAIZ(x) o [Forma alternativa RC(x)]	Si $x \geq 0$ entonces RAIZ(x) retorna la raíz cuadrada de x. Ejemplo RAIZ(9) retorna 3
AZAR(<hasta>)	Retorna un número entero (n) al azar tal que $1 \leq n \leq hasta$ , Ejemplo: azar(10) retorna números al azar del 1 al 10.
ALEATORIO(<desde>, <hasta>)	Retorna un número entero (n) al azar tal que $desde \leq n \leq hasta$ , Ejemplo: aleatorio(10, 20) retorna números al azar del 10 al 20.
TRUNC(x)	Retorna el número entero del argumento, ejemplo trunc(3.1) retorna 3.
CONVERTIRTEXTO(n)	Retorna el número ingresado como tipo de dato Cadena
CONVERTIRANUMERO(texto)	Retorna el numero ingresado como texto en tipo de dato Entero o Real

### Arreglos unidimensionales o vectores

Pseudocódigo	Descripción
DEFINIR dado COMO Entero; DIMENSION dado[6]; dado[0] <- 1; dado[1] <- 2; dado[2] <- 3; dado[3] <- 4; dado[4] <- 5; dado[5] <- 6;	Arreglo de 6 posiciones. En forma genérica: DIMENSIÓN <variable>[<tamaño>]. Los índices de dado van de 0 a 5.
ESCRIBIR (dado[0]);	Muestra el elemento que ocupa la posición 0. En el caso del dado es 1.
DEFINIR arreglo COMO Entero; DEFINIR n COMO Entero; ESCRIBIR ("¿Longitud?"); LEER n; DIMENSION arreglo[n];	Leer la longitud del arreglo y definirlo.

### Arreglos bidimensionales o matrices

Pseudocódigo	Descripción
DEFINIR matriz COMO Entero; DIMENSION matriz[cantFilas,cantColumnas]; matriz[0,0] <- 1; matriz[0,1] <- 2;	Ej. DIMENSION matriz[3,5]; arreglo bidimensional de 3 filas y 5 columnas. Los índices de las filas son 0, 1, 2 y de las columnas: 0, 1, 2, 3, 4.
ESCRIBIR (matrizA[0,1]);	Muestra el elemento que ocupa la fila 0 y columna 1.
DEFINIR matrizB COMO Entero; DEFINIR cantFilas, cantCol COMO Entero; ESCRIBIR ("¿Cantidad de filas?"); LEER cantFilas; ESCRIBIR ("¿Cantidad de columnas?"); LEER cantCol; DIMENSION matrizB[cantFilas, cantCol];	Leer la longitud de las filas y columnas de la matriz y definirla.