

Entorno: Sintaxis		
Java	Pseudocódigo	PseInt
BASICO		
Apertura de programa		
<pre>Public class <clase> { Public static void main (String[] args) { ... } }</pre>	<pre>PROGRAMA <nombre> ... FIN</pre>	<pre>Proceso <nombre> ... FinProceso</pre>
Declaración de variables		
<pre>final PI = 3.14; final EDAD = 18; int a, b; double float long c; boolean d; <tipo> <tabla>[]; <tipo>[] <tabla>;</pre>	<pre>CTE PI = 3.14 EDAD = 18 VAR a, b : ENTERO c : REAL d: BOOLEANO t: ENTERO []</pre>	<pre>Definir <variable1>, <variable2> Como Entero Definir <variable1>, <variable2> Como Caracter Definir <variable1>, <variable2> Como Real Definir <variable1>, <variable2> Como Logico Definir <tabla> como Entero Carácter Real Logico Dimension <tabla>[<tamaño>]</pre>
Asignación de variables		
<pre><variable> = <valor>; <tabla> = new <tipo>[<tamaño>]; <booleano> = true false;</pre>	<pre><variable> = <valor> <tabla> = new ENTERO [100] <booleano> = true false <booleano> = verdadero falso</pre>	<pre><variable> <- <valor> <tabla>[<índice>] <- <valor> <booleano> <- verdadero falso</pre>
Mostrar mensajes por pantalla		
<pre>System.out.println ("Hola mundo"); System.out.println (<variable>); System.out.println ("El número es: " + <variable>);</pre>	<pre>ESCRIBIR ("Hola mundo") ESCRIBIR (<variable>) ESCRIBIR ("El número es: " + <variable>)</pre>	<pre>Escribir "Hola mundo" Escribir <variable> Escribir "El número es: ", <variable></pre>
Mostrar mensajes por pantalla sin saltar		
<pre>System.out.print (Mensaje sin salto de línea);</pre>	<pre>ESCRIBIR_SS ("Mensaje sin salto de línea")</pre>	<pre>Escribir sin saltar "Mensaje sin salto de línea"</pre>

Leer del teclado		
Scanner <variable> = new Scanner (System.in); <variable> = sc.next<tipo> (); sc.close ();	LEER (<variable>)	Leer <variable>
Incremento de variables		
<contador> = <contador> + 1; <contador>++; <contador>--;	<contador> = <contador> + 1 <contador>++ <contador>--	<contador> <- <contador> + 1 <contador> <- <contador> - 1
Resto de una división		
<resto> = <dividendo> % <divisor>;	<resto> = <dividendo> % <divisor> <resto> = <dividendo> MOD <divisor>	<resto> <- <dividendo> % <divisor> <resto> <- <dividendo> MOD <divisor>
Potencia de un número		
<potencia> = Math.Pow (<base>, <exponente>);	<potencia> = <base> ^ <exponente>	<potencia = <base> ^ <exponente>
CONDICIONALES		
Condicional Simple		
if (<condición>) { ... }	SI <condición> ... FIN SI	Si <condición> Entonces ... FinSi
Condicional Doble		
if (<condición>) { ... } else { ... }	SI <condición> ... SI NO ... FIN SI	Si <condición> ... SiNo ... FinSi

Condicional Anidado		
<pre> if (<condición>) { ... } else if (<condición>) { ... } </pre>	<pre> SI <condición> ... SI NO SI <condición> ... FIN SI FIN SI </pre>	<pre> Si <condición> Entonces ... SiNo Si <condición> Entonces ... FinSi FinSi </pre>
Condicional Múltiple		
<pre> switch (<variable>) { Case <valor>: ... break; Default: ... } </pre>	<pre> SEGUN <variable> CASO <valor> : ... DEFECTO : ... FIN SEGUN </pre>	<pre> Según <variable> Hacer <valor1> : ... De otro Modo : ... FinSegun </pre>
BUCLES		
Bucle Pre-evaluación		
<pre> while (<condición>) { ... } </pre>	<pre> MIENTRAS <condición> ... FIN MIENTRAS </pre>	<pre> Mientras <condición> Hacer ... FinMientras </pre>
Bucle Pos-evaluación		
<pre> Do { ... } while (<condición>); </pre>	<pre> REPETIR ... MIENTRAS <condición> </pre>	<pre> Repetir ... Hasta que <condición> </pre>
Bucle Incremental		
<pre> For (<tipo> <variable> = 0 ; <condición>; <variable>++) { ... } </pre>	<pre> VAR i : ENTERO PARA i = 1 HASTA 10 INC +2 ... FIN PARA </pre>	<pre> Para i <- 0 hasta 10 con paso 2 hacer ... FinPara </pre>

FUNCIONES		
Declaración de funciones y procedimientos		
Public static <tipo> <nombre_func> ([<parametro1>, . .]) { [<tipo> <valorRetorno>;] . . . [return <valorRetorno>;] }	FUNCION <nombre_func> ([<parámetro1>:<tipo>]): [<tipo>] VAR [<valorRetorno> : <tipo>] . . . [DEVOLVER <valorRetorno>] FIN FUNCION	
Truncar un número		
<parteEntera> (int) <numeroReal>;	<parteEntera> = TRUNC (<numeroReal>)	<parteEntera> <- TRUNC (<numeroReal>)
Tablas (primitivas)		
Declaración y definición de tablas		
<tipo>[] <nombreTabla>; <nombreTabla> = new <tipo> { valor1, valor2, . .};	<nombreTabla> : <tipo>[] <nombreTabla> = new <tipo>[<tamaño>]	