

	Convención para Pseudocódigo	Convención para PHP
Tipos de Datos	<p>En el Pseudocódigo el tipo de dato se declara estáticamente, antes de utilizar la variable debemos declarar su tipo.</p> <p>Otros lenguajes que declaran los tipos explícitamente son JAVA, C#, C</p> <p>ENTERO</p> <p>BOOLEAN</p> <p>FLOAT (= DECIMAL. Se utiliza el punto como separador de decimales)</p> <p>STRING (= TEXTO. Para delimitar la cadena de caracteres usaremos comillas dobles: " ")</p>	<p>En PHP, el tipo de dato de una variable se establece dinámicamente cuando se ejecuta el programa, y depende del valor que se le asigne a la variable</p> <p>Integer</p> <p>Boolean (valores TRUE o FALSE)</p> <p>Float (se utiliza el punto como separador de decimales)</p> <p>String (para delimitar la cadena de caracteres usaremos comillas dobles: " ")</p> <p>(http://php.net/manual/es/language.types.php)</p>
Variable	<p>Identificador: utilizaremos notación <u>lowerCamelCase</u>, letras mayúsculas para inicializar las palabras, excepto la primera letra. No utilizaremos guión bajo (_).</p> <p>Los identificadores deben ser nombres significativos del problema que estamos resolviendo.</p> <p>ejemplos:</p> <p><u>ladoMenor</u></p> <p><u>ladoMayor</u></p> <p><u>perimetroRectangulo</u></p> <p>Declaración de variable significa: indicar el nombre y tipo de dato de la variable.</p> <p>Inicializar variable significa: asignar el primer valor a la variable. A partir de la inicialización decimos que la variable está definida.</p>	<p>Identificador (con la misma convención <u>lowerCamelCase</u> que el Pseudocódigo) al que antepondremos el signo \$</p> <p>ejemplos:</p> <p><u>\$ladoMenor</u></p> <p><u>\$ladoMayor</u></p> <p><u>\$perimetroRectangulo</u></p>
comentarios	<p>Comentarios de varias líneas:</p> <p>(*comentario varias lineas *)</p>	<p>Comentarios de varias líneas:</p> <p>/* comentario varias lineas */</p> <p>//comentario 1 linea</p>
Instrucción de Asignación	<p>nombreVariable Expresion</p> <p>Donde Expresion puede ser:</p> <ul style="list-style-type: none"> un valor una variable con valor combinación de operandos y operadores funciones que retornen valores (tema de unidad 4) <p>Ejemplo:</p> <p><u>ladoMenor</u> 20</p>	<p>\$nombreVariable = Expresion ;</p> <p>Donde Expresion puede ser:</p> <ul style="list-style-type: none"> un valor una variable con valor combinación de operandos y operadores funciones que retornen valores (tema de unidad 4) <p>Ejemplo:</p> <p><u>\$ladoMenor</u> = 20 ;</p>

		(observación: en php las instrucciones terminan con ;)																														
Instrucción de Entrada	Ejemplo: LEER (ladoMenor)	Ejemplo: \$ladoMenor = trim(fgets(STDIN)) ; (observación: en php las instrucciones terminan con ;)																														
Instrucción de Salida	Ejemplo: ESCRIBIR (" El perímetro del rectángulo es " , perimetroRectangulo) (observación: Para la operación de concatenación de string se utiliza coma ",")	Ejemplo: echo " El perímetro del rectángulos es " . perimetroRectangulo; (observación: en php las instrucciones terminan con ; Para la operación de concatenación de string se utiliza punto ".")																														
Separación de instrucciones	Una instrucción por renglón, indentando adecuadamente las instrucciones.	Utilizando punto y coma ";" (Por prolijidad siempre conviene 1 instrucción por renglón, indentando adecuadamente las instrucciones)																														
Estructura de control Secuencial (se ejecuta una instrucción debajo de la otra)	ALGORITMO nombreAlgoritmo (*descripción del algoritmo*) Declaración de variables: Tipo nombresVariables instruccion1 ... instruccionN FIN ALGORITMO	<?php /* Declaración de variables: Tipo nombresVariables */ instruccion1; ... instruccionN; ?> Obs: En lenguajes como PHP, donde el tipo de las variables es dinámico, no hay declaración de tipos. Por eso comentaremos la declaración o no la incluiremos. Podemos decir que es el tipo que esperamos tenga una variable por los valores que serán asignados a dichas variables.																														
Operadores para expresiones	<table border="1"> <thead> <tr> <th></th><th>Pseudocódigo</th><th>PHP</th></tr> </thead> <tbody> <tr> <td>Resta, suma, multiplicación, división</td><td>- + * /</td><td>- + * /</td></tr> <tr> <td>Módulo o Resto de división</td><td>MOD</td><td>%</td></tr> <tr> <td>Concatenación de cadenas de caracteres</td><td>, (coma)</td><td>. (punto)</td></tr> <tr> <td>Comparación igual, distinto</td><td>= <></td><td>== <></td></tr> <tr> <td>Comparación mayor , mayor igual</td><td>> >=</td><td>> >=</td></tr> <tr> <td>Comparación menor , menor igual</td><td>< <=</td><td>< <=</td></tr> <tr> <td>Lógicos / Booleanos</td><td>AND OR NOT</td><td>&& !</td></tr> <tr> <td>Raiz cuadrada de E \sqrt{E}</td><td>raiz(E)</td><td>sqrt(E)</td></tr> <tr> <td>Operador Condicional / Ternario</td><td>SI E1 ENTONCESE2 SINO E3</td><td>E1? E2 : E3</td></tr> </tbody> </table> <p>Uso del operador ternario:</p> <p>PROGRAMA MayoriaEdad (*determinar si una persona es mayor de edad*) Boolean esMayor, Entero edad, String nombre,mensaje ESCRIBIR("Ingrese su nombre") LEER(nombre) ESCRIBIR("Ingrese su edad") LEER(edad) esMayor <- (edad >= 18) mensaje <- SI esMayor ENTONCES nombre, " es mayor de edad" SINO nombre, " es menor de edad "</p>			Pseudocódigo	PHP	Resta, suma, multiplicación, división	- + * /	- + * /	Módulo o Resto de división	MOD	%	Concatenación de cadenas de caracteres	, (coma)	. (punto)	Comparación igual, distinto	= <>	== <>	Comparación mayor , mayor igual	> >=	> >=	Comparación menor , menor igual	< <=	< <=	Lógicos / Booleanos	AND OR NOT	&& !	Raiz cuadrada de E \sqrt{E}	raiz(E)	sqrt(E)	Operador Condicional / Ternario	SI E1 ENTONCESE2 SINO E3	E1? E2 : E3
	Pseudocódigo	PHP																														
Resta, suma, multiplicación, división	- + * /	- + * /																														
Módulo o Resto de división	MOD	%																														
Concatenación de cadenas de caracteres	, (coma)	. (punto)																														
Comparación igual, distinto	= <>	== <>																														
Comparación mayor , mayor igual	> >=	> >=																														
Comparación menor , menor igual	< <=	< <=																														
Lógicos / Booleanos	AND OR NOT	&& !																														
Raiz cuadrada de E \sqrt{E}	raiz(E)	sqrt(E)																														
Operador Condicional / Ternario	SI E1 ENTONCESE2 SINO E3	E1? E2 : E3																														

	ESCRIBIR (mensaje) FIN PROGRAMA	
Modularización : Funciones	<pre> (** *Descripción del módulo sin retorno (¿qué hace?) *) MODULO nombre(tipo1 pf1, tipo2 pf2,...,tipoN pfN) RETORNO ∅ instruccion1 ... instruccionN FIN MODULO (** * Descripción del módulo con retorno (¿qué hace?) *) MODULO nombre(tipo1 pf1, tipo2 pf2,...,tipoN pfN) RETORNO tipo instruccion1 ... instruccionN RETORNO(E) FIN MODULO (obs.: tipo puede ser boolean, integer, float, string, etc.) </pre>	<pre> /** *Descripción del modulo sin retorno (¿qué hace?) * @param tipo \$pf1 * @param tipo2 \$pf2 * ... * @param tipo \$pfN */ function nombre(\$pf1, \$pf2,...,\$pfN){ instruccion1; ... instruccionN; } /** *Descripción del modulo con retorno (¿qué hace?) * @param tipo1 \$pf1 * @param tipo2 \$pf2 * ... * @param tipoN \$pfN * @return tipo */ function nombre(\$pf1, \$pf2,...,\$pfN){ instruccion1; ... instruccionN; return E; } </pre>
Alternativa: Instrucción IF	<pre> SI (condición) ENTONCES instruccion1 instruccionN FIN SI SI (condición) ENTONCES instruccionA1 instruccionAN SINO instruccionB1 instruccionBN FIN SI SI (condición) ENTONCES instruccionA1 instruccionAN OTRO-SI (condiciónB) ENTONCES instruccionB1 instruccionBN OTRO-SI (condiciónM) ENTONCES instruccionM1 instruccionMN FIN SI SI (condición) ENTONCES </pre>	<pre> if (condición){ Instrucción1; InstrucciónN; } if (condición){ InstrucciónA1; InstrucciónAN; } else{ InstrucciónB1; InstrucciónBN; } if (condiciónA){ InstrucciónA1; InstrucciónAN; } elseif (condiciónB){ InstrucciónB1; InstrucciónBN; } elseif (condiciónM){ InstrucciónM1; InstrucciónMN; } if (condiciónA){ </pre>

	<pre> instruccionA1 instruccionAN OTRO-SI (condiciónB) ENTONCES instruccionB1 instruccionBN OTRO-SI (condiciónM) ENTONCES instruccionM1 instruccionMN SINO instruccionN1 instruccionNN FIN SI</pre>	<pre>InstrucciónA1; InstrucciónAN; } elseif (condiciónB){ InstrucciónB1; InstrucciónBN; } elseif (condiciónM){ InstrucciónM1; InstrucciónMN; } else{ InstrucciónN1; InstrucciónNN; }</pre>
--	--	--