



UT 2

IDENTIFICACIÓN DE LOS ELEMENTOS DE UN PROGRAMA INFORMÁTICO. PARTE II

Elementos básicos de un lenguaje de programación. Variables. Tipos de datos. Constantes. Comentarios. Literales.



IES JUAN DE LA CIERVA
DPTO. INFORMÁTICA

ELEMENTOS BÁSICOS DE UN PROGRAMA

- Identificadores
- Variables. Declaración, inicialización y utilización. Almacenamiento
- Tipos de Datos
- Literales
- Constantes
- Entrada y salida de Información.
- Operadores y Expresiones
- Conversiones de tipos (casting)
- Comentarios

ELEMENTOS DE UN PSEUDOCODIGO

□ DATOS:

- CONSTANTES.
- VARIABLES.

□ OPERADORES:

- ARITMÉTICOS.
- RELACIONALES.
- LÓGICOS.

□ **EXPRESIONES**, combinación de constantes, variables, operadores y paréntesis. Se evalúan de izquierda a derecha y tienen prioridades diferentes.

ELEMENTOS DE UN PSEUDOCODIGO

❑ **INSTRUCCIONES**, combinación de expresiones y palabras reservadas que indican acciones a llevar a cabo.

➤ **INSTRUCCIONES DE ENTRADA Y SALIDA.**

➤ **INSTRUCCIONES ALTERNATIVAS.**

➤ **SIMPLES.**

➤ **DOBLES.**

➤ **MÚLTIPLES.**

➤ **SEGÚN**

➤ **INSTRUCCIONES REPETITIVAS.**

➤ **MIENTRAS**

➤ **PARA**

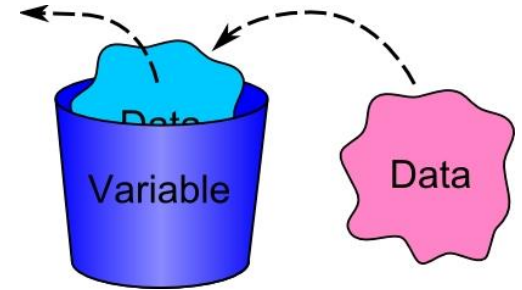
➤ **REPETIR_HASTA**

TIPOS DE DATOS



- Enteros : 3 , -4
- Reales : 3,1416
- Caracteres o cadenas : “Hola”, “Pedro”
- Lógico: Verdadero o Falso

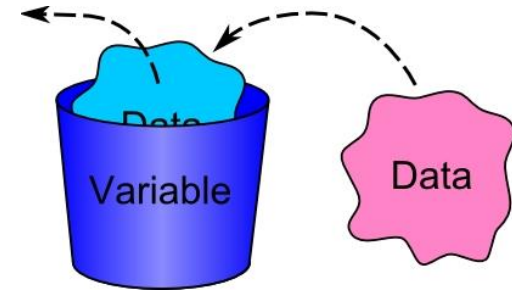
DECLARACIÓN DE VARIABLES



- Variables. Nos permiten almacenar información
- Debemos indicar las variables que vamos a utilizar y sus tipos. Recomendable hacerlo al principio del pseudocódigo
- ¿Cómo la defino o declaro?

Definir <var1>,<var2>,...,<varN> como <Tipo de Dato>;

DECLARACIÓN DE VARIABLES



➤ Tipos de datos en Pseint:

Entero: **Entero**

Real: **Real, Numérico o Número**

Cadena de caracteres: **Caracter, Texto o Cadena**

Lógico: **Lógico**

➤ Ejemplo:

Definir edad **como Entero**;

Definir saludo **como Cadena**;

Definir superficie, diametro **como Real**;

EXPRESIONES

The diagram shows the expression $X + 4X \cdot 2^2 - (3/X)$ with green arrows pointing to its parts: 'variable' points to the first X , 'coeficiente' points to the 4, 'operadores' points to the $+$, \cdot , and $-$ signs, 'exponente' points to the 2 in 2^2 , and 'paréntesis' points to the parentheses around $3/X$.

Combinación de literales, variables, operadores y funciones que proporcionarán un resultado que podré mostrar o asignar a una variable.

Operadores:

- Aritméticos: + - * / ^ % o mod
- Relacionales o de Comparación: Su resultado es un valor lógico
- Lógico : Su resultado es un valor lógico.

Funciones: Operaciones complejas programadas en el lenguaje y que tengo a mi disposición.

ASIGNACIÓN DE VALORES A VARIABLES

Para asignar un valor o el resultado de una expresión a una variable utilizamos el operador de asignación.



- No confundir el operador de asignación con el operador.



Algunas reglas:

- No asignar valores a variables no definidas
- El valor asignado debe ser del mismo tipo que la variable a la que se asigna.
- No se puede utilizar una variable sin inicializar.
- En cada asignación se pierde el valor anterior de la variable.

OPERADORES

ARITMÉTICAS RELACIONALES LÓGICAS

+	>	AND
-	<	OR
*	<=	NOT
/	==	
	!=	

Operadores

Son todos aquellos símbolos que representan enlaces entre cada uno de los argumentos que intervienen en una operación.

Se les llama operadores y sirven para construir expresiones.

TIPOS DE OPERADORES

ARITMÉTICAS

+

-

*

/

RELACIONALES

>

<

>=

<=

==

!=

LÓGICAS

AND

OR

NOT

Operadores

OPERADORES ARITMÉTICOS

Operador	Descripción
-	operador unario de cambio de signo
+	Suma
-	Resta
*	Producto
/	División (tanto entera como real)
%	Resto de la división entera

Se utilizan, junto con variables numéricas e identificadores para formar expresiones aritméticas.

OPERADORES ARITMÉTICOS

Ejemplo:

Expresión	Resultado
$12 * 12$	144
12^2	144
$123 \text{ DIV } 4$	30
$12 \text{ MOD } 5$	2

OPERADORES ALFANUMÉRICOS

Concatenación

Unen expresiones alfanuméricas.

Ejemplo:

Expresión	Resultado
"Pseudo" + "Código"	"Pseudocódigo"
"3" + "." + "1416"	"3.1416"
"Hola" + "Que tal"	"Hola Que tal"

OPERADORES RELACIONALES

Listado de operadores relacionales.	
Operador relacional	Significado
=	Igual a...
>	Mayor que...
<	Menor que...
>=	Mayor o igual que...
<=	Menor o igual que...
<>	Diferente o no igual a...

Se utilizan para formar expresiones lógicas, es decir, expresiones que al ser evaluadas dan un resultado lógico.

Resultado lógico:

- ✓ True, verdadero
- ✓ False, falso

OPERADORES RELACIONALES

Ejemplo:

Comparación	Resultado
$25 \leq 25$	Verdad
$25 < > 25$	Falso
$25 < > 4$	Verdad

OPERADORES LÓGICOS O BOOLEANOS

Signo	Significado
OR	Suma Lógica
AND	Producto Lógico
NOT	Negación

Se comportan de acuerdo al Algebra de Boole.

Operan sobre expresiones lógicas y dan como resultado un valor lógico (true o false)

OPERADORES LÓGICOS O BOOLEANOS

- **OR:** Es un operador binario, afecta a dos operadores. La expresión que forma es cierta cuando al menos uno de sus operadores es cierto.
- **AND:** Es un operador binario, la expresión que forma es cierta cuando ambos operadores son ciertos al mismo tiempo.
- **NOT:** No es un operador binario, afecta un operador cambiando su estado lógico, si era verdad lo convierte a falso.

OPERADORES LÓGICOS O BOOLEANOS

Ejemplo:

Expresión	Resultado
Verdad AND Falso	Falso
NOT Falso	Verdad
Verdad OR Falso	Verdad
Verdad AND Verdad	Verdad

OPERADORES LÓGICOS O BOOLEANOS

AND		
VARIABLES		ESTADO
SA	SB	Y
False	False	False
False	True	False
True	False	False
True	True	True

OR		
VARIABLES		ESTADO
SA	SB	Y
False	False	False
False	True	True
True	False	True
True	True	True

XOR		
VARIABLE		ESTADO
SA	SB	Y
False	False	False
False	True	True
True	False	True
True	True	False

NOT	
VARIABLE	ESTADO
SA	Y
True	False
False	True

OPERADORES LÓGICOS O BOOLEANOS

Ejemplos

Operador	Descripción	Ejemplo de expresión	Resultado del ejemplo
!	Negación - NOT (unario)	<code>!false</code> <code>!(5==5)</code>	<code>true</code> <code>false</code>
	Suma lógica – OR (binario)	<code>true false</code> <code>(5==5) (5<4)</code>	<code>true</code> <code>true</code>
^	Suma lógica exclusiva – XOR (binario)	<code>true ^ false</code> <code>(5==5) (5<4)</code>	<code>true</code> <code>true</code>
&	Producto lógico – AND (binario)	<code>true & false</code> <code>(5==5) & (5<4)</code>	<code>false</code> <code>false</code>
	Suma lógica con cortocircuito: si el primer operando es <code>true</code> entonces el segundo se salta y el resultado es <code>true</code>	<code>true false</code> <code>(5==5) (5<4)</code>	<code>true</code> <code>true</code>
&&	Producto lógico con cortocircuito: si el primer operando es <code>false</code> entonces el segundo se salta y el resultado es <code>false</code>	<code>false && true</code> <code>(5==5) && (5<4)</code>	<code>false</code> <code>false</code>

OPERADORES - PRECEDENCIA

++(suf)	--(suf)				
++(pref)	--(pref)	!	casting		
*	/	%			
+	-				
>	>=	<	<=		
==	!=				
&					
&&					
=	+=	-=	*=	/=	%=

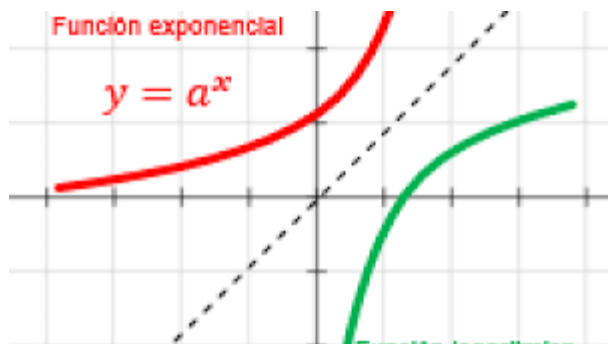
ENTRADA/SALIDA DE DATOS



OTRAS INSTRUCCIONES

- **Borrar Pantalla;**
- **Esperar Tecla;**
- **Esperar <número>
segundos;**

FUNCIONES



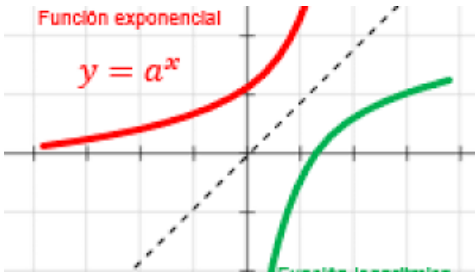
- **F. Matemáticas:**
 - valor absoluto
 - truncar
 - número aleatorio
 - seno de un ángulo

- **F. de Cadenas**

FUNCIONES MATEMÁTICAS (en Pseint)

- **rc(número) o raiz(número):** devuelve la raíz cuadrada del número.
- **abs(número):** Devuelve el valor absoluto del número
- **ln(número):** Devuelve el logaritmo natural del número
- **exp(número):** Devuelve la función exponencial del número.
- **sen(número):** Devuelve el seno de número.
- **cos(número):** Devuelve el coseno de número.
- **tan(número):** Devuelve la tangente de número.
- **asen(número):** Devuelve el arcoseno de número.
- **acos(número):** Devuelve el arcocoseno de número.
- **atan(número):** Devuelve el arcotangente de número.
- **trunc(número):** Devuelve la parte entera de número.
- **redon(número):** Devuelve el entero más cercano a número.
- **azar(número):** Devuelve el entero aleatorio en el rango [0;número-1].
- **aleatorio(numero1,numero2):** Devuelve el entero aleatorio en el rango [numero1;numero2].

FUNCIONES MATEMÁTICAS



Escribir "Raíz cuadrada de 9: ",rc(9);

Escribir "Valor absoluto de -3: ",abs(-3);

Escribir "Seno de 90 grados: ",sen(90 * PI / 180);

Escribir "Truncamos 3.7: ",trunc(3.7);

Escribir "Redondeamos 2.7: ",redon(2.7);

Escribir "Un número al azar del 0 al 9: ",azar(10);

Escribir "Un número al azar entre 10 y 20: ",aleatorio(10,20);

FUNCIONES DE CADENA

(en Pseint)

- **longitud(cadena)**: Devuelve la cantidad de caracteres de la **cadena**.
- **mayusculas(cadena)**: Devuelve una copia de la **cadena** con todos sus caracteres en mayúsculas.
- **minusculas(cadena)**: Devuelve una copia de la **cadena** con todos sus caracteres en minúsculas.
- **subcadena(cadena,pos_ini,pos_fin)**: Devuelve una nueva cadena que consiste en la parte de la **cadena** que va desde la posición **pos_ini** hasta la posición **pos_fin**.
- **concatenar(cadena1,cadena2)**: Devuelve una nueva cadena resulta de unir las cadenas **cadena1** y **cadena2**.
- **convertirANumero(cadena)**: Recibe una cadena de caracteres que contiene un número (caracteres numéricos) y devuelve una variable numérica con el mismo.
- **convertirATexto(numero)**: Recibe un **numero** y devuelve una variable cadena de caracteres de dicho real.

FUNCIONES DE CADENA (en Pseint)

Definir cad1,cad2 como cadena;

Definir num como Entero;

cad1<-"informática";

Escribir "La longitud de cad1 es ",longitud(cad1);

Escribir "El primer carácter de cad1 es ", subcadena(cad1,0,0); Escribir
"El último carácter de cad1 es ", subcadena(cad1,longitud(cad1)-
1,longitud(cad1)-1);

Escribir "La cad1 en mayúsculas es ",mayusculas(cad1);

cad2<-concatenar(cad1," es muy interesante");

Escribir cad2; num<-ConvertirANumero("10");

Escribir num; Escribir Concatenar("El número es
",ConvertirATexto(num));

Escribir "El número es ",num;