

# Taller de Conocimientos Específicos

## Departamento de Informática

Lic. en Ciencias de la Computación  
Lic. en Sistemas de Información

Facultad de Ciencias Exactas, Físicas y Naturales  
UNSJ

### CONCEPTOS BÁSICOS

---

#### INTRODUCCIÓN

El hombre se enfrenta constante y habitualmente a innumerables problemas. Muchas soluciones las alcanza recurriendo a lo aprendido a través de la experiencia, para otras necesita el apoyo de herramientas aportadas por las diversas ramas de la tecnología. Una de las herramientas que ha provocado mayor impacto es la computadora.

La computadora no puede resolver nada sin que el hombre determine la forma y recursos necesarios para hacerlo. Es claro, que dar respuesta a un problema exige conocimiento, reflexión, razonamiento lógico y alguna dosis de ingenio y sagacidad. La experiencia indica que aprender un lenguaje de programación no resulta tarea complicada, lo complejo resulta encontrar el camino adecuado para la resolución de la problemática planteada.

En este curso se brindarán algunas estrategias para resolver problemas, para eso se analizan además los tipos de datos y estructuras de control que permitirán expresar las órdenes adecuadas para realizar luego una codificación.

La tarea de aprender a programar en un determinado lenguaje, puede resultar más sencilla si previamente se utiliza un lenguaje cuya sintaxis resulte más familiar y cercana al lenguaje coloquial. En este curso, a través del uso de pseudocódigo, se introducirán los elementos básicos y estructuras, que constituyen un lenguaje.

#### FORMALIZACIÓN DE CONCEPTOS PARA LA CONSTRUCCIÓN DE ALGORITMOS

Ahora se tratará de formalizar algunos conceptos utilizando un pseudolenguaje llamado **pseudocódigo**.

Concepto de Algoritmo

**Un algoritmo es un método para resolver problemas, que consiste de una secuencia ordenada y finita de acciones o pasos, cada una de las cuales con un significado preciso, que debe ejecutarse en un tiempo finito.**

Es importante hacer notar, que el concepto de algoritmo, aunque similar y obviamente relacionado, no debe confundirse con el concepto de programa. Algoritmo, es la especificación de un conjunto de pasos orientados a la resolución de un problema, mientras que un programa<sup>1</sup> es ese conjunto de operaciones escritas en un determinado lenguaje de programación. El diseño de un algoritmo es independiente del lenguaje que se utilice en la codificación.

---

<sup>1</sup> **Programa** es el conjunto de instrucciones ordenadas que pueden ser interpretadas por la computadora y que le permitirá resolver un problema planteado.

Un algoritmo constituye la documentación principal que se necesita para iniciar la fase de codificación y para representarlo se utilizan dos tipos de notación: pseudocódigo y diagramas de flujo (en este taller solo se verá pseudocódigo).

## RESOLUCIÓN DE PROBLEMAS

Como resolvemos un problema a través de la computadora?. El siguiente esquema plantea la secuencia de pasos necesarios para realizarlos.

En la resolución de problemas se puede distinguir las siguientes etapas:

### **Formulación del problema**

La primera condición al momento de resolver un problema, es contar con un enunciado preciso y claro, que no contenga ningún tipo de ambigüedades.

Una vez que está claro qué hacer y quién lo va a hacer, (en este caso la computadora cuando se lo pase el algoritmo a un lenguaje de programación), se deben conocer los recursos con que se cuenta para especificar cómo hacerlo.

Al conjunto de todos los recursos necesarios para se pueda realizar una tarea se denomina **Ambiente**.

### **Determinación del Algoritmo**

Una vez abstraídos los elementos esenciales del problema a resolver, la forma elegida para su solución se debe expresar como una secuencia ordenada de acciones, que el procesador se deberán ejecutar para obtener los resultados pretendidos.

Esta secuencia de acciones expresadas en un lenguaje simbólico se conoce con el nombre de **Algoritmo**.

### **Codificación**

Este proceso consiste en traducir el algoritmo a un lenguaje de programación que pueda ser interpretado por la computadora; se obtiene de esta manera un **Programa**.

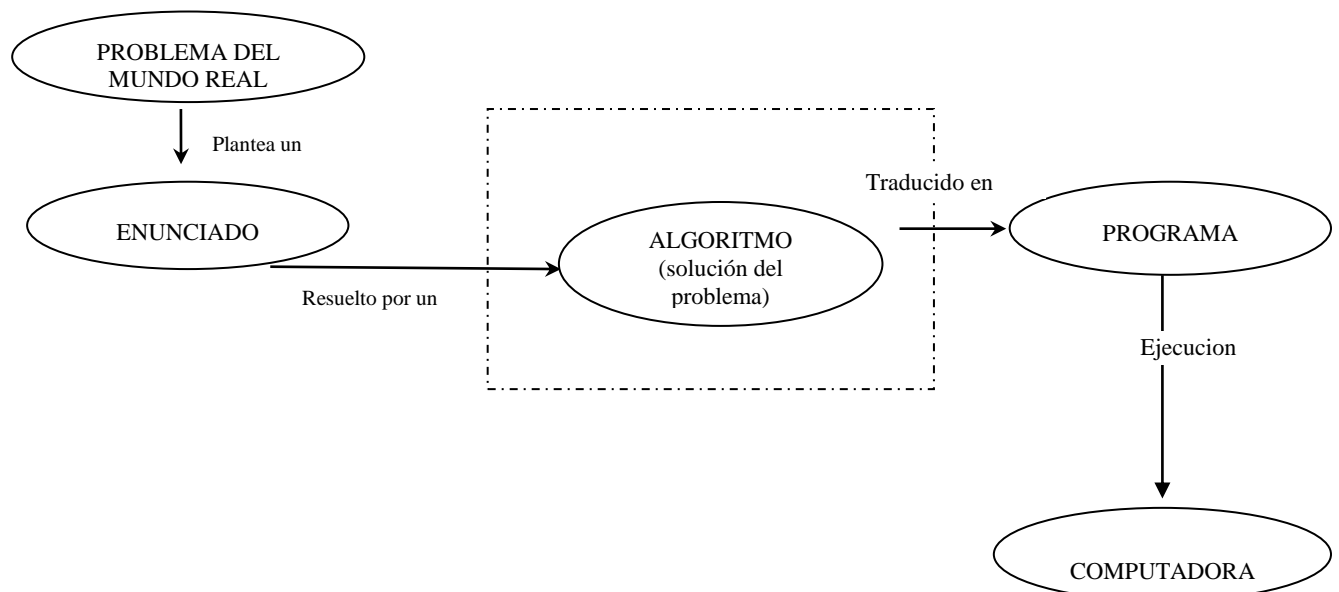
*Operaciones que puede realizar una computadora:*

- ✓ Almacenar los datos en la memoria.
- ✓ Realizar operaciones aritméticas, relacionales y lógicas.
- ✓ Reconocer y ejecutar acciones simples, como las de lectura y escritura, y acciones estructuradas, tales como las estructuras secuenciales, iterativas y selectivas.

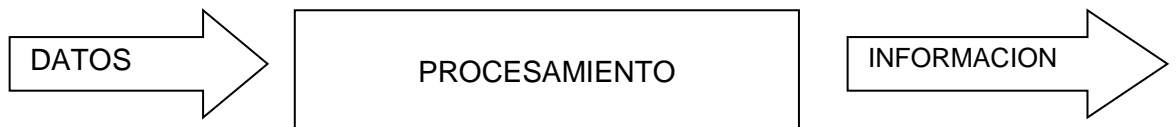
### **Ejecución**

Utilizando un editor del lenguaje seleccionado, se podrá depurar el programa de los errores de sintaxis y ejecutarlo para obtener los resultados esperados.

La siguiente figura esquematiza lo expuesto



## PARTES DE UN ALGORITMO



**DATOS:** Son las características propias de cualquier entidad u objeto. Por ejemplo: los datos de una persona como su edad, fecha de nacimiento, domicilio, número de teléfono, etc.

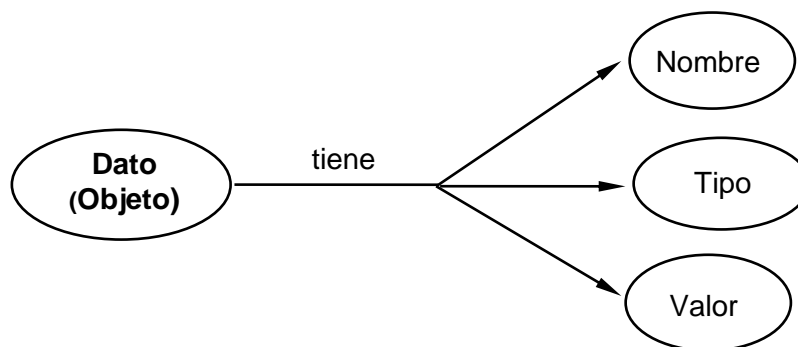
**PROCESAMIENTO [DE DATOS]:** Consiste en la recolección de datos de entrada que son analizados, evaluados y ordenados para ser colocados de manera que produzcan información útil.

**INFORMACIÓN:** Es el conocimiento relevante producido como resultado del procesamiento de datos. Esta "información" es usada por la gente para cumplir ciertos propósitos.

### DATOS

La forma de representar a los objetos del mundo real, para ser procesados por una computadora se denomina **dato**. Elegir la representación adecuada de los datos resulta indispensable para optimizar la solución de un problema.

Cada dato tiene asociado: un Nombre, un Valor y un Tipo, y ocupa un espacio de memoria de la computadora.



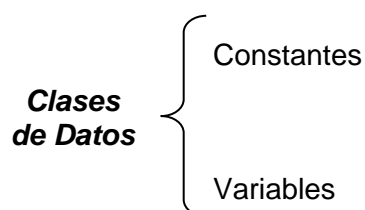
### Ejemplo 1:

real monto  
monto=1000.50

En el ejemplo se identifica una variable de **nombre** monto, de **tipo** real, a la cual se le asigna el **valor** 1000.50

### Valor de dato

Generalmente los datos cambian su valor mientras se procesa el algoritmo, en este caso se habla de datos variables o simplemente **VARIABLES**. Otros, en cambio, no modifican su valor original, por lo que se los llama **CONSTANTES**.



### **Nombre o Identificador de dato**

El nombre de un dato está representado por una secuencia de uno o más caracteres: letras, dígitos y/o algunos caracteres especiales. El primer carácter debe ser una letra y el nombre no debe contener espacios en blanco. Generalmente se seleccionan nombres representativos de los objetos para facilitar su interpretación y/o su posterior modificación.

### **Ejemplo 2:**

Los siguientes son nombres de datos válidos:

Edad, Busca\_mayor, ordena1, fecha, Fecha y FECHA

Estos son nombres de datos no válidos: 11ordena \*fecha /nombre

### **Tipos de Datos**

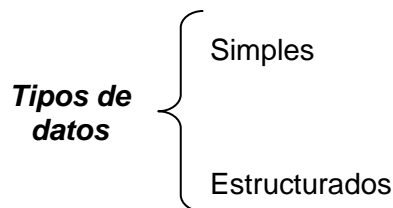
Los datos que utilizan los programas se pueden clasificar en base a diferentes criterios. Uno de los más significativos es aquel que dice que todos los datos que se utilizan son simples o compuestos.

Un tipo de dato define el rango de valores que puede tomar un dato y las operaciones que se pueden realizar con él.

Para determinar el proceso que permitirá obtener el resultado o salida requerida de un problema cuya solución debe ser resuelta por una computadora, es necesario conocer el tipo de datos que ésta puede manipular.

Si un dato contiene un valor que siempre se trabaja como una unidad, se dice que es un dato simple. Cuando el dato está formado por una colección de otros datos, se dice que es una estructura de datos o un tipo de dato estructurado.

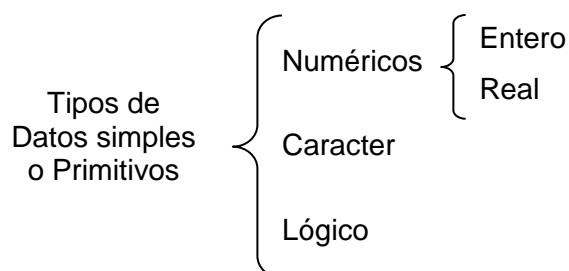
Esto permite visualizar una primera clasificación de los datos en:



### **Tipos de Datos Simples o Primitivos**

Un dato simple es indivisible (atómico), es decir, no se puede descomponer.

Los tipos de datos simples que definen la mayoría de los lenguajes de programación son:



### **Tipo de Dato Numérico**

✓ *Tipo entero:* este tipo permite representar los valores enteros, por lo tanto son útiles para simbolizar edades, códigos numéricos, números de calles, resultados provenientes de procesos de conteo etc.

✓ *Tipo real:* este tipo permite representar valores con punto decimal. Ejemplos de variables de tipo real son las que representan notas, temperaturas, presión arterial, altura, peso, costos, etc.

### Ejemplo 3:

Un año es un dato simple.

Año : 2008

Un año se expresa con un número entero, el cual no se puede descomponer.

Sin embargo, una fecha es un dato compuesto porque está formado por otros tres datos simples (día, mes, año).

Fecha:

Día : 23

Mes : 11

Año : 2015

Las siguientes son las declaraciones de algunas variables de los problemas presentados:

**real** largo, ancho

**entero** edad

**real** x,y,z

Toda variable que se utilice en un algoritmo debe ser declarada, esto es, se debe especificar su nombre y el tipo de dato asociado. La declaración de variables permite a la computadora reservar el espacio de memoria necesario para el almacenamiento de la misma, como así también verificar si las operaciones que se realizan con ella corresponden al tipo declarado. Por convención, para declarar una variable colocaremos el identificador de la misma precedido por su tipo.

### ***Tipo de Dato Character***

Un variable de tipo caracter es aquella que puede tomar uno de los caracteres del código ASCII. La sigla ASCII se refiere a Código Estándar Americano para el Intercambio de Información, y es un código normalizado que cuenta con 128 caracteres, que incluye números, letras mayúsculas, minúsculas y caracteres especiales. Actualmente, la mayoría de los procesadores utilizan el código ASCII ampliado que cuenta de 256 caracteres, entre los que se han agregado los caracteres del alfabeto griego.

En general, todas las computadoras manipulan los siguientes caracteres:

Letras mayúsculas: A..Z. (excepto CH, Ñ, LL)

Letras minúsculas: a..z. (excepto ch, ñ, ll)

Dígitos decimales: 0..9.

Carácter de espacio en blanco.

Caracteres especiales: +, -, %, ... @, μ, λ, α

Signos de puntuación: , ; :

Estos caracteres están ordenados de 0 a 255, característica que permite la comparación entre ellos. Los dígitos están ordenados en su propia secuencia numérica y las letras están dispuestas acorde al orden alfabético, precediendo las mayúsculas a las minúsculas.

Muchos problemas requieren trabajar con variables cuyos valores son caracteres. Supongamos el caso de una variable que se utiliza para identificar el resultado del tratamiento iniciado por un paciente, que puede tomar los valores E o F por ejemplo, que representan el éxito o fracaso del mismo en cada uno de los pacientes. Otro ejemplo es el referido al rendimiento académico de los alumnos donde se puede utilizar una variable que puede tomar los valores E, B, R y M, para representar el nivel de rendimiento.

Como se dijo, las variables de tipo caracter pueden tomar uno de los valores de este conjunto, el que se debe escribir entre apóstrofes ' ', para evitar confundirlo con el nombre de una variable, operador o número.

#### **Ejemplo 4:**

Una letra es un ejemplo de dato simple tipo caracter.

Letra : t

Entonces, para el caso del rendimiento de un alumno, la variable se deberá declarar de la siguiente forma:

caracter rendim

*Nombre del dato* → rendim

*Tipo de dato* → Caracter

*Valor* → 'M'

#### **Tipo de Dato Lógico**

Existen variables que pueden tomar los valores *Falso* o *Verdadero*, corresponden al tipo de dato lógico o Booleano.

La variable **result**, usada para identificar el éxito o fracaso del tratamiento iniciado por un paciente, puede ser también considerada de tipo booleano. Así al tomar esta variable el valor Falso, indica que el tratamiento no dio resultado en ese paciente.

booleano result

*Nombre del dato* → result

*Tipo de dato* → booleano

*Valor* → Falso

Se llaman **tipos ordinales**, aquellos cuyos elementos están ordenados discretamente, esto es, para cada elemento, salvo el primero y el último, existe un anterior y otro posterior. De los tipos de datos estudiados, el único que no es ordinal es el conjunto de los números reales. Dado un número real no se puede determinar el anterior ni el siguiente, ya que entre dos números reales siempre existe otro número real.

### **Tipos de Datos Estructurados**

#### **Tipo de Dato Cadena**

En muchos casos es necesario trabajar con variables cuyos valores son un conjunto de caracteres, por ejemplo las que se utilizan para representar nombres. En estos casos no es posible el uso del tipo carácter, por ello la mayoría de los lenguajes de programación cuentan con el tipo de dato cadena, que permite resolver la situación planteada.

Un dato tipo cadena puede tomar como valor una cadena de caracteres, esto es, una secuencia finita de caracteres encerrada entre comillas " ".

Si la variable **nomb** es utilizada para representar los nombres de las carreras que se cursan en la Facultad de Ciencias Exactas, se puede declarar:

**cadena** nomb

*Nombre del dato* → nomb

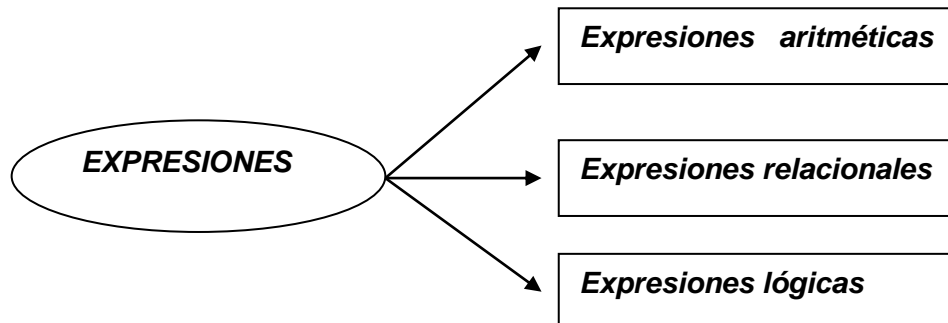
*Tipo de dato* → Cadena

*Valor* → "Tecnicatura en Programación WEB"

## EXPRESIONES

Las expresiones son descripciones de cálculos que al evaluarse devuelven un único resultado.

El tipo de resultado depende de los operadores y operandos que intervienen, por lo que se las clasifica en:



### Estructura de una Expresion

Toda expresion se forma colocando un operador entre dos operandos, del siguiente modo:

*Operando* **Operador** *Operando* [**Operador** *Operando*...]

### Expresiones aritméticas:

Las expresiones aritméticas son descripciones de cálculos matemáticos que al evaluarse devuelven un único resultado de tipo numérico.

OPERANDOS	OPERADORES	TIPO DE RESULTADO
Variables o Constantes numéricas	<b>+</b> suma <b>-</b> resta <b>*</b> multiplicación <b>/</b> cociente <b>div</b> división entera <b>raiz</b> <b>resto</b> <b>potencia</b>	Numérico

Los operadores **raiz** y **resto** se aplican solo a operandos enteros, siendo su resultado de tipo entero.

Modo de uso:

**raiz**(operando)

operando1 **resto** operando2 (el operador **resto** entrega el resto de dividir operando1 en operando2)

Si bien la función **potencia** no está definida en todos los lenguajes de programación, adaptaremos en pseudocódigo un formato similar al utilizado en Lenguaje C:

**Potencia**(base,exponente)

### Reglas de evaluación de una expresión aritmética

Son las mismas reglas que se utilizan en matemática. Se comienza a evaluar la expresión de izquierda a derecha. Si aparece un cálculo entre paréntesis se debe resolver primero. Si aparecen paréntesis anidados, es decir un par de paréntesis dentro de otro, primero se resuelve la expresión del paréntesis interior.

Las operaciones de una expresión aritmética se ejecutan de acuerdo con el siguiente orden de precedencia:

ORDEN DE PRECEDENCIA	OPERADOR
1	* / <i>div</i>
2	+ -
3	<i>raíz resto potencia</i>

En caso de operadores con la misma prioridad, las operaciones se realizan de izquierda a derecha, es decir en el orden en que se encuentran.

Si los operandos de la expresión aritmética son del mismo tipo, el resultado es del mismo tipo.

Si son de distinto tipo, uno entero y otro real por ejemplo, el resultado es real.

#### Ejemplo 5:

Expresiones aritméticas válidas

entero a,b,c,d

real e

a+b+5+d expresión entera

a-b\*c/4 expresión entera

a/b+c-d expresión entera

(a+b)\*(c-d)/e expresión real

Para a=2, b=8 y d=3, el **resultado** de la **evaluación** de la primera expresión será 18.

#### Expresiones relacionales

Las expresiones relacionales son descripciones de cálculos de comparación que al evaluarse devuelven un único resultado de tipo lógico.

En una expresión relacional aparecen dos operandos que deben ser del mismo tipo.

Los operadores utilizados se muestran a continuación:

OPERANDOS	OPERADORES	TIPO DE RESULTADO
Datos del mismo tipo: Ambos numéricos, carácter, cadena o de tipo lógico.	> Mayor < Menor == Igual != Distinto >= Mayor o Igual <= Menor o Igual	Valor lógico (Verdadero o Falso)

#### Ejemplo 6:

Expresiones relacionales válidas

**entero a,b,c,d**

a+b > c+d

a <= d

**carácter m,n**

m != n

m == n

#### Expresiones lógicas:

Las expresiones lógicas son descripciones de cálculos de operaciones lógicas que al evaluarse devuelven un único resultado de tipo lógico.



Los operandos de las expresiones lógicas son expresiones relacionales, variables o constantes de tipo lógico; los operadores utilizados se muestran a continuación:

<b>OPERANDOS</b>	<b>OPERADORES</b>	<b>TIPO DE RESULTADO</b>
Expresiones relacionales, Variables o constantes de tipo lógico.	<b>NO</b> negación <b>Y</b> conjunción <b>O</b> disyunción	Valor lógico (Verdadero o Falso)

El orden de precedencia de los operadores lógicos, es decir, la prioridad que tienen en caso de no estar entre paréntesis es:

<b>ORDEN DE PRECEDENCIA</b>	<b>OPERADOR</b>
1	NO
2	Y
3	O

El valor booleano que resulta de la evaluación de una expresión lógica, se deduce de la **tabla de verdad** que se obtiene combinando los valores posibles que pueden tomar las variables lógicas A y B:

A	B	A Y B	A O B	NO A
V	V	V	V	F
V	F	F	V	F
F	V	F	V	V
F	F	F	F	V

#### **Ejemplo 7:**

Los siguientes son ejemplos de expresiones lógicas:

**caracter a,b,c,d**

(a != b) o (c == d)

( a <= b) y (c != d)

(a >= b) o (b= d)

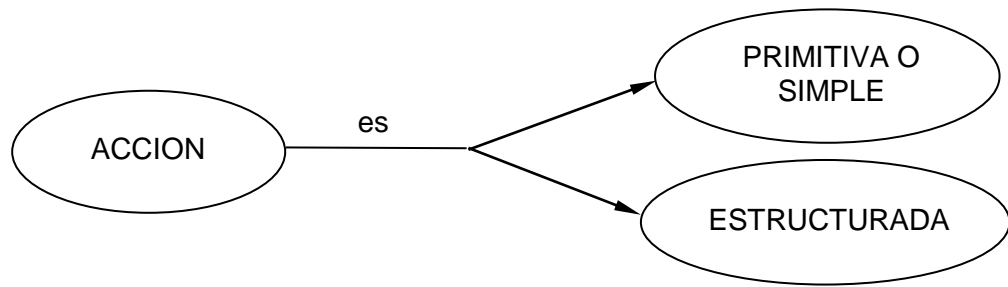
Para a='r', b='z', c='m', d='m', el resultado de la evaluación de la primera expresión será el valor booleano Verdadero.

#### **ACCIONES**

Se llama acción a cada uno de los pasos que el procesador ejecuta para resolver el problema. Las acciones pueden ser simples o estructuradas.

Una acción simple es un paso que es entendido en forma inmediata por el procesador y no puede ser descompuesto en otros pasos.

Una acción estructurada es un paso compuesto por otras acciones simples o estructuradas.



### ***Acciones Primitivas o Simples***

#### ***Lectura:***

La acción primitiva de lectura permite almacenar a una variable, un valor especificado desde un dispositivo externo como el teclado. Por tanto, esta acción permite introducir al ambiente valores desde fuera de él.

El formato de esta acción es:

**Leer** <nombre de variable,.. >

donde:

<nombre de variable> representa uno o más nombres de variables del ambiente.

#### **Ejemplo 8:**

entero a  
real b  
caracter c  
Leer a  
Leer b  
Leer c

#### ***Asignación:***

La acción primitiva de asignación permite almacenar en una variable un valor que proviene de una constante, de otra variable o del resultado de una expresión.

El formato de esta acción es:

<nombre de variable> = <valor>

donde:

**<nombre de variable>** es el nombre de la variable a la que el procesador le asignará un valor

**=** representa el símbolo de asignación

**<valor>** puede ser una constante, una variable o el resultado de una expresión.

Una asignación altera el valor almacenado en la variable que aparece a la izquierda de la asignación, sin producir ningún efecto en la expresión que se encuentra a la derecha de la misma.

Existen por lo tanto, dos formas de ligar un valor a una variable, a través de una acción de lectura (Leer) o por medio de una asignación desde el mismo programa (=).

#### ***Escritura:***

La acción primitiva de escritura, permite comunicar al exterior valores almacenados en el ambiente, a través de un dispositivo, tales como pantalla o impresora.

El formato de esta acción es:

**Escribir** <constante, nombre de variable, expresión...>

donde:

**<constante>** es el valor de un dato constante.

**<nombre de variable>** representa una o más nombres de variables que contienen los valores que se quieren mostrar.

**<expresión>** puede ser una expresión aritmética, relacional o lógica.

## ALGORITMOS Y SEUDOCÓDIGO

En adelante se utilizará pseudocódigo para construir algoritmos que resuelvan un determinado problema. Por lo tanto, para evitar algunas ambigüedades en lo que se desee expresar, es importante respetar las **sintaxis** de cada una de las acciones y las reglas para la construcción de algoritmos.

Un algoritmo escrito en pseudocódigo siempre se suele organizar en tres secciones: *cabecera*, *declaraciones* y *cuerpo*.

En la sección de **cabecera** se escribe el nombre del algoritmo.

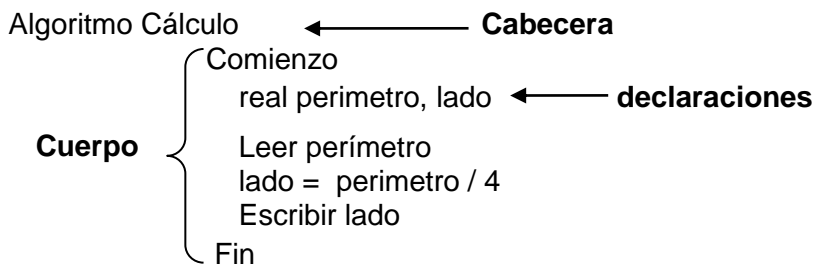
En la sección de **declaraciones** se declaran los datos (variables, constantes,...) que va a utilizar el algoritmo, es decir se especifica su ambiente.

En el **cuerpo** están descritas las acciones necesarias para resolver el problema planteado.

El inicio y el final de un algoritmo deben marcarse con las palabras "Comienzo" y "Fin"

### Ejemplo 9:

Dado el perímetro de un cuadrado calcular la longitud del lado.



Otra forma de resolver este problema es:

Algoritmo Cálculo

Comienzo

constante cant\_lados=4 /\*constantes del ambiente\*/

real perimetro,lado /\*variables del ambiente\*/

Leer perimetro

lado = perimetro / cant\_lados

Escribir lado

Fin

En este ejemplo se observa que se puede asociar un nombre a una constante, esta situación debe ser declarada en el ambiente.

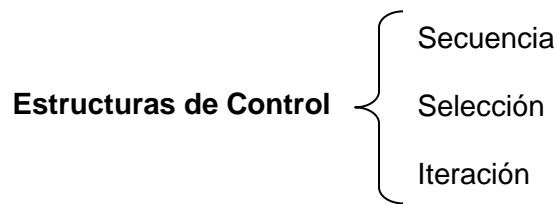
## Acciones estructuradas

El orden en que se ejecutan las acciones de un algoritmo se llama *flujo de control*.

Recordemos que cuando el algoritmo cuenta con un número reducido de acciones que se ejecutan una tras otra, en el mismo orden en el que aparecen, se habla de ejecución secuencial.

También se vio que el flujo de control de un algoritmo puede ser alternativo, cuando se deben tomar decisiones que dependen de ciertas condiciones o iterativo cuando es necesario repetir acciones varias veces.

Se llaman **acciones estructuradas** a aquellas que determinan un flujo de control específico; de ahí que también se denominan estructuras de control. Las sentencias estructuradas se ejecutan teniendo en cuenta los distintos tipos de flujo de control: secuencia, selección e iteración.



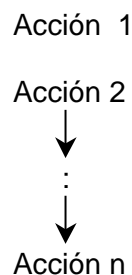
En este curso veremos solamente la estructura de control de Secuencia.

### ESTRUCTURAS DE CONTROL

#### Secuencia

Una secuencia está representada por un conjunto de acciones que se ejecutan en forma consecutiva, una sola vez. En este caso el orden de ejecución coincide con el orden físico en que se representan las órdenes.

El siguiente esquema representa esta estructura:



**Esquema de estructura Secuencial**

## PRÁCTICA

### Ejercicio 1

Definir como datos (Identificador, Tipo y Valor) la siguiente información y clasificarlos como variables o constantes.

- La cantidad de kilómetros de la Vuelta Ciclistica San Juan.
- El saldo de la tarjeta sube.
- El numero 3
- La altura de un edificio.
- La distancia de la Luna y la Tierra en el fenómeno Luna de Sangre del 2019.
- La cantidad de habitaciones de un hotel.
- El ancho de banda de una conexión a internet.
- Cantidad de suscriptores de un canal de Youtube.
- El precio de una prenda en una tienda.
- La cantidad de pasajeros de un colectivo de larga distancia.
- El resultado de hacer una apuesta (ganar o perder).
- La letra 'a' escrita en mayúscula.
- La capital de un país.
- La capital de Argentina.

### Ejercicio 2

Indique, cuáles de los siguientes identificadores son inválidos. Justifique.

Practico pseudocódigo

7siete

2016\_ejercicios

IDentificador

dato\_1

\*\*expresiones\*\*

cui-da-do

¡Recreo!

\* \_ \*

VaLiDo

### Ejercicio 3

Escribir las siguientes expresiones aritméticas para que sean válidas en pseudocódigo:

$\frac{y^2a+b}{a-y} + \frac{2a+1}{5b}$	$\frac{5z+x}{x-y} - \sqrt{25x}$	$\frac{r + \frac{t}{2} + 3}{m}$	$\frac{2y \cdot x + b \cdot y}{a \cdot 4w - v \cdot y}$
--	---------------------------------	---------------------------------	---

**Observación:** analizar las restricciones sobre los operandos, en caso de ser necesario (el argumento de una raíz par debe ser cero o positivo, el denominador no puede ser nulo, etc) .

### Ejercicio 4

Evaluar las siguientes expresiones.

- 1)  $6 - 3 * 4 / 4 + 5 * 2$
- 2)  $2 * 3 + (6 \text{ div } 3 - 1) * 8$
- 3)  $6 \text{ resto } 3 - (9 / 4) * 2 + 1$
- 4)  $(5 / 2 * 3 - 1) = = 9 \text{ div } 2$
- 5)  $\text{Potencia}( (6 \text{ div } 3), 4)$
- 6)  $(7 \text{ div } 3) > \text{Potencia}((4 / 4), 12) * (2 + 10 \text{ resto } 5)$

- 7)  $10 \leq 3 + 4 - 2 * \text{raíz}(9 \text{ resto } 5)$   
 8)  $\text{no} ( (15 > \text{potencia}(7, 2)) \text{ o } (43 - 8 * 2 \text{ div } 4) \neq (3 * 2 \text{ div } 2))$   
 9)  $\text{no} (15 > \text{potencia}(7, 2)) \text{ o } (43 - 8 * 2 \text{ div } 4) \neq (3 * 2 \text{ div } 2)$   
 10)  $(15 \geq \text{potencia}(2 * 3, 2)) \text{ y } (8 > 3) \text{ y } (15 > 6) \text{ o } \text{no} (7 * 3 < 5 + \text{potencia}(2 * 2 \text{ div } 3, 2))$   
 11)  $\text{no} ((7 * 3 \text{ div } 2 * 4 < 20) \text{ o } (15 / 2 * 6 \leq 15 * 2 / 17))$

### Ejercicio 5

Escribir expresiones lógico-matemáticas para expresar los siguientes conceptos

Concepto	Expresión
N es positivo	
N es un número par	$N \text{ resto } 2 = 0$
N es múltiplo de 7 y múltiplo de 4	
N no es múltiplo de 5 ni múltiplo de 8	$\text{no} (N \text{ resto } 5 = 0) \text{ y } (N \text{ resto } 8 \neq 0)$
N es múltiplo de 3 pero no es múltiplo de 8	
N es positivo o impar	
N no es divisor de P	
N tiene al menos dos dígitos	
N tiene exactamente tres dígitos	

### Ejercicio 6

Complete el cuadro especificando el **tipo** solicitado:

Expresión	Operador/es que clasifica a la expresión	Operandos	Resultado
$(a-b) > (x+z)$			
$(m/n) + x * z - 6$			
$((k-x) == 58) \text{ y } (k > 0)$			
$H \geq 'h'$			
$((((a*c)+58)/F) > 0) \text{ o } M$			
$Q/2 - 68 + q * 5 - J$			

### Ejercicio 7

Complete el cuadro. Considerando los valores que se presentan para el cálculo.

a)  $T = -2$   $C = 7$   $K = 'c'$

Expresión	Tipo de expresión	Resultado	Tipo de resultado
$3 * T \geq C * 2$			
$(K \leq 'f') \text{ o } ((T * 2) > C)$			
$T * (-1) + 3 * C - T * 4$			

b)  $P = 6$   $Q = -5$   $B = \text{falso}$

Expresión	Tipo de expresión	Resultado	Tipo de resultado
$P+7 > Q*3+2$			
$8 - Q*3*2 + P/2$			
no B y $(Q (-1) < P+2)$			

c)  $M = 8$   $N = -3$   $Z = \text{verdadero}$   $A = 'm'$

Expresión	Tipo de expresión	Resultado	Tipo de resultado
$(A \geq 'm') \text{ o } ((M / 2) > 3)$			
$Z \text{ y } (((-3) * N) \leq M*2)$			
$M * (-1) + ((M * N)/4)$			

### Ejercicio 8

Dadas las siguientes declaraciones:

CINCO = 5

entero  $a = 8$ ,  $b = 15$

real  $x = 0,005$ ,  $m = 2$ .

¿qué valores se obtienen al evaluar las siguientes expresiones?

- 1)  $a == b \text{ resto CINCO}$
- 2)  $16 \text{ div CINCO} < \text{CINCO resto } 2$
- 3)  $((\text{CINCO} + b - 1) < a) \text{ o } (b \geq -b * a) \text{ y } (a * 2 \leq 10)$
- 4)  $x * m * 10 == \text{potencia}(10,2)$
- 5)  $(b \text{ resto } a) \text{ div CINCO}$
- 6)  $\text{no } ((x * a) > (m / b))$

### Ejercicio 9

Suponiendo que A, B, C son números enteros, proponga expresiones que verifiquen:

- 1) C es mayor que A o B
- 2) A está entre B y C
- 3) A es menor que B y C, y además B es menor que C y mayor que A
- 4) A es menor que B mas C, pero mayor que ambos y además no es par

### Ejercicio 10

Expresa los siguientes enunciados como una expresión:

- a) La quinta parte de una variable entera X mas 150
- b) Un número más su doble es par
- c) El módulo 3 de un número entero es mayor o igual a la diferencia entre la variable M y 29
- d) El 20% de incremento a la variable sueldo
- e) El promedio de las variables M y N

- f) El 15% de descuento de la variable importe no supera el valor 124

### Ejercicio 11

Construir la expresión correspondiente a cada uno de los siguientes enunciados lógicos A.

Datos de alumno: **Reg-alumno, nombre, cod-carrera y cant-mat-rend**

- a) Registro de alumno igual a diez mil y cuyo código de carrera sea 'I' ó 'G'.
- b) Alumnos con código de carrera igual a 'G' y que tengan entre cinco y ocho materias rendidas.
- c) Alumnos cuyo código de carrera es igual a 'A' con al menos 10 materias rendidas

B.

Datos de artículo: **Código-artículo, nombre-artículo, precio y stock**

- a) Código de artículo menor que cien y mayor que cuarenta cuyo precio no supere los noventa pesos.
- b) Código de artículo comprendido entre 80 y 150 cuyo stock no sea menor a 25.
- c) Nombre de artículo que cueste como máximo 38 pesos.

C.

Datos de Video en Youtube: Nombre del propietario (quién publica), fecha de publicación, cantidad de visualizaciones, cantidad de Me gusta y cantidad de No me gusta.

- a) Videos que superaron las 500 millones de visualizaciones.
- b) Videos que registran la misma cantidad de Me gusta y No me gusta.
- c) Videos que no superen los 30000 me gusta el día 04/09/18.
- d) Videos publicados por "Juan Pérez" con no más de 1 millón de visualizaciones.

D.

Datos de empleado: **Código de empleado: Num\_empl, Edad: E, Provincia donde nació: Prov, Sexo: S** (codificada F:femenino M:masculino), **Sueldo básico: Sueldo y Años de antigüedad: Antig**

- a. Empleados hombres mayores de 35 años que tienen sueldo básico superior a \$500
- b. Empleados de sexo femenino mayores a 25 años o de sexo masculino entre 30 y 45 años.
- c. Empleados nacidos en San Juan con menos de 5 años de antigüedad.
- d. Empleados de sexo masculino nacidos en Mendoza con código de empleado inferior a 100.

E.

Datos de paciente: **Identificación de paciente: ID\_pac, Edad: E, Sexo: S** (codificada F: femenino M: masculino), **Estado civil: Est\_civil** (codificado S:soltero, C:casado, V:Viudo), **Especialidad médica atendida: Esp\_medica** (codificada 1:Clinica Médica, 2:Ginecología: 3:Pediatría 4:.... 10: Urología) **Código doctor que atiende: Cod\_dr** (codificado 1:Sanchez 2:Guzmán 3: Clavel 4:Perez .... 15: Jofré )

- a. Pacientes hombres mayores de 40 años que los atiende el doctor Jofré
- b. Pacientes de sexo femenino solteras de menos de 30 años y más de 20 años que necesitan atención en la especialidad de ginecología
- c. Pacientes del doctor Guzmán casados y con identificación de paciente que no sea superior a 950.
- d. Pacientes de la especialidad pediatría entre 0 y 4 años atendidos por el doctor Clavel.



F. Datos de Alumno:

**Nro inscripción del alumno:** *Num\_insc*, **Edad:** *E*; **Sexo:** *S* (codificada F: femenino, M: masculino), Estado civil: *Est\_civil* (codificado S:soltero, C:casado, V:Viudo) **Nota promedio de la secundaria:** *Nota\_prom*, **Código ingreso a la UNSJ:** *Cod\_ing* (codificado S: ingresó N: no ingresó)

- Alumnos ingresantes varones cuya edad sea entre 18 y 20 años
- Alumnos aspirantes solteros de sexo femenino con nota promedio superior a 6.50
- Alumnos ingresantes que no sean solteros con número de inscripción inferior a 1000.

## Ejercicio 12

Indique el contenido de las variables después de realizar las siguientes operaciones:

$M = Z$

$Z = M$

$Z = X$

$X = M$

$X = Z$

$M = X$

Teniendo en cuenta cada uno de los siguientes lotes de prueba:

- $X=12$ ,  $M=8$ ,  $Z=20$
- $X=3$ ,  $M=-2$ ,  $Z=5$
- $X=4$ ,  $M=6$ ,  $Z=-7$

## Construir y realizar el seguimiento de los siguientes algoritmos:

### Ejercicio 13

Escriba un algoritmo que permita pasar una distancia medida en metros a pies.

*Nota:* 1 metro equivale a 39.27 pulgadas y 12 pulgadas equivale a 1 pie.

### Ejercicio 14

Escriba un algoritmo que realice el cálculo de la hipotenusa de un triángulo rectángulo, en función de los catetos.

### Ejercicio 15

Realice un algoritmo que ingrese dos números enteros y muestre el resultado de cada una de las siguientes operaciones: suma, resta, multiplicación y división.

### Ejercicio 16

Un estudio biológico demostró que el número de sonidos emitidos por un grillo en un minuto, es en función de la temperatura ambiente expresada en grados Fahrenheit. Como resultado podría utilizarse al grillo como termómetro ambiental.

La fórmula que obtuvieron y determina esto es:  $T = N/4 + 40$  (T: temperatura en grados Fahrenheit; N: número de sonidos emitidos por el grillo)

Se pide realizar un algoritmo que:

Teniendo en cuenta el número de sonidos emitidos por el grillo diga la temperatura ambiental en grados Celsius o centígrados.

*Nota:* Para convertir grados Fahrenheit a grados Celsius o centígrados la fórmula es:  $^{\circ}\text{C} = (^{\circ}\text{F} - 32) \times 0,556$