

programación I

<clase> 2 = programas y datos </clase>



UNIVERSIDAD
NACIONAL DE CUYO

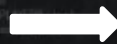


FACULTAD DE INGENIERIA
en acción continua...

Dra. Elina Pacini
Lic. Leandro Spadaro
Ing. Silvina Manganeli
Lic. Laura Noussan Lettry

programa

datos de
entrada



instrucción 1

instrucción 1

...

...

...

...

instrucción N



datos de
salida

¿en qué se diferencia con el algoritmo?
¿y el pseudocódigo no es un programa?

programa: elementos

```
//programa que calcula el área de un círculo
Algoritmo area_circulo
```

Definir radio, resultado Como Real

escribir "Ingrese el radio"

leer radio

resultado = $3.14 * \text{radio} * \text{radio}$

escribir "Area = ", resultado

FinAlgoritmo

instrucciones

identificadores

variables

constantes

comentarios

instrucciones

conjunto de acciones y componentes de un lenguaje básicos
para su funcionamiento

//programa que calcula el área de un círculo
Algoritmo area_circulo

Definir radio, resultado Como Real

escribir "Ingrese el radio"

leer radio

resultado = $3.14 * \text{radio} * \text{radio}$

escribir "Area = ", resultado

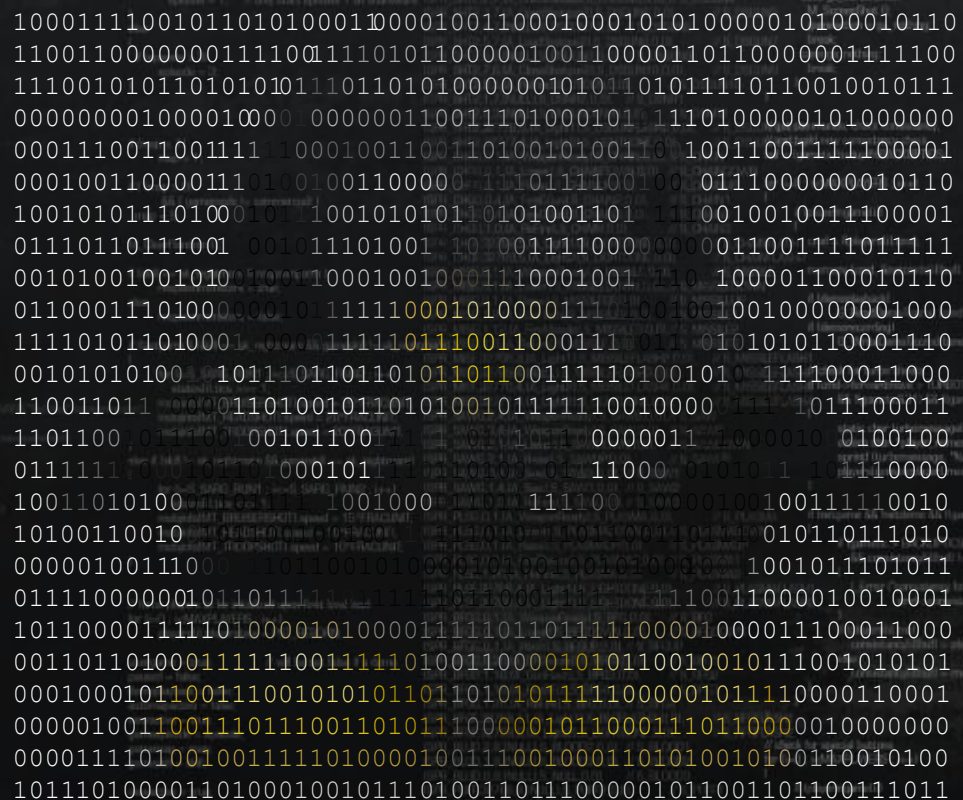
FinAlgoritmo

palabras
reservadas

expresiones

asignación

lectura
escritura



datos: clasificación

tipo

dimensión



complejidad

π

mutabilidad

* a desarrollar en mayor profundidad en unidades posteriores

tipo: numérico

enteros

- diferentes *tamaños*: de 1 a 8 bytes (en general).
- con y sin signo.
- se escriben sin separador de miles.
- ej.:
3782
-16513

reales

- diferentes *precisiones*: de 4 a 8 bytes (en general).
- se representan mediante un mecanismo de coma flotante (simil notación científica)
- el punto separa decimales.
- ej.:
-3.14
5.986E-9

tipo

tipo: lógico

verdadero

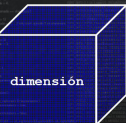
falso

true

false

tipo: caracteres

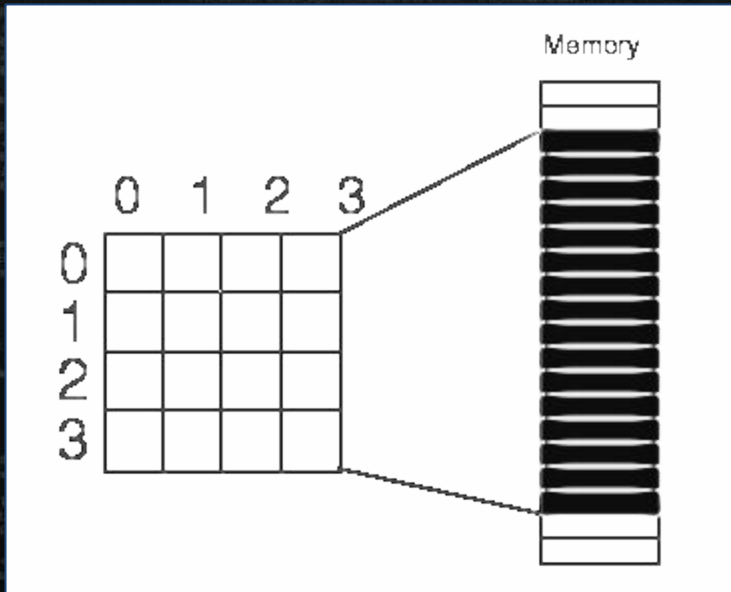
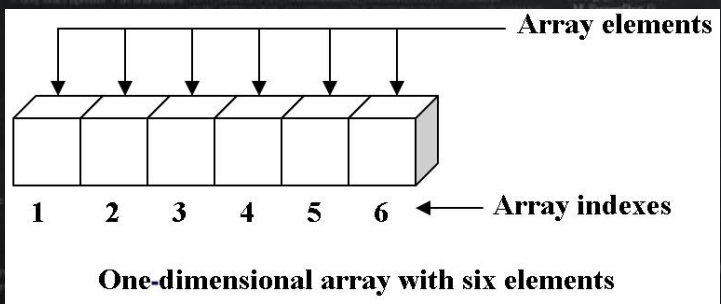
- se encierran entre *comillas* (simples o dobles).
- un *caracter* (unidimensional):
`8' o "8"
- una *cadena* (multidimensional):
"esto es una cadena"



dimensión: 1 o n

- **unidimensional:** valor único o escalar.
- **multidimensional:** más de un valor. Ej arreglos (vectores), matrices.
- **homogéneos:** siempre del mismo tipo.
- reserva **estática** y **contigua** de memoria.

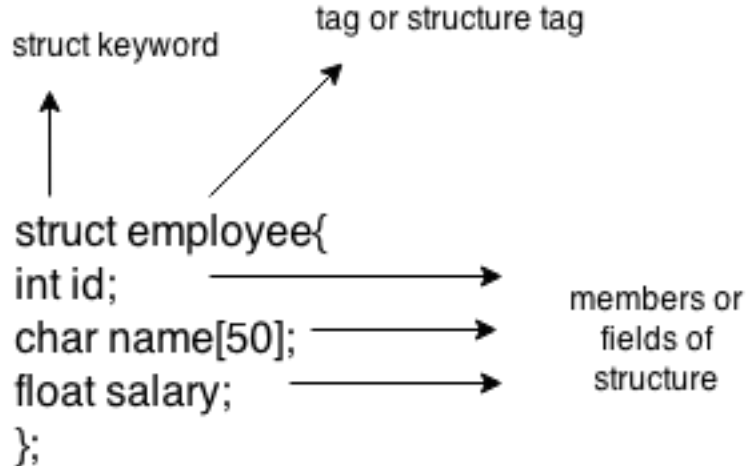
300516





complejidad

300516



```
employee pepe;  
pepe.id = 6;  
pepe.name = 'josé';  
pepe.salary = 15000.0;
```

- ***simples:*** primitivos del lenguaje. representan un valor único de su tipo.
- ***complejos:*** composición de datos primitivos potencialmente distintos.
- ej.: estructuras, clases, tuplas, enumeraciones, etc.

π

variables

- mutables
- exigen identificación válida
- definición pseudocódigo/PseInt:

<tipo_de_dato> <nombre_variable> [= <expresión de inicialización>]

Definir <nombre_variable> Como <tipo_de_dato>

- bool soy*cool = falso
- entero edad
- real ¥ = 1000
- cadena 8000 = 'ocho mil'
- cadena nick = "coronel"



π

constantes

- inmutables
- opcionalmente identificables
- definición pseudocódigo:
`const <tipo_de_dato> <nombre_constante> = <expresión>`
- Algunos lenguajes no admiten su definición (ej. PseInt no permite constantes indentificadas)
- ej.
894
'hola mundo'
true
const real PI = 3.1415

expresiones

combinación de constantes, variables, funciones y operadores
que producen un nuevo valor

//programa que calcula el área de un círculo
Algoritmo area_circulo

Definir radio, resultado Como Real

escribir "Ingrese el radio"

leer radio

resultado = 3.14 * radio * radio

escribir "Area = ", resultado

FinAlgoritmo

operador/es

operando/s

valor resultante

expresiones: tipos

±

aritméticas

≤

relacionales

&

lógicas

C

propias del
lenguaje

aritméticas: operadores

+ (suma y unitario positivo)

- (resta y unitario negativo)

* (multiplicación)

/ (división)

// o floor() (división entera)

^ o ** o pow() (exponenciación)

% o mod() (módulo o resto)

++ -- (incremento y decremento)

ej.
resultado*8.5
10%2*5
sueldo++

operandos:
tipo numérico
resultado:
tipo numérico

precedencia operadores

orden de evaluación de las operaciones de una expresión que contenga dos o más operadores

- siguen las **convenciones aritméticas**
- por definición **de izquierda a derecha**
- ej $1 + 10 / 2$ vs $(1 + 10) / 2$

1. de la más interna a la más externa
2. operador ()
3. operadores ++, --, + y - (unitarios)
4. operador de potencia (^) y raíz
5. operadores *, /, % (producto, división, módulo)
6. operadores +, - (suma y resta).

relacionales

{expresión} **operador** {expresión}

operandos:

todos los tipos*

resultado:

tipo lógico

ej.

A * 2 == 16

'1' != 'uno'

< (menor que)

> (mayor que)

=, == (igual que)

<= (menor o igual que)

>= (mayor o igual que)

<>, != (distinto de)

a) 67 < 'C'

b) 'a' > 'A'

c) falso > verdadero

d) 100 <= x < 1000

e) a > 'A'



Lógico: negación

[no o !] {expresión}

a	no a
VERDADERO	FALSO
FALSO	VERDADERO

operando:
tipo lógico
resultado:
tipo lógico

ej.
no ($X > 0$)
! valor

Lógico: conjunción

{expresión} [y o &] {expresión}

a	b	a y b
VERDADERO	VERDADERO	VERDADERO
VERDADERO	FALSO	FALSO
FALSO	VERDADERO	FALSO
FALSO	FALSO	FALSO

operandos:
tipo lógico
resultado:
tipo lógico

ej.
no (X > 0 y Z > 1)
falso & cualquier_cosa
x >= 100 y x < 1000

lógico: disyunción

{expresión} [o ó |] {expresión}

a	b	a o b
VERDADERO	VERDADERO	VERDADERO
VERDADERO	FALSO	VERDADERO
FALSO	VERDADERO	VERDADERO
FALSO	FALSO	FALSO

operandos:
tipo lógico
resultado:
tipo lógico

ej.
(X > 0 y Z > 1) o Y = 3
ser | no_ser

precedencia general

1. ()

2. ++, --, + (unitaria), - (unitaria), !

+

3. *, /, %

4. +, - (suma, resta)

5. <, <=, >, >=

<

6. == (igual a), != (no igual a)

7. & (y lógica)

8. | (o lógica)

&

9. = (asignación)

investigar **siempre** las reglas propias del lenguaje