



Informática

Curso 2024

Facultad de Ingeniería

Universidad de la República

Programa Informático

- Se compone de:
 - Los **datos** que forman parte del problema a resolver o que se requieren para resolverlo.
 - Las **acciones** necesarias para resolver el problema, y que procesan los datos.

Diseñar un Algoritmo

- Un ***algoritmo*** es un procedimiento detallado paso por paso para resolver un problema.
- Las **instrucciones** deben ser **claras y sin ambigüedades** y lo bastante específicas para ejecutarse y **terminarse** en un número **finito de pasos**.
 - Andar en bicicleta
 - Receta de cocina
 - Obtener el máximo de una lista de números

Representar el algoritmo formalmente

Pseudo-código

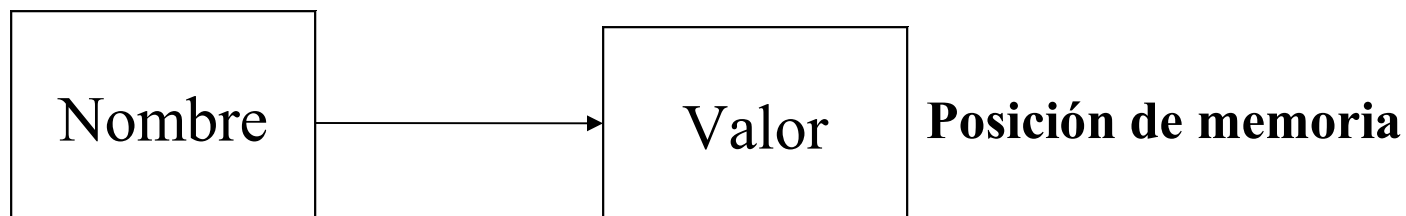
- El Pseudo-código es una representación de un programa en un lenguaje natural pero con formalidades propias de un lenguaje de programación
- Se tratará de escoger uno que ofrezca las mismas estructuras que el lenguaje en el que se prevé hacer la programación

Variables

- Son datos cuyo valor asociado o significado puede cambiar durante la ejecución del programa.
- Su cambio de valor no es arbitrario, sino producto de la ejecución de ciertas sentencias en el programa.
- Son posiciones de memoria a las que asignamos un nombre y a través de las cuales podremos almacenar y recuperar datos.

Variables

- Por cada variable se reserva una posición en la memoria donde se aloja su valor corriente.
- Tal posición es solo accedida a través del nombre de la variable.



Variables

- Son introducidas mediante declaraciones de la forma:

```
<variable> = <valor>;
```

- Ejemplo: `x = 3.1416;`

Condiciones

- Los algoritmos con frecuencia presentan situaciones en las que se deben proporcionar acciones alternativas que pueden o no realizarse, dependiendo de los datos de entrada, reflejándose el cumplimiento o no de una determinada condición.

Condiciones

- Diseñar un algoritmo para calcular el salario semanal de un empleado que trabaja por horas.
- La empresa paga la hora trabajada el doble de la hora normal por todas las horas trabajadas mayores a 40.

LEER(pago_por_hora)

LEER(horas)

SI horas > 40

ENTONCES

$$\text{paga} = \text{pago_por_hora} * 40 + 2 * \text{pago_por_hora} * (\text{horas} - 40)$$

SINO

$$\text{paga} = \text{pago_por_hora} * \text{horas}$$

FIN_SI

Condiciones

- Las condiciones nos permiten hacer **preguntas** sobre el estado de los **datos** empleados en un programa
- De acuerdo a si las respuestas son **satisfactorias** o no, pueden **tomarse decisiones** sobre las acciones a desarrollarse.
- Las respuestas posibles a las preguntas sobre el estado de los datos pueden ser solo dos:
 - **verdadero** (true), o **falso** (false)
 - **1** o **0**

Condiciones

Expresiones Lógicas

- Son expresiones que solamente pueden tomar uno de dos valores, **verdadero** o **falso**
- Los operandos de una expresión lógica pueden ser:
 - **Expresiones relacionales:** que comparan dos valores utilizando operadores relacionales y determinan si existe o no una cierta relación entre ellos.
 - **Expresiones lógicas:** que se relacionan entre si mediante operadores lógicos.

Condiciones

Expresiones Lógicas

■ Expresiones relacionales

□ Forma general

- Exp_1 *operador_relacional* Exp_2
- Exp_1 debe ser de igual tipo que Exp_2
 - Numéricos
 - Caracteres

□ Operadores relacionales

- Menor que <
- Menor o igual que <=
- Mayor que >
- Mayor o igual que >=
- Distinto que != ~=
- Igual que ==

Condiciones

Expresiones Lógicas

Operadores lógicos:

NOT

AND

OR

Condiciones

Expresiones Lógicas

■ Simples

- $2+2==4$
- $5>6$
- 3 es múltiplo de 8

■ Complejas

- $(5>3) \text{ AND } (3>1)$
- (5 es múltiplo de 2) AND (4 es múltiplo de 2)
- $(4>5) \text{ OR } (5>4)$
- $(5\sim=6) \text{ OR } (5==7)$

Condiciones

- Expresiones lógicas:
 - Operandos
 - Variables
 - Operadores
 - Operadores relacionales
 - Operadores lógicos

Condiciones

Expresiones Lógicas

OP1	OP2	AND
V	V	V
V	F	F
F	V	F
F	F	F

OP1	OP2	OR
V	V	V
V	F	V
F	V	V
F	F	F

Condiciones

Expresiones Lógicas (tablas de verdad: NOT)

OP1	NOT
V	F
F	V

Instrucciones

- **Asignación**
 - Asignan valores a variables.
- **Estructuras de Control**
 - Organizan el curso (flujo) de ejecución.
 - Pueden basarse en Condiciones

Asignación

- El objetivo de una sentencia de asignación es cambiar el valor almacenado en una variable.

`<variable> = <expresión>`

- Ejemplos:

`x = 3 + 4`

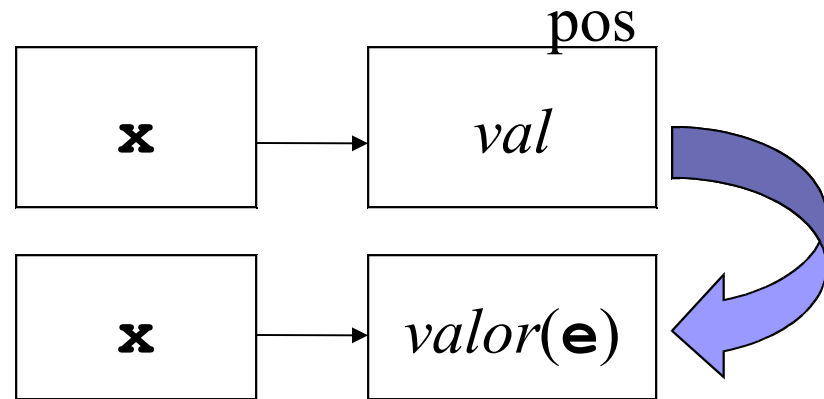
`x = x + (3.0 / 5.9)`

`c = 'a'`

Asignación

- Una asignación $x = e$ es ejecutada siguiendo estos pasos:

- 1.- Se evalúa la expresión e
- 2.- Se reemplaza el valor almacenado en la posición de memoria pos , correspondiente a la variable x , por el valor de e .



Asignación

- La ocurrencia de una variable en el lado izquierdo de una asignación denota la posición de memoria donde almacenar el valor resultante de evaluar la expresión en el lado derecho.

$$x = 1;$$

- La ocurrencia de una variable en el lado derecho de una asignación denota su valor actual.

$$y = x + 1;$$

Asignación

- Una misma variable puede aparecer en la parte izquierda y derecha de una asignación.

$$x = x + 1$$

- Esto NO debe interpretarse como una ecuación matemática!
- Sólo significa que estamos usando el valor actual de la variable x para calcular su nuevo valor.