

Fundamentos de Informática: Introducción

Oier Lopez de Lacalle y Patxi Ramírez

Languages and Information Systems

2021/2022



GASTEIZKO
INGENIARITZAKO
UNIBERTSITATE ESKOLA
ESCUELA UNIVERSITARIA
DE INGENIERÍA
DE VITORIA-GASTEIZ

Contents

- 1 **Conceptos Básicos**
- 2 **Desarrollo de Programas**
- 3 **Programación: Introducción**

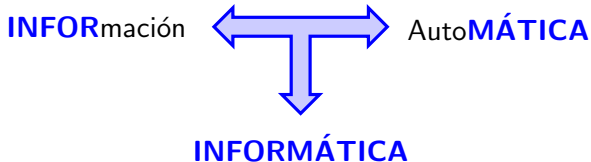
Contents

- 1 Conceptos Básicos**
- 2 Desarrollo de Programas
- 3 Programación: Introducción

Informática



Ciencia encargada del tratamiento **automático** de la **información**

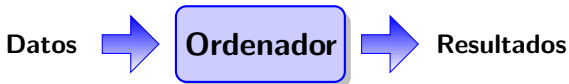


Ordenador



Máquina electrónica digital que realiza operaciones aritmético-lógicas con los datos de entrada hasta que obtiene el resultado

- Máquina que ejecuta **algoritmos**
- La ejecución o procesamiento de un **algoritmo** supone la **transformación** de **información de entrada** (*datos*) en **información de salida** (*resultados*)



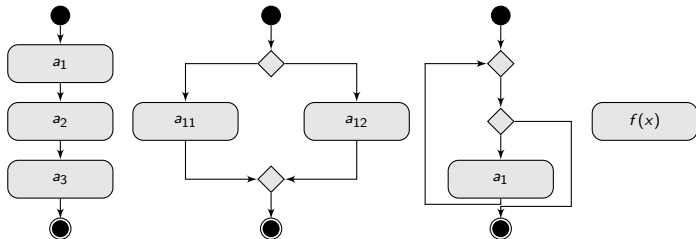
Programa



Program

Secuencia finita de instrucciones, reglas o pasos que describen de forma precisa las operaciones que un ordenador debe realizar para llevar a cabo una tarea en tiempo finito.

Structures



Lenguaje de programación y Lenguaje máquina



Lenguaje máquina

Lenguaje cuyas instrucciones son ejecutables por el hardware de un ordenador.



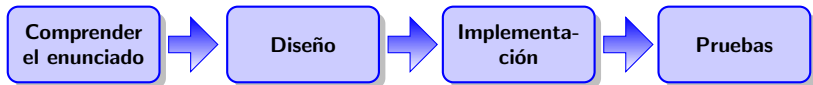
Lenguaje de programación

Lenguaje formal que especifica una serie de instrucciones para que una computadora produzca diversas clases de datos. Las instrucciones deben ser comunicadas en lenguaje máquina.

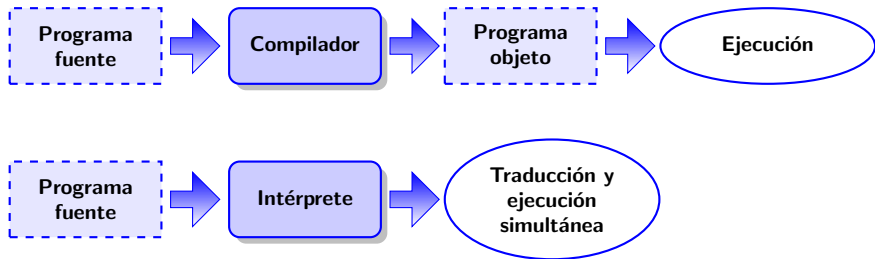
Contents

- 1 Conceptos Básicos
- 2 Desarrollo de Programas**
- 3 Programación: Introducción

Pasos en el desarrollo de un programa



Compilación e Interpretación



Contents

- 1 Conceptos Básicos
- 2 Desarrollo de Programas
- 3 Programación: Introducción**

Lenguaje de Programación

MatLab

Utilizaremos el lenguaje y el entorno de programación MatLab. MatLab es la abreviación de **Matrix Laboratory**.

- Una herramienta para hacer cálculos matemáticos que utiliza como elemento básico la matriz.
- Un lenguaje de programación:
 - Interactivo
 - Órdenes (comandos)
- Plataforma de desarrollo (toolboxes)

Ventajas de MatLab

- Su programación requiere (teóricamente) menos tiempo que otros lenguajes como C, Java, etc.
- Utiliza un lenguaje más cercano a la matemática.
- Permite definir fácil y rápidamente nuevas funciones que se incorporan a Matlab (mediante el toolboxes)
- Grandes capacidades gráficas.
- Ampliamente usado en aplicaciones de ingeniería

- Análisis de datos
- Polinomios
- Gráficos 2D
- Gráficos 3D
- Ajuste de curvas
- Interpolación
- Análisis numérico

Variables



Variable

Magnitud o información manejada por un algoritmo a la que se da un nombre y registra un valor.

Variables: reglas y convenciones para nombrar variables

Los nombres de variables deben seguir las siguientes convenciones:

- 1 Comenzar por una letra
- 2 No puede contener espacios
- 3 Nombres significativos (nmemotécnico)
- 4 No se pueden usar palabras reservadas al lenguaje de (aunque los nombres de las funciones pueden ser usadas, no es recomendable)
- 5 Si contienen más de un palabra, cada palabra comenzará en mayúscula, excepto la primera letra de la variable (**camelCase**) - e.g. *fuerzaGravitacional*

Asignación



Assignment

Operación por la cual se establece un nuevo valor a una **variable**.

Syntax

nombreVariable = expresión

Example

resultado = 6 + 78

Asignación: consideración

Aviso

La asignación debe realizarse siempre en el mismo sentido. El valor de la derecha o el resultado de la expresión se guardará en la variable de la izquierda. No se puede hacer de otra manera.

~~$4 + 5 \Rightarrow \text{resultado}$~~

~~$\sin(\pi) \equiv 0$~~

Operaciones aritméticas

Operador	Description
+	sum
-	substraction
*	product
/	division. 4/2 - cuatro dividido por dos, resultado 2
^	exponencial 5^2 - five al cuadrado, resulta 25

Cambio de signo

Cuando el operador `-` se asocia con una única variable o expresión, por ejemplo `-valueVar`, significa cambio de signo, i.e., devuelve el resultado de cambiar de signo al valor guardado en `valueVar`.

Prioridad de operadores

Las expresiones aritméticas pueden ser ambiguas

¿Cómo interpretamos la expresión $a * b + c$?

- $(a * b) + c$
- $a * (b + c)$

Para evitar el problema se definen prioridades entre los operadores.

Prioridad del operador		
1	()	parentesis
2	^	exponencial
3	-	change of sign
4	*, /	producto and división
5	+, -	sum and sustracción

Echo

Aviso

Matlab produce el resultado de la operación realizada en pantalla. Para evitarlo hay que añadir el operador ; al final del enunciado.

Syntax

instrucción;

Example

```
resultado = 6 + 78;
```

Tipo de datos

Tipo de dato	Descripción	Ejemplo
Entero	Número entero	0, 352, -342, 20050
Real	Número real	4.22, -23.44, 342.015
Lógico	<i>true</i> ó <i>false</i>	0 (<i>false</i>), 1 (<i>true</i>)
Carácter	Letra ó símbolo	'A', 'a', '\$', '1', '+'
Strings	Texto	"Esto es un string"

Tipo de datos: Consideraciones

Tipo de dato por defecto

En Matlab, por defecto, todos los valores numéricos se consideran valores reales (**double**)

Aviso

El tipo de dato determina el rango de valores permitidos. No se obtendrá ningún valor mayor o menor determinado por el rango.

Funciones básicas

Función	Descripción
help	Muestra el texto de ayuda para la funcionalidad especificada con el nombre escrito justo después del comando
whos	Lista las memorias del espacio de trabajo con información de su tamaño
clear	Borra las variables del espacio de trabajo
clc	Limpia la consola

Funciones básicas

Función	Descripción
fix, floor, ceil	Funciones de conversión de números reales a enteros, redondeando o truncando. Por ejemplo, podríamos usar $fix(5/2)$ para convertir el resultado de la división en un valor entero.
rem	Calcula el resto de la división entre enteros. Por ejemplo, usaremos $rem(n, 2)$ para saber si n es par o no.
abs	Obtiene el valor absoluto de un número.
sqrt	Obtiene la raíz cuadrada de un número
sin	Obtiene el seno de un ángulo expresado en radianes
cos	Obtiene el coseno de un ángulo expresado en radianes

Lectura de datos

Solicitar datos

Podemos solicitar al usuario que nos dé algunos datos usando la función **input**

Sintaxis

input (*mensaje*)

Ejemplo

```
cantidad=input('Introduce el numero de monedas');  
nombre=input('Como te llamas?');
```

Lectura de datos: consideración

Aviso

El texto de entrada debe escribirse entre comillas dobles.

```
name=input('Como te llamas? ');  
"Mikel"
```

Mostrar valores

Mostrar valores

La función **disp** es la forma más sencilla de escribir en pantalla el resultado de una operación o el contenido de una variable.

Sintaxis

disp (*dato*)

Ejemplo

```
disp(4^7);  
disp(cantidad);
```

Escribir mensajes en pantalla

Escribir mensajes

Podemos imprimir mensajes en la consola usando la función **fprint**

Syntax

fprintf (*mensaje*)

Ejemplo

```
fprintf('Numero incorrecto de monedas');
```

Salida formateada

Escribir salida formateada

Podemos formatear la salida usando la función **printf**. Para ello, se utilizan marcadores de posición formateados (representados por %) en el mensaje y se deben incluir los datos utilizados para llenar las celdas.

Sintaxis

fprintf (*mensaje, variables*)

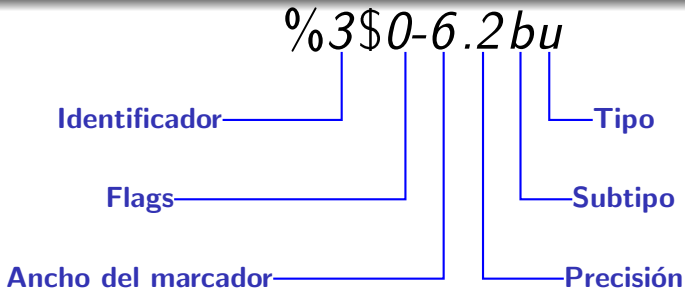
Ejemplo

```
fprintf('Numero de monedas: %d'  
, cantidad);
```

Salida formateada: marcadores de posición

Formato

Los **marcadores de posición** especifican dónde se imprimirán los valores de las expresiones que están después de la cadena. Los marcadores de posición y su formato se expresan de la siguiente manera:



Salida formateada: opciones (I)

Identificador

Este campo numérico determina qué variables deben usarse. **No es obligatorio**, pero si se incluye, debe usarse antes del signo **\$**. Indica la posición del número en la lista de variables que se utilizarán para formar la pila.

Salida formateada: opciones (II)

Flags

Especifican algunas conversiones que se llevan a cabo para generar el texto de salida.

Flag	Descripción
'-'	Justificar a la izquierda.
'+'	Imprimir siempre el de signo (+ o -) para cualquier valor numérico.
' '	Insertar un espacio antes del valor.
'0'	Rellenar el ancho del campo con ceros antes del valor.
'#'	<ul style="list-style-type: none">• Para %o, %x, or %X, imprime los prefijos 0, 0x, o 0X.• Para %f, %e, or %E, imprime siempre en decimales.• Para %g or %G, no elimina los ceros finales ni el punto decimal.

Salida formateada: opciones (III)

Ancho del marcador

Número mínimo de caracteres utilizados para visualizar los datos. Si los datos son más cortos, los espacios (o ceros) se agregarán antes de los dígitos (o el signo).

Precisión

Numero de decimales que se utilizarán para representar números reales.

Salida formateada: opciones (IV)

Subtype

Sirve para imprimir un valor real como su valor octal, decimal o hexadecimal. Se puede usar precisión simple (**t**) o doble (**b**).

Salidad formateada: tipos

Carácter de conversión	Tipo de dato
%d or %i	Número entero
%u	Número natural
%o	Número natural, Base 8
%x	Número natural, Base 16, letra minúscula
%X	Número natural, Base 16, letra mayúscula
%f	Número real
%e	Número real, notación exponencial, letra minúscula
%E	Número real, notación exponencial, letra mayúscula
%g	El formato más compacto entre las opciones %e o %f
%G	El formato más compacto entre las opciones %E o %F
%c	Carácter
%s	Texto

Salida formateada: caracteres especiales

Character	Description
\t	Tabulador
\n	Salto de línea
% %	Porcentaje
"	Comillas simple

Scripts



Script

| Secuencia de instrucciones especificadas en un fichero

```
% This is is an script
```

```
price = 75;
```

```
result = price * 1.21
```

code.m

Comments



Comment

Texto descriptivo que nos ayuda a comprender el código.

MatLab no tiene en cuenta este texto.

Sintaxis

`% texto`

Ejemplo

`%Ejemplo de comentario`

Fundamentos de Informática: Introducción

Oier Lopez de Lacalle y Patxi Ramírez

Languages and Information Systems

2021/2022



GASTEIZKO
INGENIARITZAKO
UNIBERTSITATE ESKOLA
ESCUELA UNIVERSITARIA
DE INGENIERÍA
DE VITORIA-GASTEIZ