

Programación

PLG
Planning and Learning Group

Universidad Carlos III de Madrid

Tema 1: Introducción a la Programación

¿Qué es Programar?

- ▶ Según el diccionario de la RAE:
Inform. Elaborar programas para la resolución de problemas mediante ordenadores
- ▶ Otra definición informal pero más explicativa:
Proporcionar a un ordenador un conjunto de datos y unas instrucciones sobre lo que se debe hacer con esos datos con el objetivo de resolver algún problema

¿Qué es Programar?

- ▶ **Objetivo:** Resolver un problema
- ▶ **Cómo resolverlo:** Utilizando un conjunto de instrucciones que le indican al ordenador paso a paso cómo resolver el problema

Algoritmo

Lista bien definida, ordenada y finita de operaciones que permite hallar la solución a un problema

Ejemplo de Algoritmo

- ▶ Algoritmo para cambiar la rueda de un coche
- ▶ Datos: rueda pinchada, y ubicación del gato, de la rueda de repuesto y de la llave inglesa

Ejemplo de Algoritmo

- ▶ Algoritmo para cambiar la rueda de un coche
- ▶ Datos: rueda pinchada, y ubicación del gato, de la rueda de repuesto y de la llave inglesa
 - ▶ PASO 1. Aflojar los tornillos de la rueda pinchada con la llave inglesa
 - ▶ PASO 2. Colocar el gato mecánico en su sitio
 - ▶ PASO 3. Levantar el gato hasta que la rueda pinchada pueda girar libremente
 - ▶ PASO 4. Quitar los tornillos
 - ▶ PASO 5. Quitar la rueda pinchada
 - ▶ PASO 6. Poner rueda de repuesto
 - ▶ PASO 7. Poner los tornillos y apretarlos ligeramente
 - ▶ PASO 8. Bajar el gato hasta que se pueda liberar
 - ▶ PASO 9. Sacar el gato de su sitio
 - ▶ PASO 10. Apretar los tornillos con la llave inglesa

Resolución de Problemas

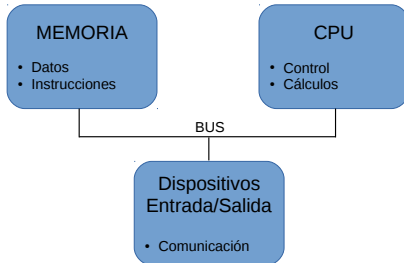
- ▶ El objetivo cuando se escribe un programa es resolver un problema
- ▶ Pasos generales para resolver problemas
 - ▶ Entender el problema
 - ▶ Si el problema es grande partirlo en piezas manejables
 - ▶ Diseñar una solución (un algoritmo)
 - ▶ Implementar la solución
 - ▶ Probar la solución y reparar errores

Buen algoritmo

- ▶ Preciso
- ▶ No ambigüo
- ▶ Correcto
- ▶ Eficiente
- ▶ Mantenable

Arquitectura Básica de un Ordenador

- ▶ Hardware vs. Software
- ▶ La inmensa mayoría de los ordenadores tiene una arquitectura de Von Neumann (propuesta inicialmente por Eckert y Mauchly).
 - ▶ Datos e instrucciones comparten memoria
 - ▶ Cada espacio (celda) de la memoria se identifica con un número llamado dirección



Componentes de la Arquitectura de un Ordenador

- ▶ Unidad Central de Procesamiento (CPU)
 - ▶ Se encarga fundamentalmente de ejecutar las instrucciones y coordinar el resto de elementos
- ▶ Memoria
 - ▶ Almacena los datos, las instrucciones y los resultados
 - ▶ Clasificación: principal/secundaria, permanente/volátil, acceso directo/secuencial
- ▶ Dispositivos de Entrada/Salida
 - ▶ Para proporcionar los datos e instrucciones y recibir los resultados
- ▶ Bus de Datos
 - ▶ Para compartir la información entre los componentes anteriores

Tipos de Software

- ▶ Software de Sistema
Proporciona control sobre el hardware y sirve de base a las aplicaciones
- ▶ Software de Aplicaciones
Programas con finalidades específicas, resuelven un problema o familia de problemas determinados
 - ▶ Ofimática
 - ▶ Contabilidad
 - ▶ Diseño
 - ▶ Juegos

Lenguaje de Programación

- ▶ Programa = datos + instrucciones
- ▶ Para comunicarle al ordenador el programa se usa un lenguaje de programación
- ▶ Los ordenadores no entienden lenguaje natural
- ▶ ¿Cómo decimos al ordenador lo que tiene que hacer?:
Escribiendo un programa en un lenguaje de programación determinado, para implementar ese algoritmo

Tipos de Lenguajes de Programación

- ▶ Lenguaje Binario o Código Máquina
 - ▶ Con 0 y 1
- ▶ Lenguaje de bajo nivel
 - ▶ Instrucciones básicas (mover datos, sumar, ...)
- ▶ Lenguajes de alto nivel
 - ▶ más cercanos al lenguaje natural
 - ▶ ... aunque tampoco demasiado

Código Máquina

- ▶ Único lenguaje que entiende el ordenador
- ▶ Datos e instrucciones se codifican mediante conjuntos de 0 y 1
- ▶ El más rápido: hablamos al ordenador en su propio lenguaje
- ▶ Muy propenso a errores, muy complicado

Representación de Datos e Instrucciones

- Información codificada en binario
- La memoria está compuesta por bits que sólo pueden valer 1 o 0
- Los bits se agrupan en bytes (8 bits)
- Cada celda de memoria contiene entre 1 y 8 bytes y almacena un dato, un resultado o una instrucción
- ejemplo:

0	0	1	0	1	1	1	1	1	→	95
1	0	0	0	0	0	1	1	1	→	7
2	1	0	0	1	1	0	1	0	→	sumar
3	0	1	1	0	0	1	1	0	→	102

Lenguajes de Bajo Nivel

- ▶ Se usa un traductor para convertir desde un lenguaje textual a código máquina
- ▶ El traductor es un programa que le dice al ordenador cómo realizar la traducción
- ▶ Nace el lenguaje ensamblador: cambiamos 0 y 1 por texto, pero sigue siendo poco intuitivo
- ▶ Depende totalmente del tipo de procesador

Lenguajes de Alto Nivel

- ▶ Intentan acercar el lenguaje de programación al lenguaje humano
- ▶ Luego el ordenador se encargará de traducir
- ▶ Ideal: poder usar lenguaje natural
- ▶ Existen más de 300 (unos 2400 con dialectos)
- ▶ Los pioneros incluían conceptos como:
 - ▶ Variables: no es necesario gestionar los datos directamente en la memoria
 - ▶ Estructuras de datos complejas
 - ▶ Nuevas instrucciones, distintas de las que proporciona el ordenador
- ▶ Historia:

<http://manuelpereiragonzalez.blogspot.com/2009/09/historia-de-la-informatica-los.html>

Uso de Algunos Lenguajes



Compilación e Interpretación

La traducción desde un lenguaje de programación a binario se puede hacer de dos formas:

- ▶ Todo a la vez: compilador
 - ▶ Ejecución más rápida
- ▶ Instrucción a instrucción: intérprete
 - ▶ Ejecuta aunque haya errores en el código
 - ▶ Permite cambios “en caliente”

Paradigmas de Programación

Tres formas principales de darle instrucciones al ordenador

- ▶ Programación Imperativa
 - ▶ Se describen los pasos necesarios para solucionar el problema
 - ▶ Ejemplos: C, Java, Visual Basic
- ▶ Programación Funcional
 - ▶ Las instrucciones se dan mediante funciones “matemáticas” que transforman los datos
 - ▶ Ejemplos: LISP, Erlang, Haskell
- ▶ Programación Lógica
 - ▶ Se describe el problema pero no se dan instrucciones: se resuelve mediante inferencia lógica
 - ▶ Ejemplo: Prolog