Fundamentos de programación

1 Computadoras y programación

Grado en Ingeniería Informática
Grado en Ingeniería del Software
Grado en Ingeniería de Computadores

Material de la Prof.ª Mercedes Gómez Albarrán
Versión revisada y ampliada del material del Prof. Luis Hernández Yáñez
Facultad de Informática
Universidad Complutense

Informática, computadora, y programación

¿Qué es la Informática?

¿Qué es una computadora?

Hardware vs. software

¿En qué consiste la programación de computadoras?

El hombre y sus primeros intentos de procesar la información

Ábaco – 2.000 a.c.

El sumador de Pascal – mediados s.XVII

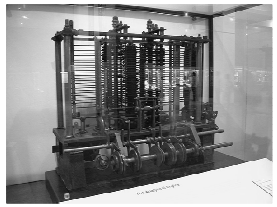
La *Pascalina* (Wikipedia)

La calculadora de G.W. von Leibniz – finales s.XVII

La máquina de diferencias de Babbage – s.XIX

- El concepto de programa externo

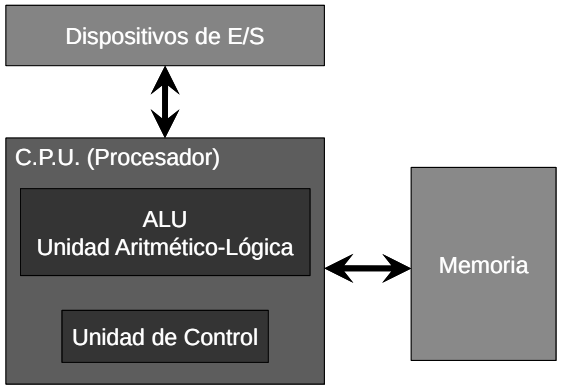
Lady Ada Lovelace es considerada la primera programadora



Comienza la era de la Informática

1945

- El modelo de J. von Neumann: estructura de la computadora
- El concepto de programa interno



Algunos hitos en la vida de la Informática

¿Cuándo se crean los primeros lenguajes de programación? ¿cuáles fueron? ¿Es C++ mayor de edad? ¿Quién es más joven: C++ o Java?

¿Cuál puede considerarse el primer virus informático?

¿Quién surge primero: MS-DOS, Windows, UNIX, Linux, Android?

¿Cuándo se crea la WWW?

¿Qué soportes externos conoces: disquete, CD, DVD? ¿Alguno nació antes que tú?

¿Quién es Alan Turing?

Las computadoras están por todas partes

¡Con múltiples formas distintas de un PC!

¿Qué entiende la computadora?

Computadora: Máquina electrónica digital, dotada de ...

La computadora manipula información digital: esquema binario

¿Por qué no se usa una representación analógica de la información?

¿A qué nos conduce la solución adoptada para evitar el problema tecnológico?

Lenguaje máquina

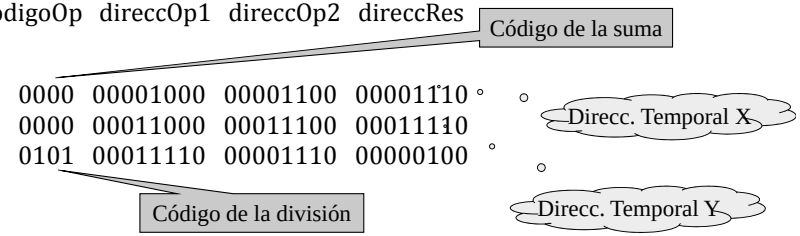
$$a = (b + d)/(c + e)$$

Pasos

- sumar *c* y *e*, y guardar el resultado en una dirección de memoria temporal *X*
- sumar *b* y *d*, y guardar el resultado en una dirección de memoria temporal *Y*
- dividir el contenido de *Y* por el de *X* y guardar en la dirección de *a*

Ejemplo de código máquina

- codigoOp direccOp1 direccOp2 direccRes



Lenguaje máquina

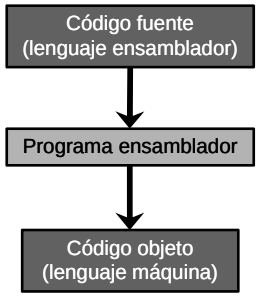
- Lenguaje de programación de bajo nivel
  Nulo nivel de abstracción: los códigos contienen los ceros y unos que gobiernan directamente los circuitos de la CPU
- Totalmente dependiente de la máquina
  Cada familia de procesadores usa sus propios códigos, distintos de los de otras familias
- Programación: muy tediosa
- Grandes posibilidades de error

Lenguaje ensamblador

- Lenguaje simbólico con una mínima capacidad de abstracción
  - Nemotécnicos para los códigos que representan instrucciones
  - Nombres simbólicos para las direcciones de memoria

$$a = (b + d)/(c + e)$$

```
ADD   C, E, X
ADD   B, D, Y
DIV   Y, X, A
```



- Mayor legibilidad
- ¿Cómo entiende la máquina este código?
- Dependiente de la máquina

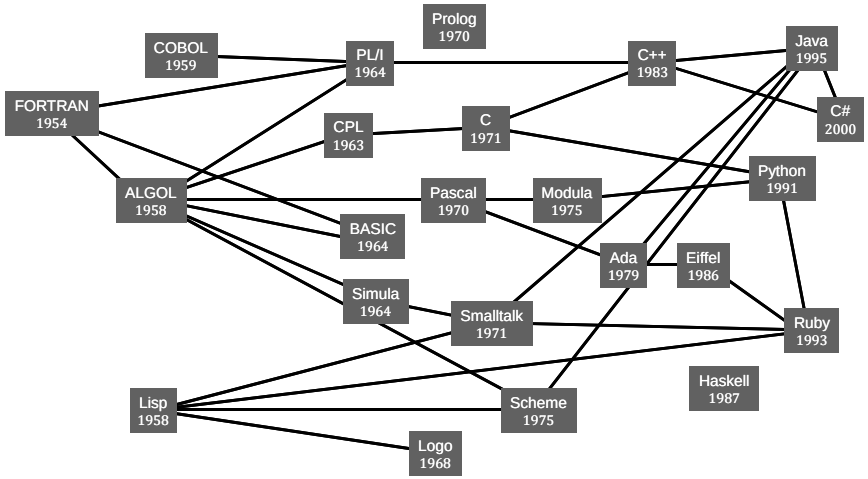
Lenguajes de programación de alto nivel

- Lenguaje que permite expresar el mecanismo de resolución de problemas usando instrucciones independientemente de la computadora
- Más cercanos a los lenguajes natural y matemático

$$a = (b + d)/(c + e);$$

- Capacidad de abstracción
  - Abstracción procedimental
  - Abstracción de datos
- Mayor legibilidad, mayor facilidad de codificación
- ¿Cómo conseguir que la computadora "entienda" los programas escritos en lenguajes de alto nivel?
  - Compiladores e intérpretes

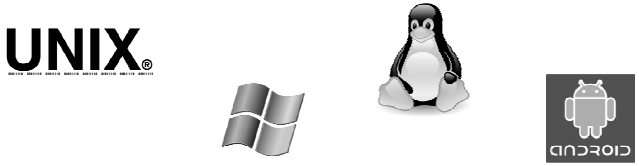
Lenguajes de programación de alto nivel



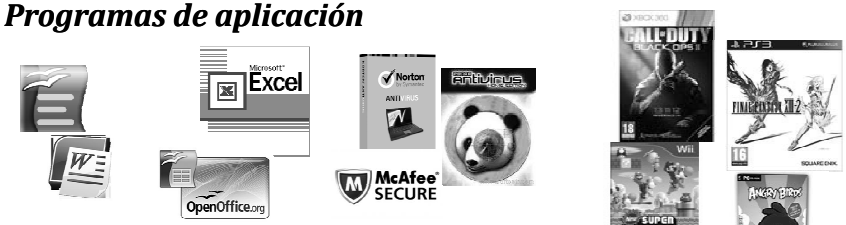
Fuente: http://www.levenez.com/lang/

Programas y más programas

El sistema operativo

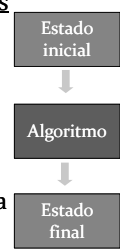


Programas de aplicación



La programación: resolución de problemas

- PROGRAMAR **ES** RESOLVER PROBLEMAS
  - Actividades implicadas en la descripción, el desarrollo y la implementación de soluciones algorítmicas eficaces y eficientes a problemas bien especificados
    - Estado inicial - Entrada del problema - Precondiciones
      Datos iniciales y relaciones entre ellos
    - Estado final - Salida del problema - Postcondiciones
      Datos finales y relaciones entre ellos
    - Algoritmo: sistemática que transforma la entrada en la salida
- PROGRAMAR **NO ES** CONOCER LA SINTAXIS DE MUCHOS LENGUAJES DE PROGRAMACIÓN
  - Los lenguajes de programación son un medio para expresar algoritmos
  - Programa: algoritmo expresado en un cierto lenguaje de programación



La programación: resolución de problemas

El primer problema del montón de fichas

Supongamos que tenemos una secuencia de fichas, cada una de las cuales tiene escrito el nombre de una persona junto con otros datos personales (fecha de nacimiento, dirección, número de teléfono). Las fichas están ordenadas alfabéticamente por el nombre.

Queremos felicitar por teléfono a los que cumplen años hoy.

¿Cómo detectamos a los cumpleañeros?

- ¿Cuál es la entrada?
- ¿Cuál es la salida?
- ¿Cuál es el algoritmo?
- ¿Qué lenguaje utilizamos para describir las respuestas a todo lo anterior?

La programación: resolución de problemas

El segundo problema del montón de fichas

Supongamos que volvemos a tener la misma secuencia de fichas.

Nos pasan el nombre de una persona a la que hay que felicitar por teléfono.

¿Cómo lo hacemos?

- ¿Cuál es la entrada?
- ¿Cuál es la salida?
- ¿Cuál es el algoritmo?
- ¿Qué lenguaje utilizamos para describir las respuestas a todo lo anterior?

La programación: resolución de problemas

El tercer problema del montón de fichas

Supongamos que volvemos a tener la misma secuencia de fichas.

Nos pasan el nombre de una persona y su nuevo número de teléfono.

¿Cómo hacemos el cambio? ¿Y si la persona no está?

- ¿Cuál es la entrada?
- ¿Cuál es la salida?
- ¿Cuál es el algoritmo?
- ¿Qué lenguaje utilizamos para describir las respuestas a todo lo anterior?

La programación: resolución de problemas

El cuarto problema del montón de fichas

De nuevo tenemos la misma secuencia de fichas.

Nos dan una nueva ficha y hay que incorporarla al montón existente sin romper el orden.

¿Cómo averiguamos dónde incorporarla?

- ¿Cuál es la entrada?
- ¿Cuál es la salida?
- ¿Cuál es el algoritmo?
- ¿Qué lenguaje utilizamos para describir las respuestas a todo lo anterior?

La programación: resolución de problemas

El quinto problema del montón de fichas

¡Nos han desordenado la secuencia de fichas!

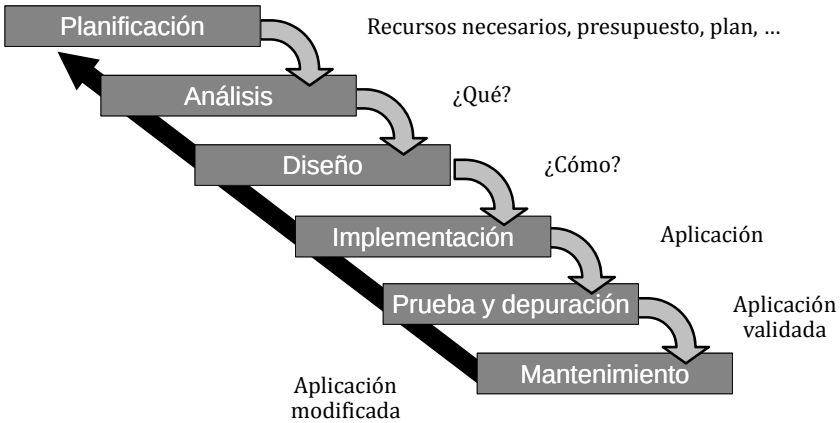
De nuevo nos pasan el nombre de una persona a la que hay que felicitar.

¿Cómo lo hacemos ahora?

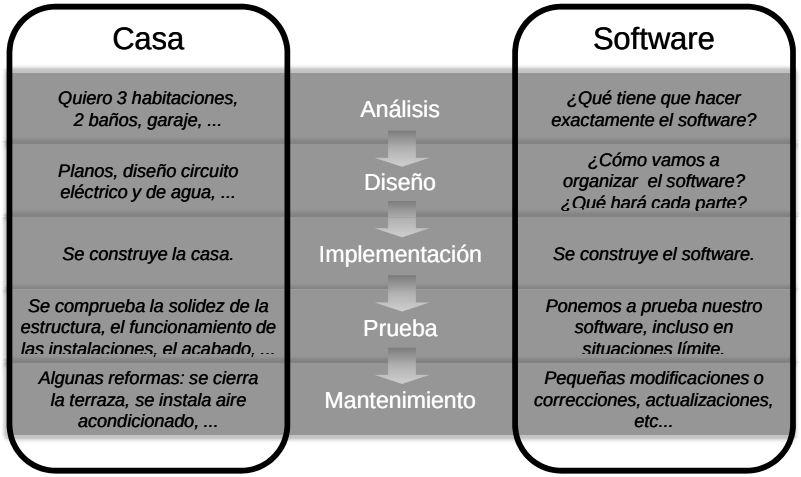
- ¿Cuál es la entrada?
- ¿Cuál es la salida?
- ¿Cuál es el algoritmo?
- ¿Qué lenguaje utilizamos para describir las respuestas a todo lo anterior?

La ingeniería del software

El modelo de desarrollo “en cascada”



La ingeniería del software



Los aspectos de los lenguajes de programación

Los lenguajes de programación pueden describirse:

- A nivel sintáctico
  - Descripción de cómo se pueden construir y secuenciar los elementos del lenguaje
  - Es una descripción hecha usando algún formalismo artificial
- A nivel semántico
  - Descripción del significado de cada elemento del lenguaje
  - Puede usarse lenguaje natural o expresarse de manera formal
- A nivel pragmático
  - Descripción de cómo se utiliza el lenguaje de forma práctica
  - Lo típico: usar tutoriales y ejemplos de programas

Sintaxis de los lenguajes de programación

Sintaxis = reglas que especifican y permiten verificar la corrección de las sentencias de un lenguaje

Formalismos:

- BNF (*Backus-Naur Form*)
- EBNF (*Extended Backus-Naur Form*)
- Diagramas sintácticos

Sintaxis de los lenguajes de programación: BNF

- TERMINAL — Símbolo del lenguaje que se está definiendo
- <no terminal> — Símbolo que se define en términos de otros (terminales y no terminales)
- Regla de producción — Descripción de símbolos no terminales.
  Parte izquierda: símbolo no terminal
  Parte derecha: combinación de terminales y no terminales; vacío
- Metasímbolos — Símbolos del formalismo
  - ::= Equivalencia
  - | Alternativa

Sintaxis de los lenguajes de programación: BNF

- Reglas BNF para expresar la sintaxis de un número entero positivo en un cierto lenguaje de programación

<numero entero> ::= <signo opcional> <secuencia digitos>
<signo opcional> ::= + | <nada>
<secuencia digitos> ::= <digito> | <digito> <secuencia digitos>
<digito> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<nada> ::=

```
+23
1374
1+34
3.4
002
```

- Reglas BNF para expresar la sintaxis de un número entero

Sintaxis de los lenguajes de programación: EBNF

BNF
- TERMINAL
- <no terminal>
- Metasímbolos
  ::= Equivalencia
  | Alternativa

- Recursividad

EBNF
- "terminal"
- No-terminal
- Metasímbolos
  ::= Equivalencia
  | Alternativa
  {...} Aparición 0, 1, 2,... veces
  [...] Opcionalidad
  (...) Agrupaciones

- Recursividad no permitida
- Coincidencia de metasímbolo con símbolo del lenguaje: el metasímbolo entre comillas simples

Sintaxis de los lenguajes de programación: EBNF

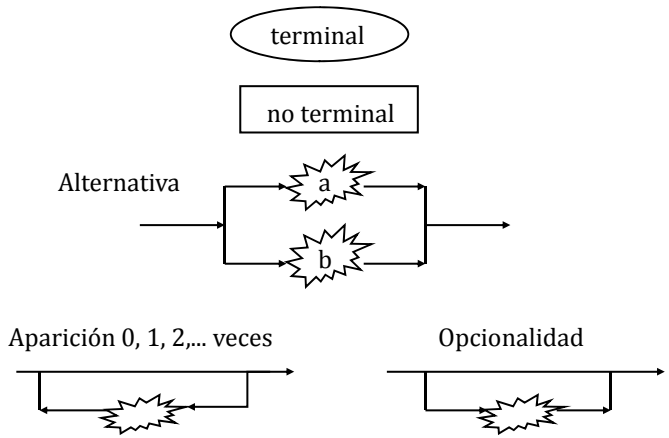
- Reglas EBNF para expresar la sintaxis de un número entero positivo en un cierto lenguaje de programación

Numero-entero ::= [Signo] Secuencia-digitos
Signo ::= "+"
Secuencia-digitos ::= Digito {Digito}
Digito ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

```
+23
1374
1+34
3.4
002
```

- Reglas EBNF para expresar la sintaxis de un número entero

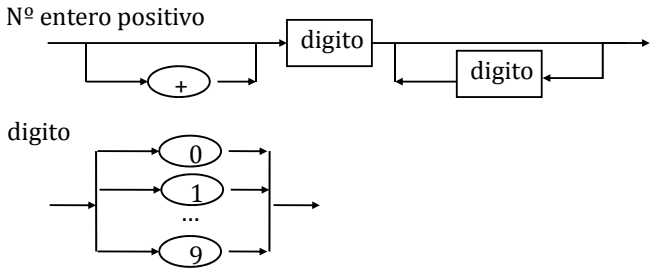
Sintaxis de los lenguajes de programación: diagramas sintácticos



Donde va esto puede ir un terminal, un no terminal, una secuencia de terminales y no terminales,...

Sintaxis de los lenguajes de programación: diagramas sintácticos

- Diagramas sintácticos para expresar la sintaxis de un número entero positivo en un cierto lenguaje de programación




```
+23
1374
1+34
3.4
002
```

- Diagramas sintácticos para expresar la sintaxis de un número entero

Acerca de *Creative Commons*

Licencia CC (Creative Commons)

Este tipo de licencias ofrecen algunos derechos a terceras personas bajo ciertas condiciones.

Este documento tiene establecidas las siguientes:

Reconocimiento (*Attribution*):
En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.

No comercial (*Non commercial*):
La explotación de la obra queda limitada a usos no comerciales.

Compartir igual (*Share alike*):
La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

En http://es.creativecommons.org/ y
http://creativecommons.org/puedes saber más de Creative Commons.