Compiladores e intérpretes
Introducción

Primer semestre 2008

Creación colectiva: Mario Agüero, Élida Herrera, Antonio Luna, José A. Sánchez, Ignacio Trejos

¿Para qué lenguajes de programación?

- ¿Qué es un lenguaje de programación?
- ¿Cuáles tipos de lenguajes de programación existen?
- ¿Cómo se implementan los lenguajes de programación?
- ¿Por qué hay tantos lenguajes de programación?
- ¿Necesita el mundo nuevos lenguajes?

Bueno …

"Some believe that we lacked the programming language to describe your perfect world"

Agent Smith - **The Matrix**

InfoWorld

Bill Gates casts Visual Studio .Net

By Matt Berger
February 13, 2002 11:56 am PT

SAN FRANCISCO -- Microsoft's Bill Gates cast his company's .Net initiative wide Wednesday, releasing the final version of the long-anticipated developer toolkit, Visual Studio .Net, as well as the underpinnings of its emerging Web-based development platform, called the .Net Framework.

"When we started out we said this could be one of the biggest pieces of work we have to do on a tool," Gates said of Microsoft's efforts to remodel its development tools already used by millions of Visual Basic and C++ developers to add new support for building Web-based applications.

Straying from its typical two-year release cycle, the latest incarnation of Microsoft's application development environment has been in the making for more than three years. New features will allow developers to write applications using more than 20 different programming languages that can run on computers ranging from cell phones to servers and interact with applications written for virtually any computing platform, according to Microsoft.

EETIMES

Sun invites IBM, Cray to collaborate on high-end computer language

By Rick Merritt, EE Times
December 16, 2003 (8:14 p.m. EST)
URL: http://www.eetimes.com/story/OEG20031216S0031

MOUNTAIN VIEW, Calif. — Sun Microsystems is inviting competitors IBM Corp. and Cray Inc. to collaborate on defining a new computer language it claims could bolster performance and productivity for scientific and technical computing. The effort is part of a government-sponsored program under which the three companies are competing to design a petascale-class computer by 2010.

Desarrollos lingüísticos recientes

- Java 6
- Groovy
- C#
- Ruby
- Python
- Mozart
- Aspect Oriented Programming
  - AspectJ, Aspect.Net
- BPM (Business Process Management)
  - BPEL-J, PLEW4WS

¿De qué se trata este curso?

- Procesamiento (implementación) de lenguajes de programación
  - Léxico, sintaxis y contexto de los lenguajes
  - Estructuras para la representación y la ejecución de programas
  - Interpretación
  - Compilación
- Formalismos para descripción de características de los lenguajes
- Principios, técnicas y herramientas
- Comprender y extender programas complejos
- Reflexión sobre diseño (de lenguajes, de programas)

Descripción sintética

- Estudia principios y técnicas necesarios para la construcción de procesadores de lenguajes de programación, con énfasis en compiladores e intérpretes. El aprendizaje permite aplicar modelos abstractos (lenguajes formales y autómatas), enfrentar problemas de manipulación de información simbólica, realizar programación modular avanzada, analizar sistemas complejos de software y hacer modificaciones sistemáticas, y ampliar el repertorio de métodos para resolución de problemas informáticos.

Objetivo general

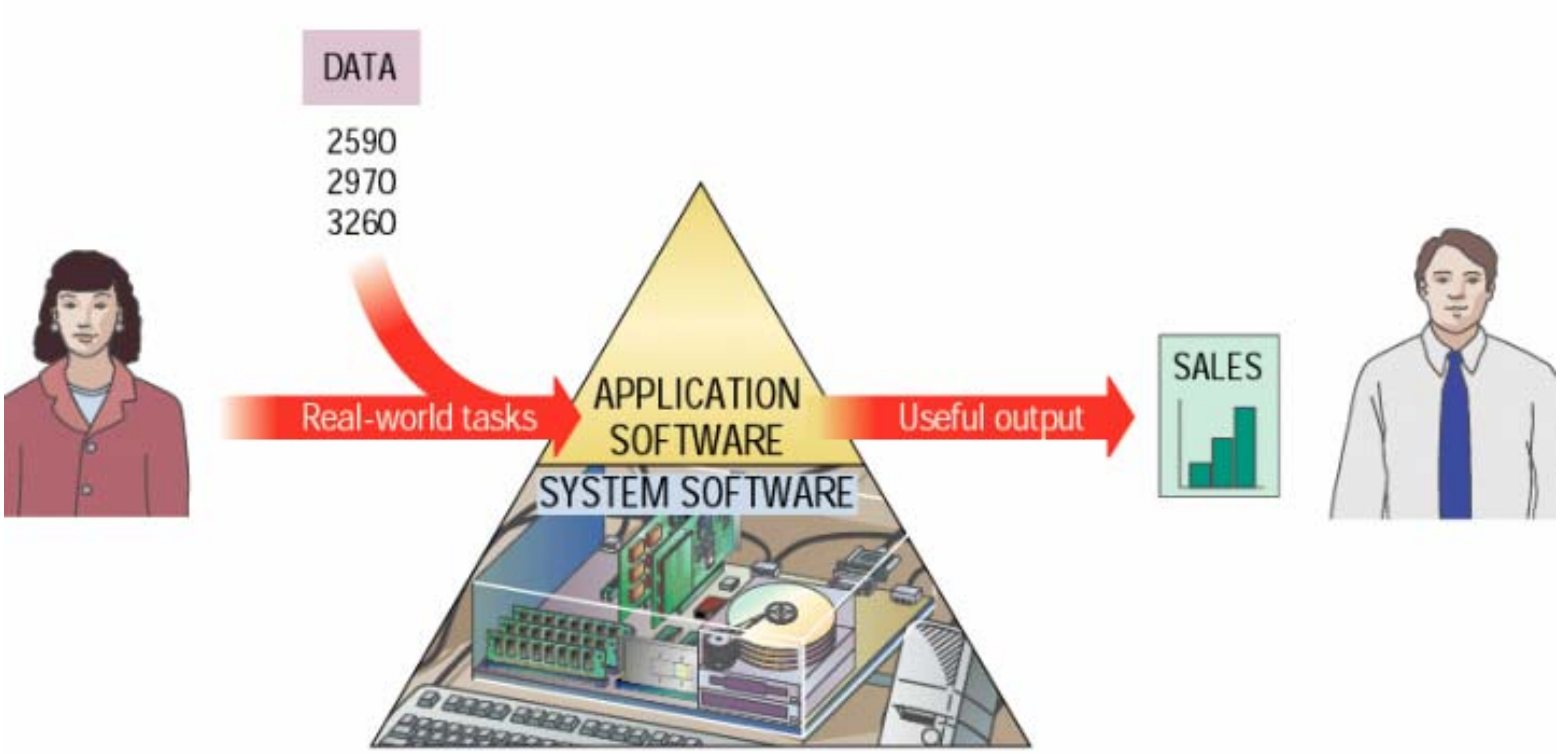
- Aplicar principios, modelos y técnicas para el diseño y la construcción de procesadores de lenguajes de programación de alto nivel, con énfasis en compiladores e intérpretes.

Objetivos específicos

- Identificar los principales problemas relacionados con la implementación de lenguajes de programación.
- Comprender principios y métodos para implementar lenguajes de programación.
- Conocer las tareas que deben realizarse para traducir un lenguaje de programación a un lenguaje de máquina.
- Comprender modelos abstractos de lenguajes y máquinas formales (gramáticas, expresiones regulares y autómatas) y su aplicabilidad para la implementación de lenguajes de programación.
- Entender técnicas para construir compiladores e intérpretes que procesen lenguajes procedimentales.
- Diseñar y construir, de manera sistemática, un compilador para un lenguaje imperativo con estructura de bloques.
- Diseñar y construir implementaciones prototipo de lenguajes de programación vía intérpretes.
- Comprender literatura técnica en idioma Inglés.

Clasificación general del software

- El software que corre sobre una computadora se puede clasificar como:



Clasificación general del software

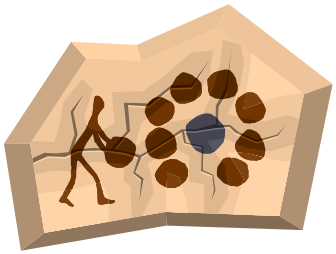
- El software de aplicación está formado por las aplicaciones de propósito específico que ayudan a los usuarios de computadoras a resolver problemas

Clasificación general del software

- El software del sistema:
  - conjunto de programas que controlan la operación de la computadora,
  - ofrecen un entorno más cómodo para el desarrollo y ejecución de programas,
  - constituyen una **abstracción** (**máquina abstracta**) sobre las peculiaridades de las computadoras reales.
- Independientes de cualquier área específica de aplicación.
- Ejemplos:
  - el sistema operativo,
  - compiladores, enlazadores, editores,
  - manejadores de bases de datos.

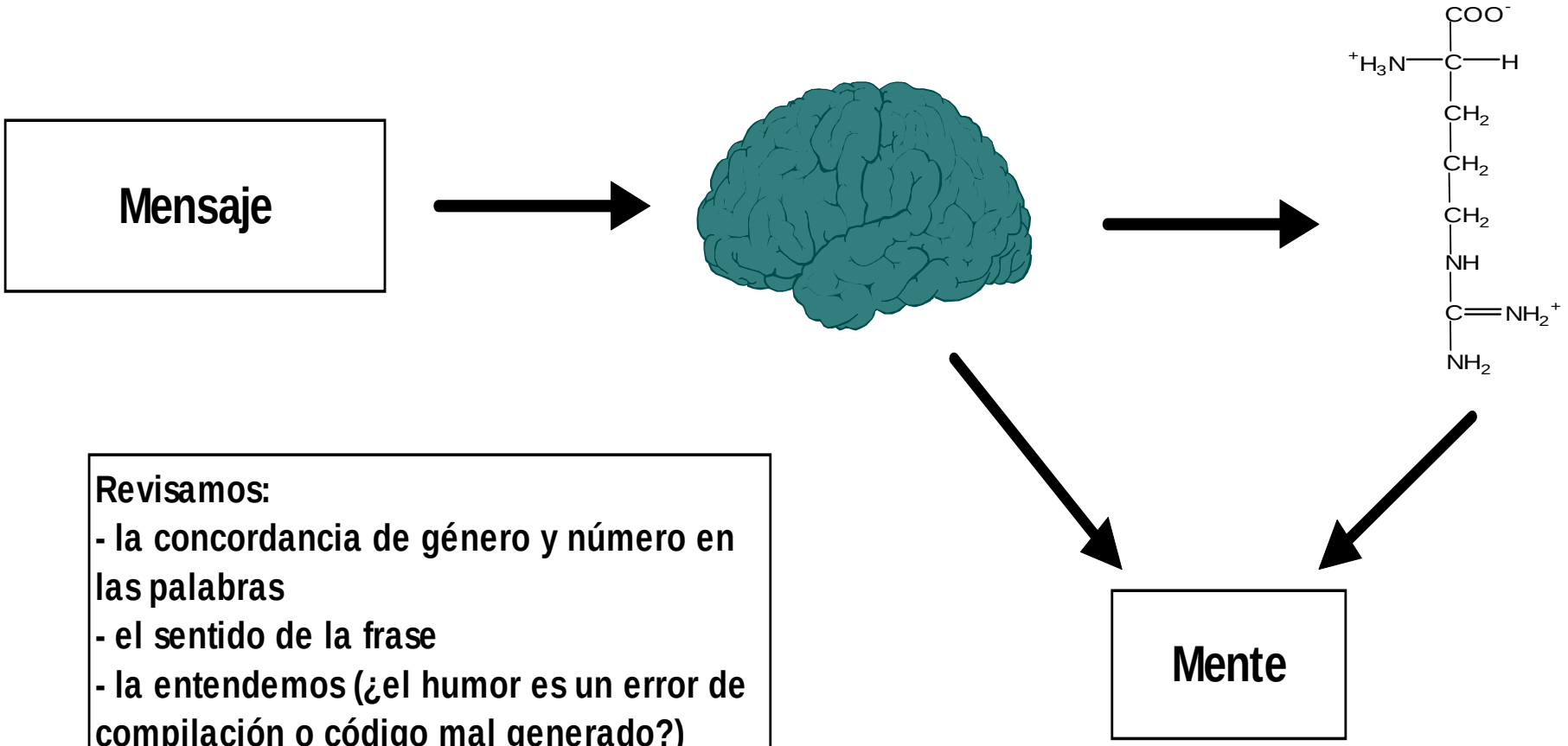
¿Por qué?

- Inconscientemente, nosotros procesamos constantemente lenguajes



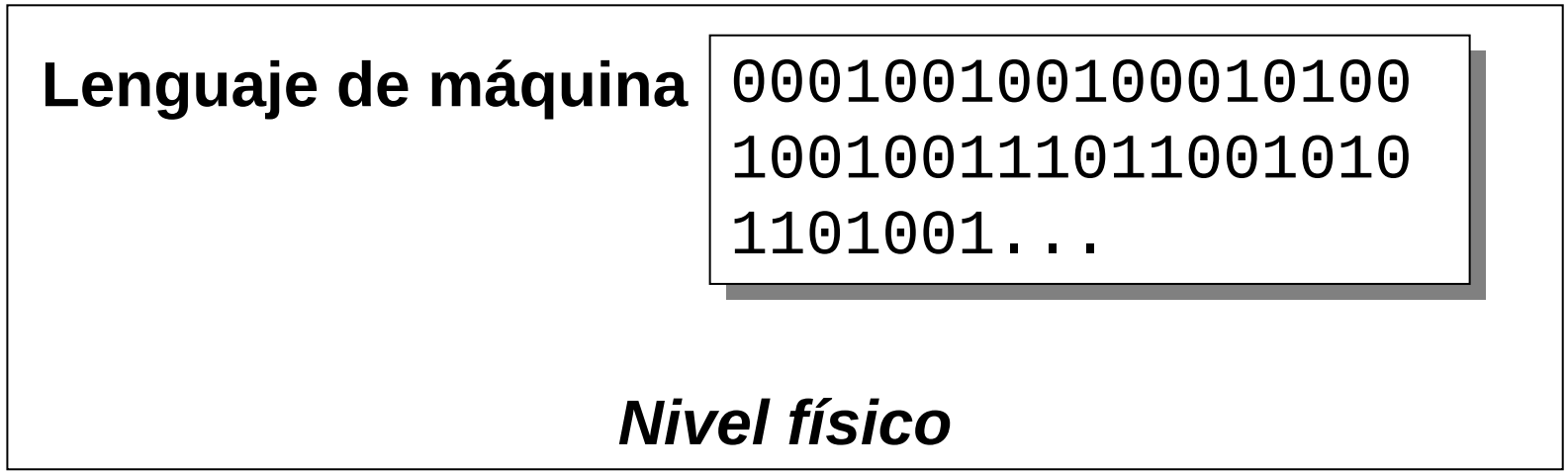
¿Por qué?

- Se da un procesamiento de los mensajes recibidos en nuestro cerebro para ser entendidos:



¿Por qué?

- Así como el procesamiento cerebral se da a nivel electro-químico, el procesamiento de datos en una computadora se da en 0's y 1's (lenguaje de máquina)



¿Por qué?

- Se han implementado niveles superiores de abstracción y lenguajes u operaciones en cada uno para simplificar la ejecución de código.

Lenguajes de alto nivel (Java, C++, VB, DELPHI, PASCAL,…)

```
for i := 1 to 10 do
  graficar (imagen[i])
end
```

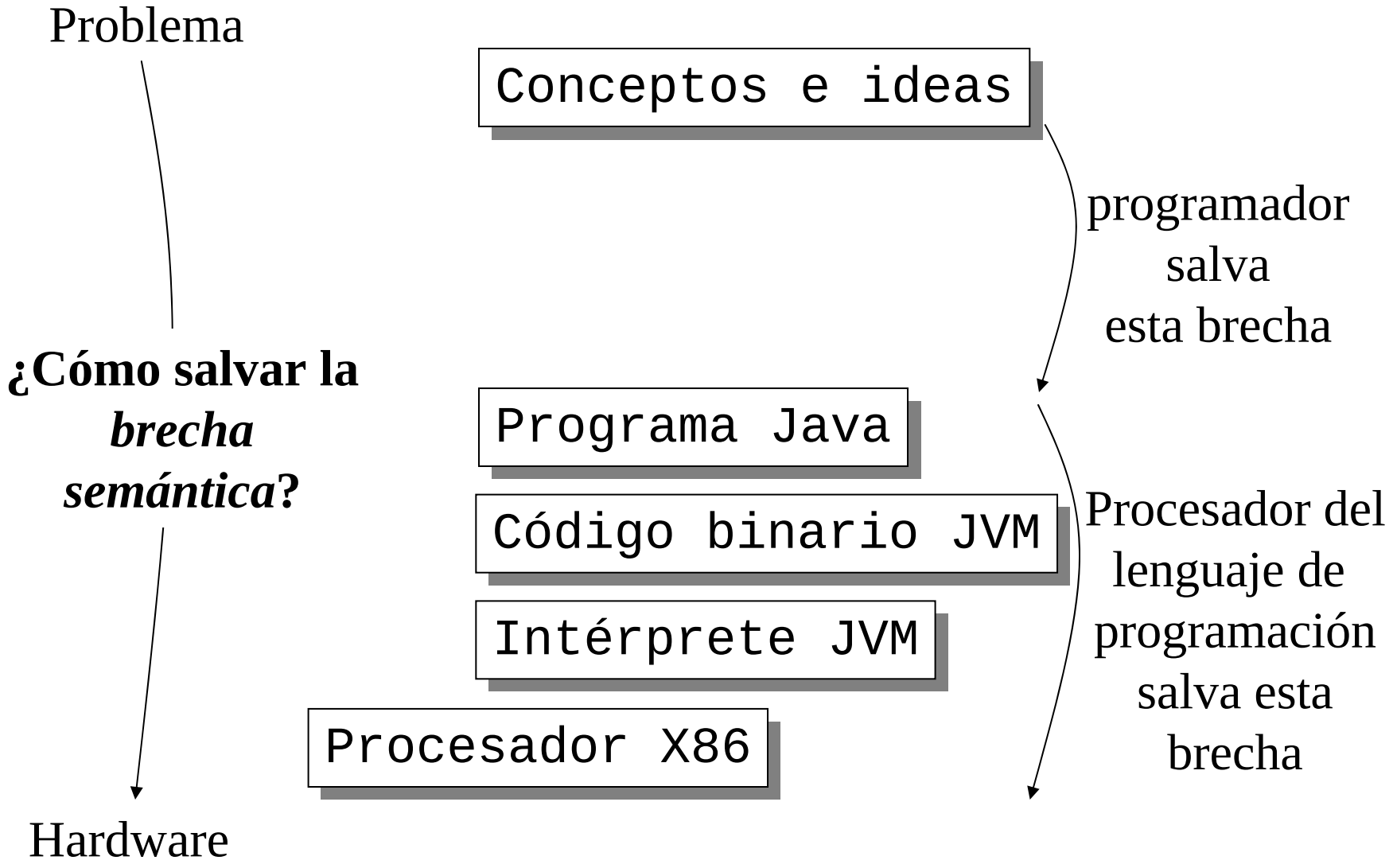
Lenguaje ensamblador (bajo nivel)

```
LOAD r1,b
LOAD r2,h
RET
```

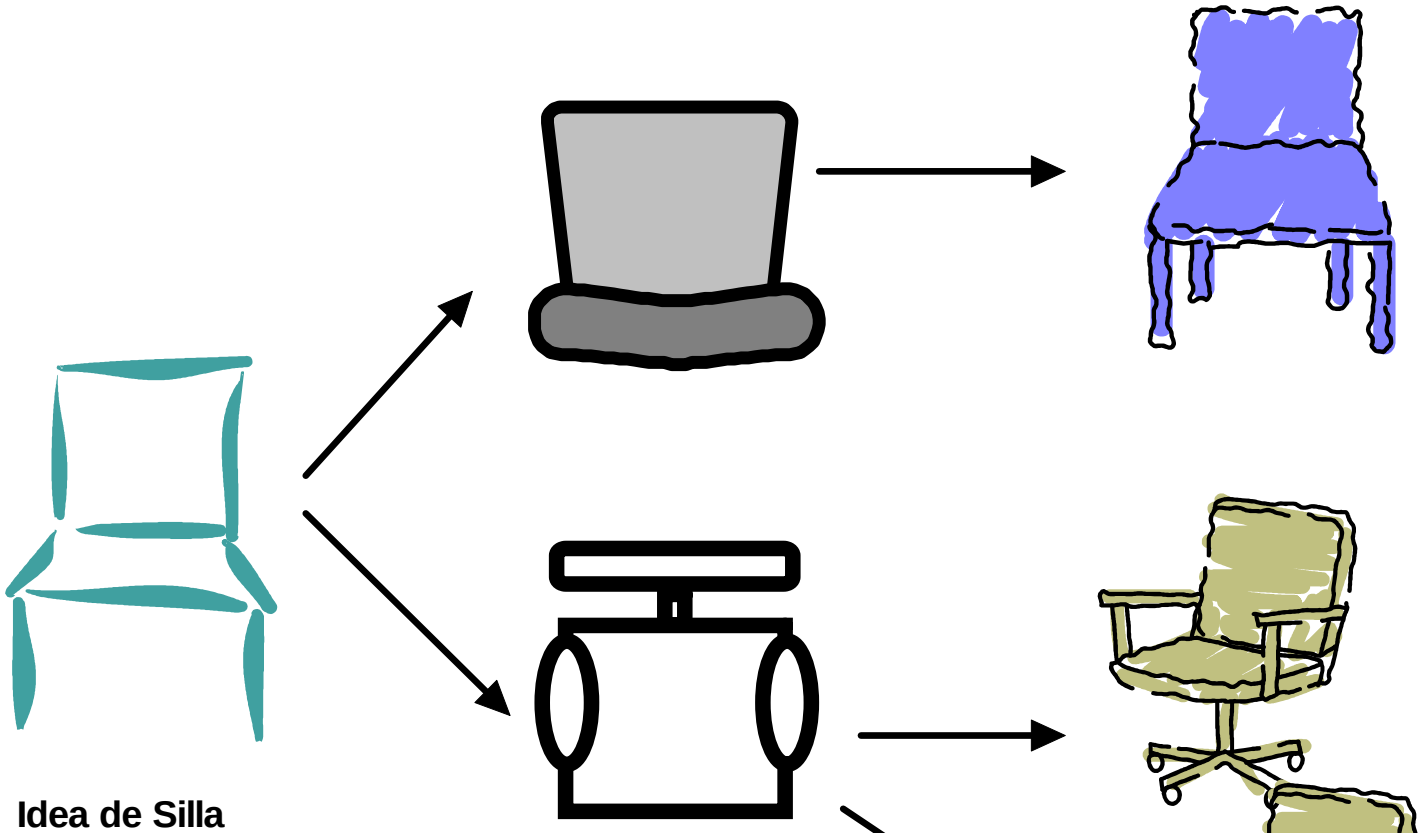
Lenguaje máquina

```
00010010010001010 0
1101001...
```

¿Para qué los lenguajes de programación?



El proceso de abstracción



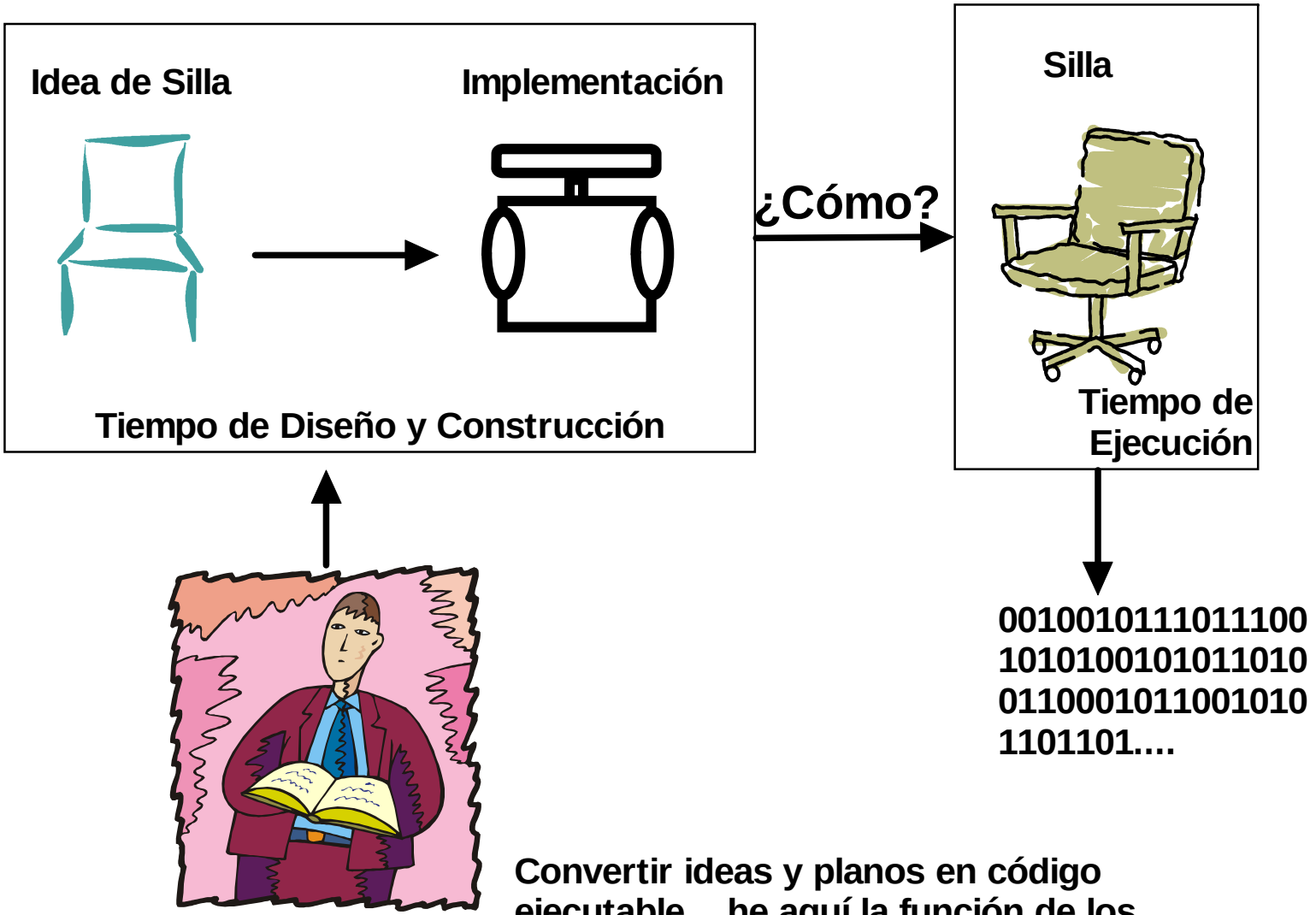
Idea de Silla

Plano de silla

¡Sillas!

De un mismo plano puede haber varias sillas. De una misma idea, puede haber varios planos.

El problema…



Convertir ideas y planos en código ejecutable... he aquí la función de los compiladores

Definiciones

- Una *máquina* es un dispositivo (real o virtual) que realiza una tarea computacional
- Un *programa* es un texto que describe (y prescribe) la conducta de una máquina
- Un *lenguaje de programación* es un lenguaje útil para escribir programas; un sistema de notación que permite describir cómputos

Definiciones

- Un *intérprete* es una máquina que acepta cualquier programa y se comporta según la especificación de la máquina descrita por el programa
- Una computadora es un intérprete que está implementado en hardware
- Un *compilador* es una máquina que traduce programas de un lenguaje (*lenguaje fuente*) de programación a otro (*lenguaje objeto*).

¿Qué necesita un compilador?

- Tres cosas le son imprescindibles:
  - Lenguaje **fuente** (origen), en que se escriben los programas que traduce
  - Lenguaje **objeto** (destino): en que la salida del compilador estará escrita
  - Lenguaje de **implementación**: lenguaje en que el propio compilador está escrito.

Evolución

- 1G: Lenguajes máquina
  - Binario, instrucciones máquina
- 2G: Lenguajes simbólicos
  - Ensamblador, mnemónico
- 3G: Orientados a aplicaciones específicas
  - Problemas de negocios, científicos
  - Procedimentales, especificación de procedimientos
  - Inicio de la OO

```
0000 0001 0110 1110
0100 0000 0001 0010
```

```
LOAD x
ADD R1 R2
```

```
public Token scan ( ) {
while (currentchar == ‘ ’
          || currentchar == ‘\n’)
  {….}
 }
```

Evolución

- 4G: Propósito general
  - Facilitan la programación, depuración, mantenimiento
  - Aparecen los IDEs
  - Fortalecimiento de la OO
  - Uso de lenguajes declarativos (SQL, etc.)

```
select fname, lname
from employee
where department='Sales'
```

Evolución

- 5G y más allá: Conocimiento:
  - Programas auto-optimizables
  - Inteligencia Artificial
  - ¿Componentes inteligentes?
  - Agentes

```
fun parse p = junk bind (fn _ =>
              p    bind (fn v =>
              result v));
```

```
uncle(X,Y) :- parent(Z,Y), brother(X,Z).
```

Lenguaje de máquina

- Único lenguaje que comprende el computador.
- Se expresaban en largas cadenas de 1s y 0s, o sus taquigrafías: octal, hexadecimal
- Se intentan hacer las instrucciones lo más simples posible.
  - Grupo para la transferencia de datos
  - Grupo aritmético
  - Grupo lógico y de desplazamientos
  - Grupo de manejo de cadenas
  - Grupo de flujo de control
  - Grupo de control del sistema

Ejecución de un programa

Una computadora solamente interpreta las instrucciones que se encuentran en el lenguaje que ella entiende: el lenguaje de máquina.

Máquina multinivel

- Los lenguajes de máquina son demasiado elementales, es difícil y tedioso utilizarlos.
- El lenguaje ensamblador es una variante del lenguaje de máquina.
- En ensamblador, se manejan identificadores en lugar de códigos reales para las operaciones, los valores y las localidades de almacenamiento de la máquina.

Máquina multinivel

- Ni el programa en lenguaje de máquina:

```
00000010101111001010
00000010111111001000
00000011001110101000
```

ni el fragmento (en algún ensamblador):

```
LOAD    I
ADD     J
STORE   K
```

son tan claros como (FORTRAN o C):

```
K = I + J
```

Máquina multinivel

- ¿Cómo expresar de modo simple para el ser humano y comprensible para la máquina?
  - Se debe de crear un conjunto de instrucciones
- Este nuevo conjunto de instrucciones forman un nuevo lenguaje denominado L2, de manera semejante al que forman las instrucciones propias de la máquina, que llamaremos L1.
- Las dos aproximaciones solo difieren en el modo en que los programas escritos en L2 son ejecutados por la computadora, ya que después de todo, sólo puede ejecutar programas escritos en L1.

Máquina multinivel

- Primer método de ejecución:
  - Para ejecutar un programa escrito en L2, se sustituye primero cada instrucción por una secuencia equivalente de instrucciones L1. El resultado es un nuevo programa totalmente escrito en instrucciones en L1.
  - La computadora ejecutará el **nuevo** programa en L1 y no el anterior en L2. Esta técnica se denomina traducción o compilación. Al traductor se le denomina compilador.

Máquina multinivel

- Segundo método de ejecución:
  - Un programa en L1 toma programas escritos en L2 como datos de entrada, examina una instrucción a la vez y ejecuta directamente la secuencia equivalente de instrucciones en L1.

    Esta técnica se denomina interpretación y el programa que la lleva a cabo, intérprete.

Máquina multinivel