

Fundamentos de programación

Clase 1

Agenda

- ¿Qué es la ingeniería de Sistemas?
- ¿Qué es un computador?
- ¿Qué es un lenguaje de programación?
- ¿Cómo se hace un programa?
- ¿Orientación a Objetos?
- UML: Unified Modeling Language
- Elementos de un programa
- Práctica

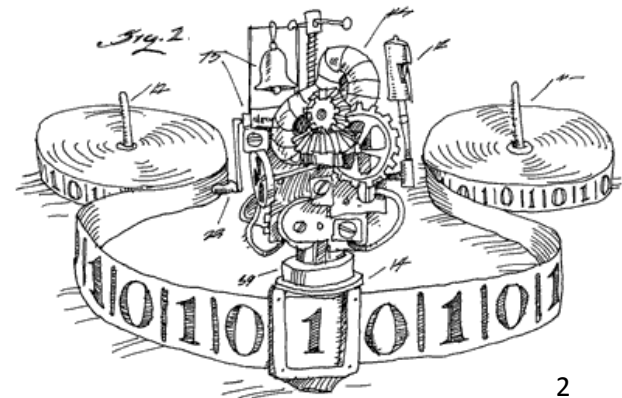
¿Qué es la Ingeniería de Sistemas?

- Ingeniero -> Engineer (hombre de la máquina). *Ingenium* (lat. producir, engendrar")
- Sistema -> *systema* (lat. unión de cosas de una manera organizada)
- Un Ingeniero de Sistemas:
 - Soluciona problemas
 - Convierte una idea en realidad
 - Aplica el ingenio y la inventiva para desarrollar una actividad
 - Transforma la realidad para resolver un problema
 - Una de sus herramientas es el computador

¿Qué es un computador?

computare lat. Calcular

- Es una máquina para calcular
- Utiliza **algoritmos** para transformar **datos** en **información**
- Se compone de **hardware** y **software**
- Se basa en el concepto de “Máquina de Turing”

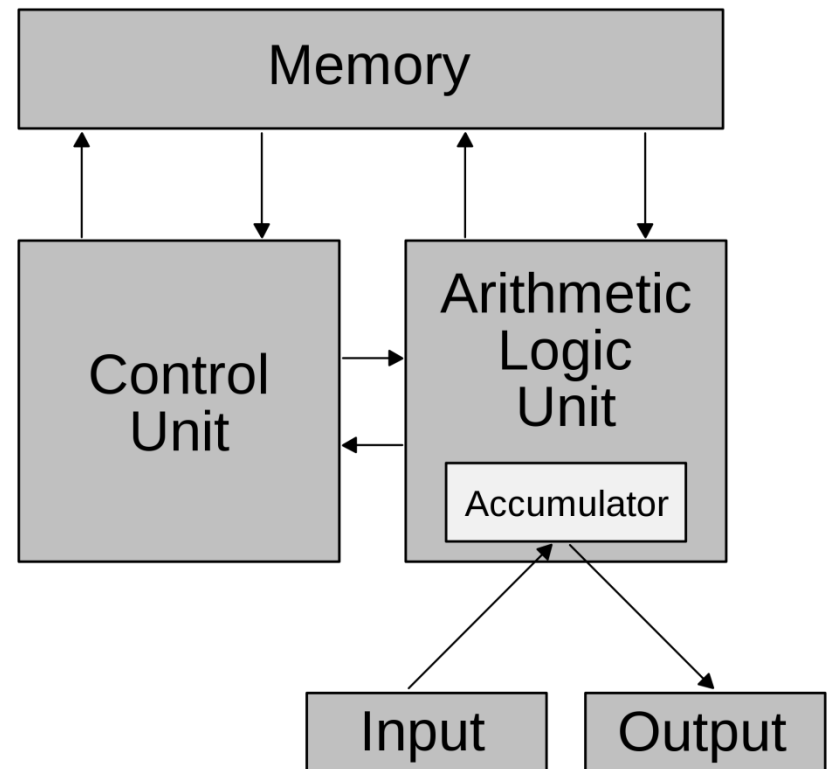


1. http://imageenvision.com/sm/0025-0802-2114-3658_clip_art_graphic_of_a_calculator_cartoon_character_flexing_his_arm_muscles.jpg

2. <http://www.worldofcomputing.net/wp-content/uploads/2013/01/turingMachine.gif>

¿Qué es un computador?

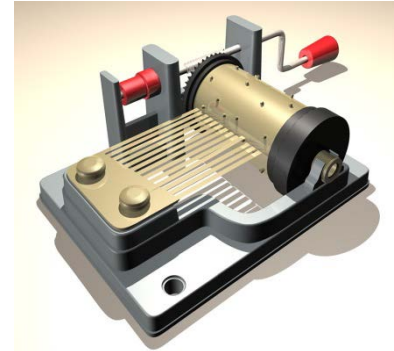
- Los computadores modernos funcionan con **transistores**
- Todos los computadores modernos utilizan la **arquitectura de Von Neumann**
- Se controlan a través de un **lenguaje de programación**



https://upload.wikimedia.org/wikipedia/commons/thumb/8/84/Von_Neumann_architecture.svg/2000px-Von_Neumann_architecture.svg.png

¿Qué es un lenguaje de programación?

- **Programa:** serie de pasos para dirigir una máquina
- **Lenguaje:** Sintaxis + semántica
- **Compilar:** Traducir un programa en lenguaje de máquina



<http://www.bruzed.com/wp-content/uploads/2008/12/musicbox0.jpg>

<http://imgs.xkcd.com/comics/compiling.png>

[illegible]

The Evolution Of Computer Programming Languages



Hex



Assembler



C



Fortran



C++

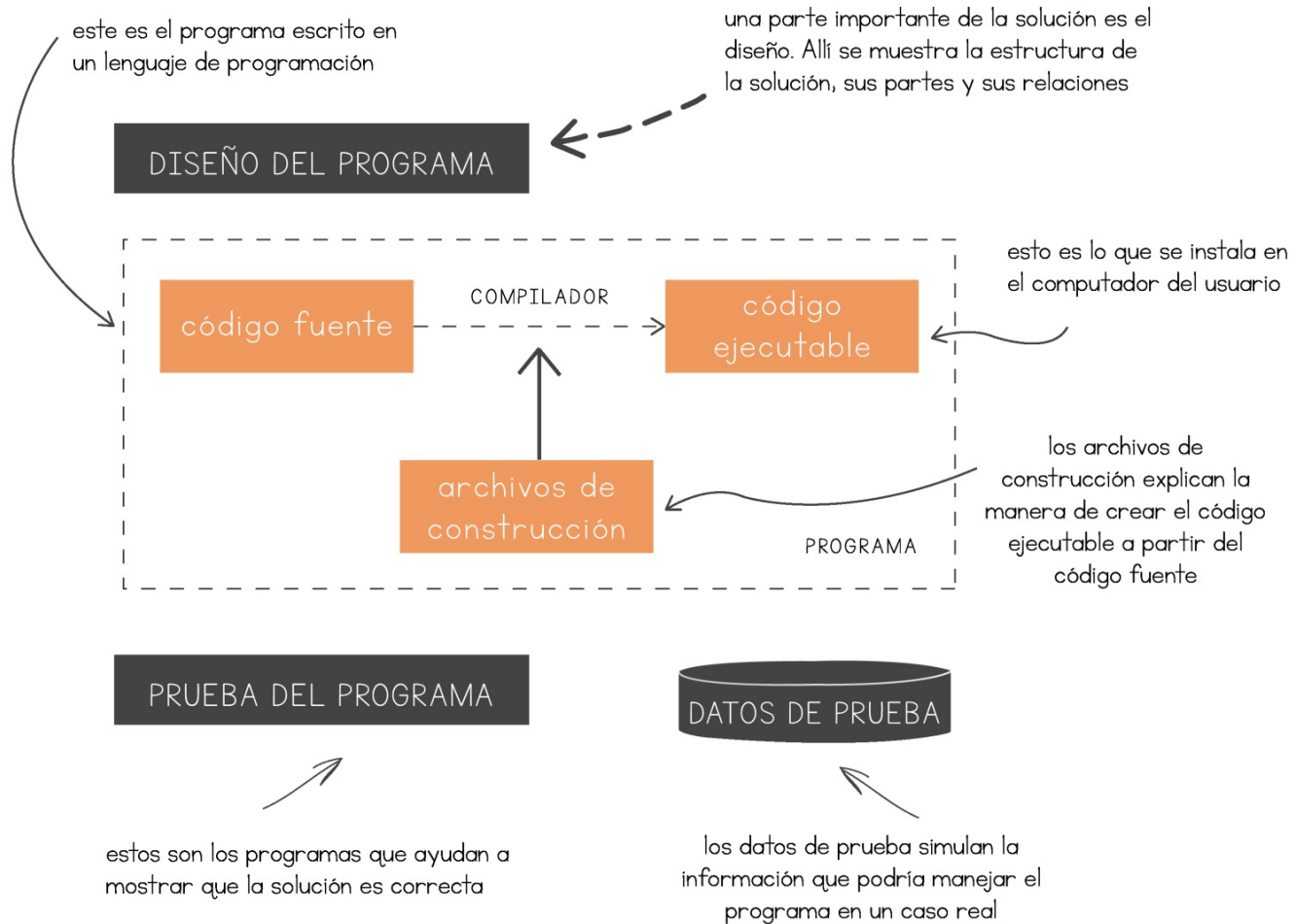


Java



Ruby

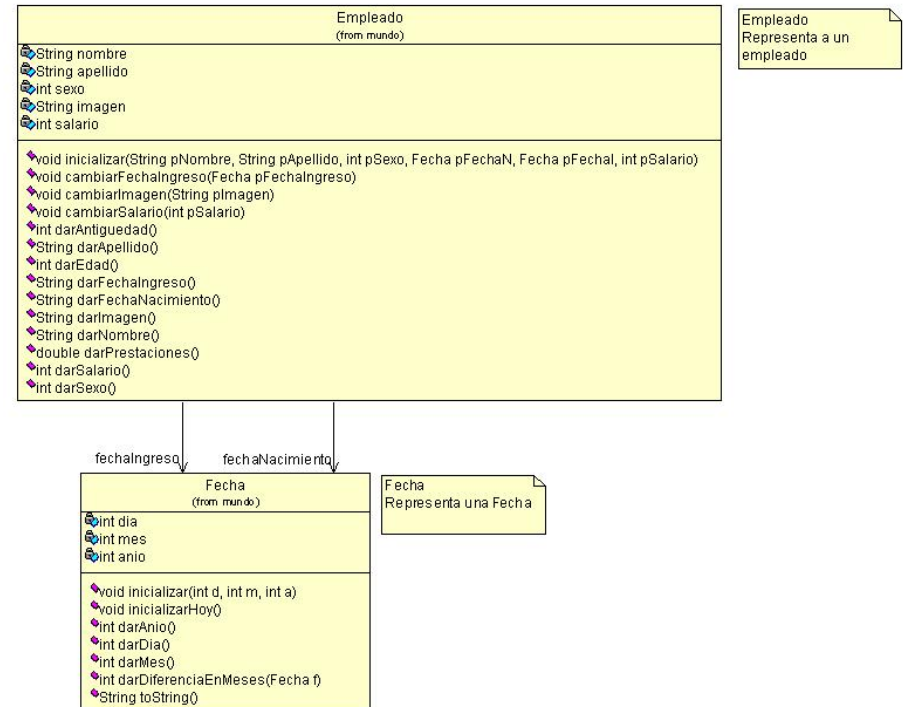
¿Cómo se hace un programa?



https://universidad-de-los-andes.gitbooks.io/fundamentos-de-programacion/content/Nivel1/3_ProblemasYSoluciones.html

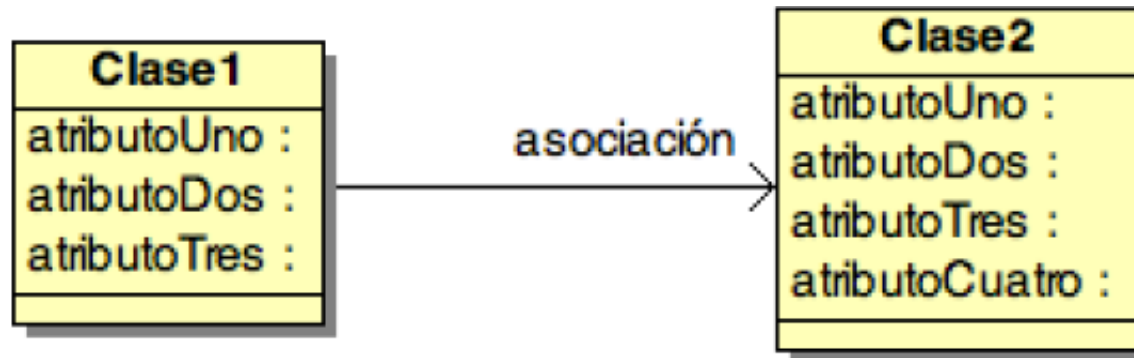
¿Orientación a objetos?

- **Lenguajes procedimentales**
 - Descripción del paso a paso de un programa
 - Programación estructurada
- **Programación orientada a objetos**
 - Los datos y métodos se encuentran **encapsulados** en una unidad llamada **objeto**
 - Un **objeto** es una instancia de una **clase**



UML: Unified Modeling Language

- Lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software
 - Diagrama de clases
 - Diagrama de casos de uso
 - ...



Elementos de un Programa

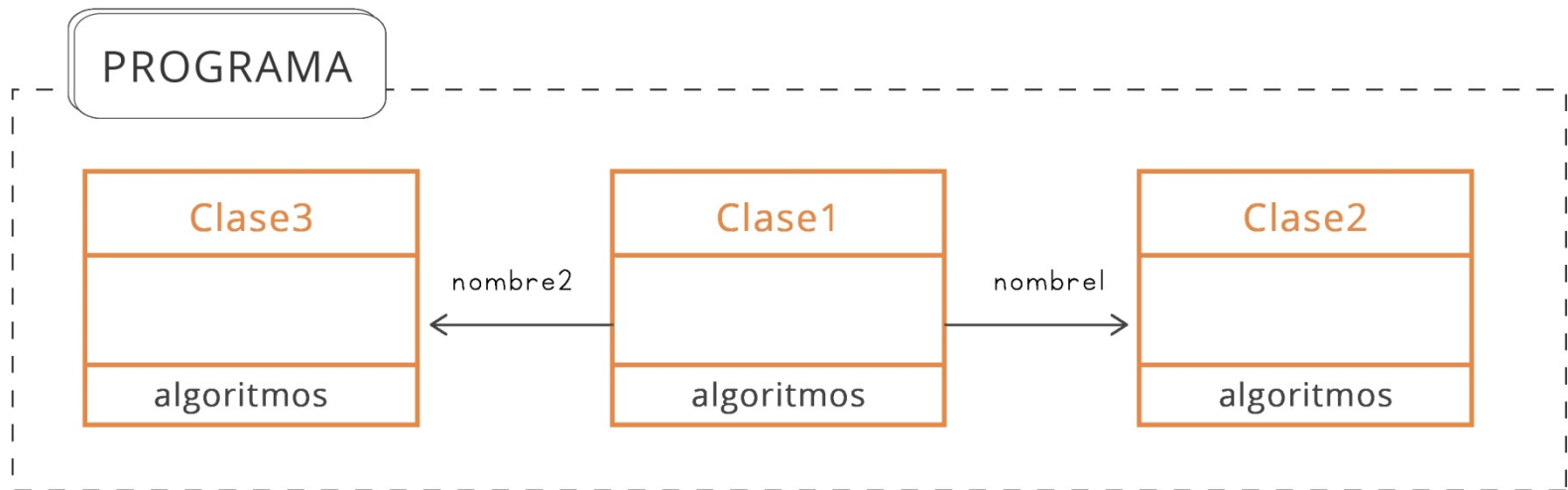
- Algoritmos e instrucciones
 - Un **algoritmo** se puede ver como la solución de un problema muy preciso y pequeño a través de una serie de instrucciones (secuencia ordenada de pasos)
 - Un computador **ejecuta** un algoritmo a través de un lenguaje de programación
 - Un **programa** es un conjunto de algoritmos

Elementos de un Programa

- Clases y objetos
 - Un **objeto** es una **instancia** de una **clase**.
 - Una **clase** define un tipo de elementos en el mundo, su estructura.
 - Un **objeto** representa un elemento individual
 - El **estado de un objeto** es el conjunto de valores específicos que tienen sus atributos en un estado de tiempo.

Elementos de un Programa

- Un programa es un conjunto de clases asociadas entre sí para resolver un problema

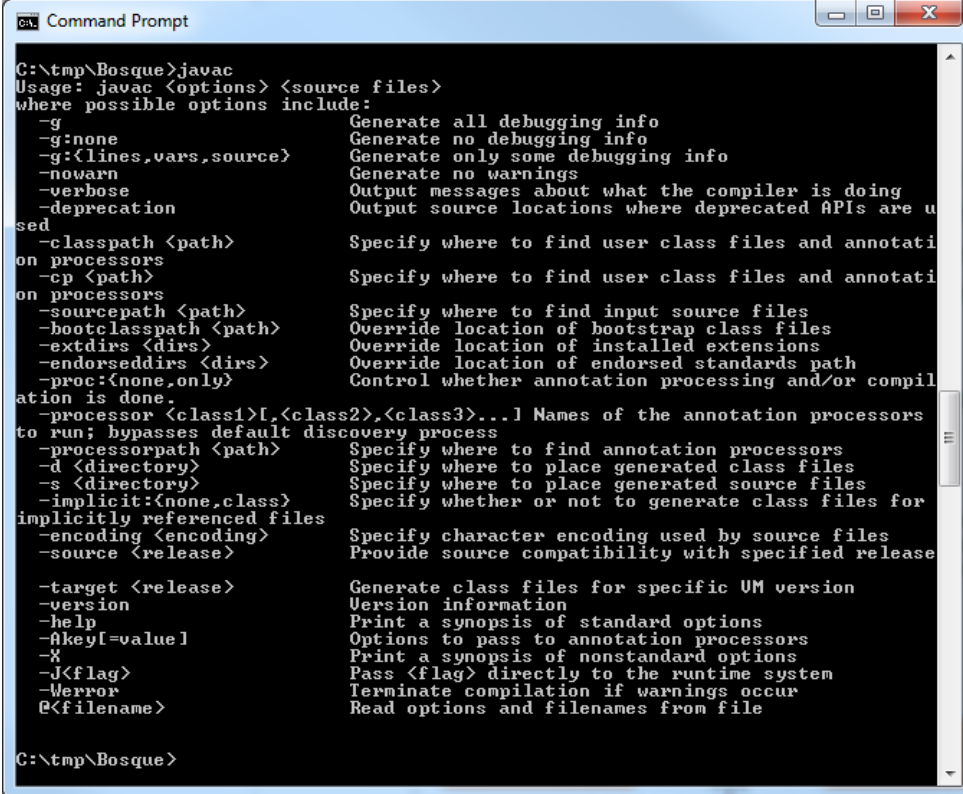


Práctica

1. Instalar Java (en sus computadores personales)
 - a) Bajar el JDK de java
 - b) Seguir las instrucciones del instalador
2. Abrir la terminal de comandos
3. Comprobar si JAVA_HOME está en el PATH
4. Crear el archivo Programa_001.java
5. Compilar y ejecutar

JAVA_HOME

- JAVA_HOME es una **variable de entorno** que le dice al computador donde **buscar Java**
- **javac**: instrucción de java para **compilar** un programa



```
C:\tmp\Bosque>javac
Usage: javac <options> <source files>
where possible options include:
  -g               Generate all debugging info
  -g:none          Generate no debugging info
  -g:<lines,vars,source> Generate only some debugging info
  -nowarn          Generate no warnings
  -verbose         Output messages about what the compiler is doing
  -deprecation     Output source locations where deprecated APIs are used
  -classpath <path> Specify where to find user class files and annotations
  -cp <path>        Specify where to find user class files and annotations
  -sourcepath <path> Specify where to find input source files
  -bootclasspath <path> Override location of bootstrap class files
  -extdirs <dirs>    Override location of installed extensions
  -endorseddirs <dirs> Override location of endorsed standards path
  -proc:<none,only> Control whether annotation processing and/or compilation is done.
  -processor <class1>[,<class2>,<class3>...] Names of the annotation processors to run; bypasses default discovery process
  -processorpath <path> Specify where to find annotation processors
  -d <directory>      Specify where to place generated class files
  -s <directory>      Specify where to place generated source files
  -implicit:<none,class> Specify whether or not to generate class files for implicitly referenced files
  -encoding <encoding> Specify character encoding used by source files
  -source <release>    Provide source compatibility with specified release
  -target <release>    Generate class files for specific VM version
  -version           Version information
  -help             Print a synopsis of standard options
  -Xkey[=value]      Options to pass to annotation processors
  -X                Print a synopsis of nonstandard options
  -J<flag>           Pass <flag> directly to the runtime system
  -Werror            Terminate compilation if warnings occur
  @<filename>        Read options and filenames from file
```

set JAVA_HOME

- Si el sistema no tiene las variables de entorno configuradas, se agrega el java_home de la siguiente manera

```
set JAVA_HOME=C:\Program Files\Java\jdk1.7.0_17  
set PATH=%PATH%;%JAVA_HOME%\bin
```


Compilar un programa de java

1. Crear el archivo Programa_001.java

```
public class Programa_001{  
    public static void main(String[] args)  
    {  
        int a = 5;  
        int b = 3;  
        int c = a + b;  
        System.out.println(c);  
    }  
}
```

2. Compilarlo con la instrucción javac

¿PREGUNTAS?