

# Introducción a Java

# Java: Motivaciones de su origen

- “Deja” atrás características “problemáticas”:
  - Punteros
  - Asignación de memoria (malloc)
  - Herencia múltiple
  - Sobrecarga de operadores
- Independiente de:
  - Tipo de computador
  - Sistema operativo
  - Sistema de ventanas (win32, Motif, etc...)

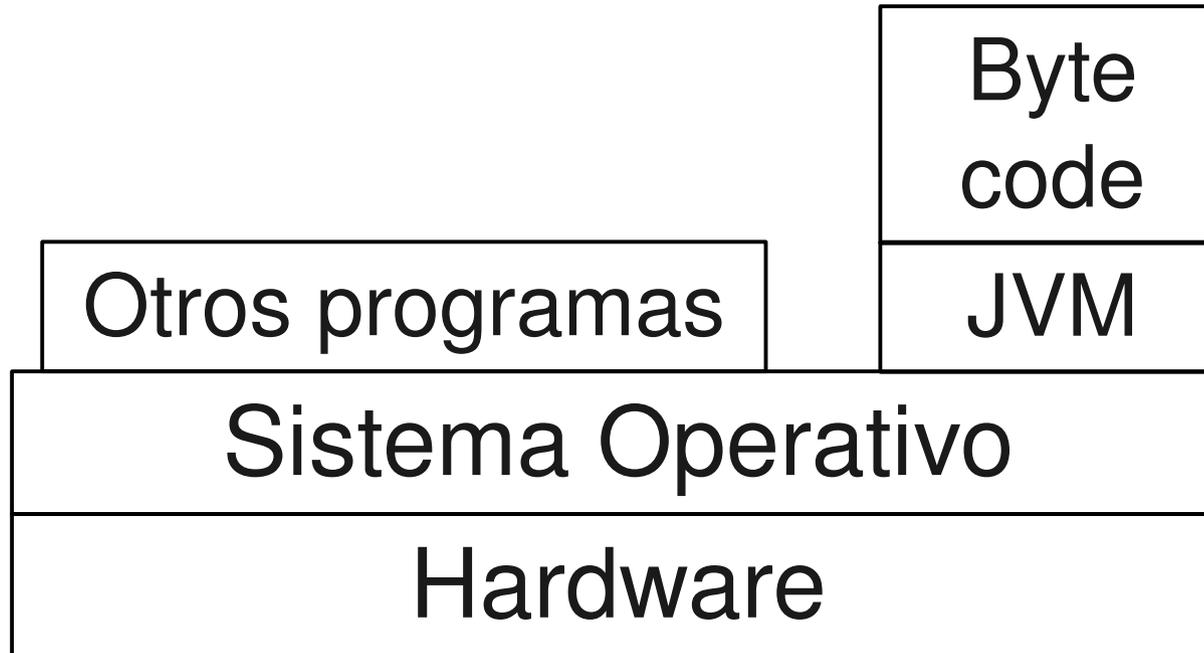
# Características “Problemáticas”

- Los punteros generan dificultades para muchos. No lo creo así para quienes estudian la estructura de un computador.
- Java tiene 8 tipos básicos (int, float, char, etc), todos los otros datos (todos los objetos) son almacenados en memoria dinámica (heap). Así el programador parece no relacionarse con la memoria dinámica.
- No se requiere liberar (free) la memoria solicitada por malloc, el lenguaje se preocupa por reutilizar la memoria liberada por objetos fuera de uso (aquellos sin nombre para ser accedidos desde el programa).
- Un problema de la herencia múltiple es la presencia de un mismo atributo en las dos clases desde donde se crea la herencia múltiple. Ayudante puede heredar de estudiante y de funcionario, pero en estudiante y en funcionario ya existe el campo nombre.

# Independiente del Computador y Sistema Operativo

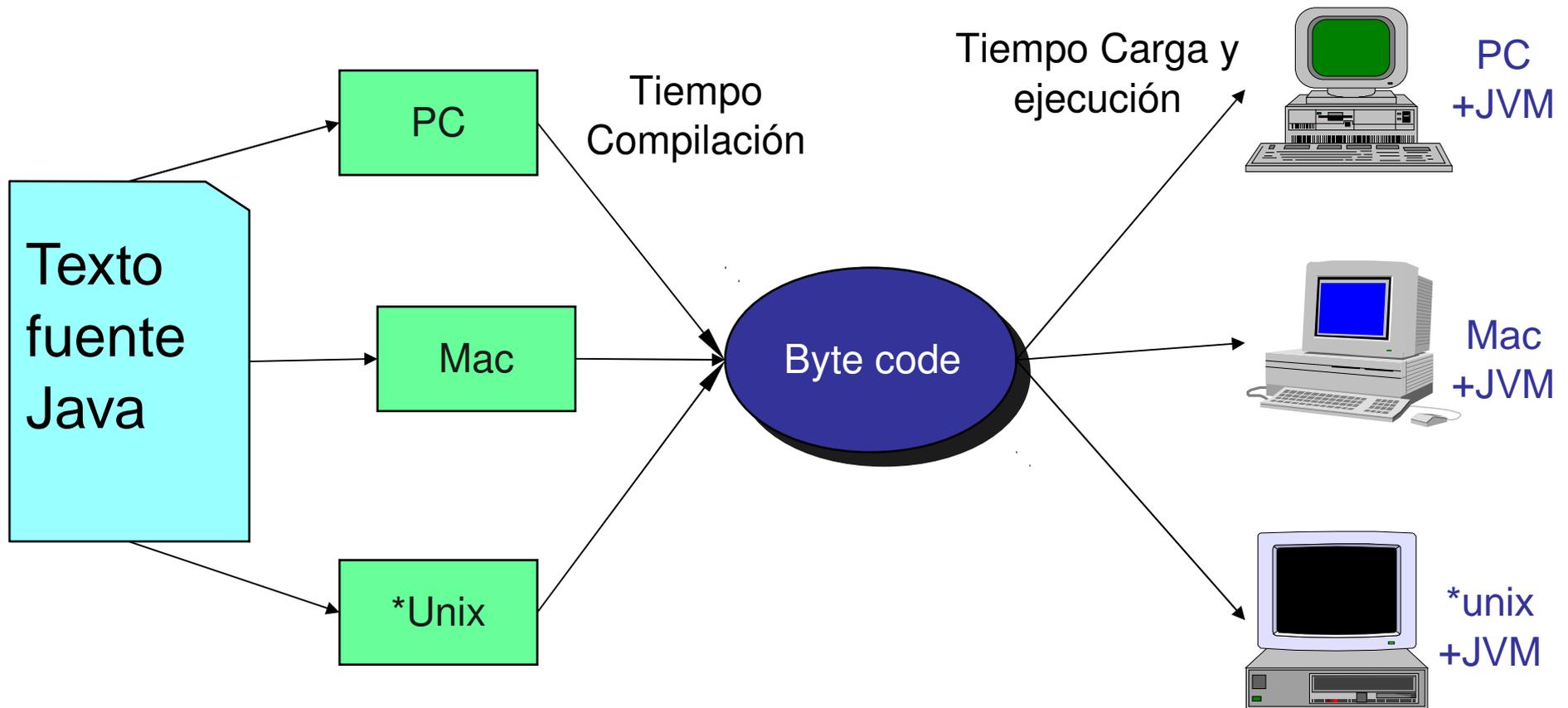
- Esto se logra por el uso de una **Máquina Virtual Java (Java Virtual Machine)**.
- Una máquina virtual es una abstracción de una máquina la cual es generada por software.
- ¿Han usado programas emuladores de Play Stations?
- ¿Han usado programas emuladores de PC dentro de un PC? Así podemos tener varios Sistemas operativos corriendo concurrentemente en la misma máquina.
- También este concepto es aplicable a sistemas operativos donde es posible crear la apariencia de tener varias máquinas independientes (jaulas o jails)

# Java Virtual Machine



- Para cada combinación hardware+SO se desarrolla una máquina virtual Java
- En todas las máquinas virtuales el programa compilado Java (byte code) corre “igual”
- Ver [java.sun.com](http://java.sun.com)

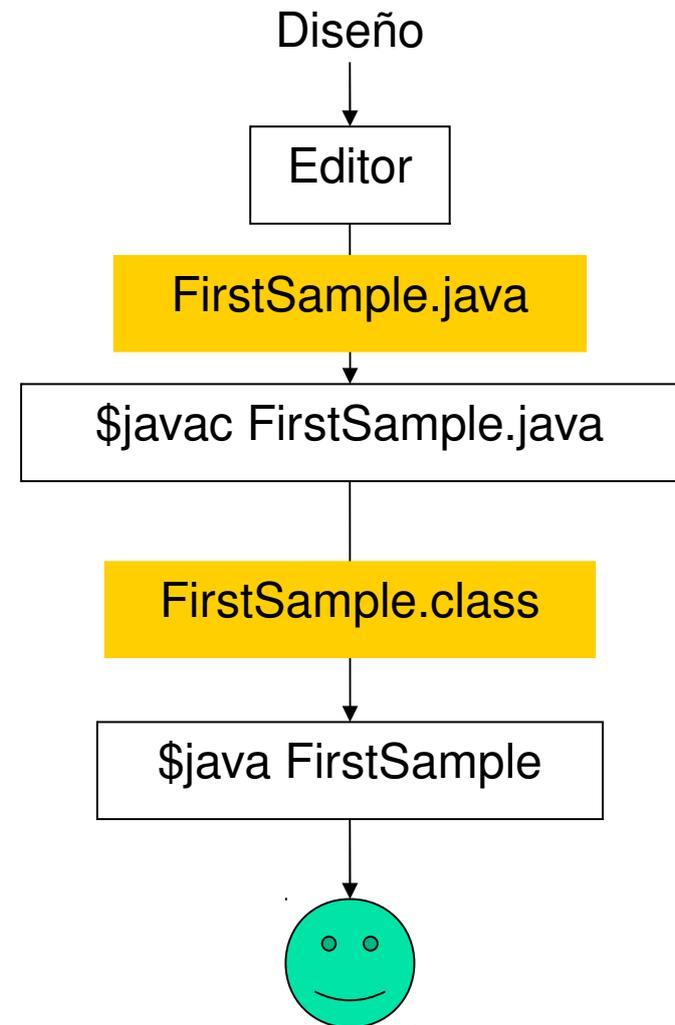
# Compilación



JVM es la **J**ava **V**irtual **M**achine,  
Una para cada plataforma.

# Trabajando con Java

- Creación programa: Con editor crear programa \*.java (FirstSample.java)
- Hacer uso de documentación en [manuales.elo.utfsm.cl](http://manuales.elo.utfsm.cl)
- **Compilación:** vía el comando el línea `$ javac FirstSample.java`  
La salida serán archivos .class, es la versión del programa en código byte.
- **Ejecución:**  
`$java FirstSample`  
Notar que java es el programa que corremos para crea la **máquina virtual** donde el byte code es ejecutado, equivale a una **interpretación** en la máquina real.
- Hay ambientes de trabajo más amigables para hacer estas tareas.



# Editores de texto

- Recomiendo aprender a digitar bien.
- **Emacs** (win o Linux) u otro. Para mi gusto un buen editor debería ayudar a indentar su programa.
- Usar ambientes integrados de Desarrollo (IDE) como:
  - **Jgraps**
  - Eclipse
  - Netbean (de la página de Sun)
- Hay otros, ver conveniencia.
- No usar notepad o similar. 
- Ver editores en página del ramo

# Sistema de Desarrollo

- Lo puede bajar de SUN:
- <http://java.sun.com>
- Versiones:
  - Java EE (Enterprise Edition),
  - Java SE (Standard Edition), <== Esta asignatura
  - Java ME (Micro-Edition)