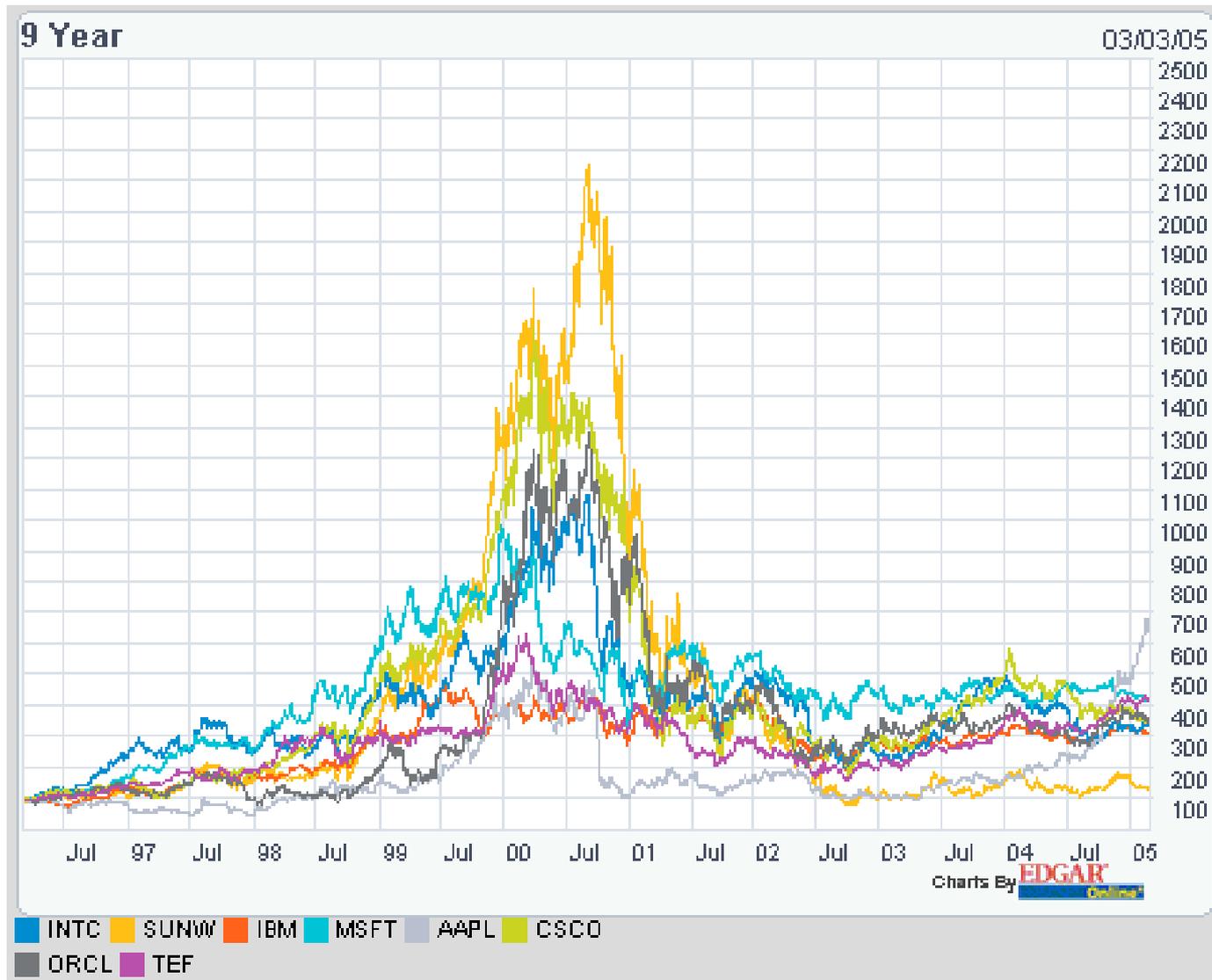


**TAP. ¿ES JAVA LA MEJOR OPCIÓN?**

Un componente importante del “boom” de las “dotcom” fue Java



Sun fue cada vez mejor representada por Java hasta el punto de llegar a cambiar su “ticker” en bolsa

Negocios y Mercados



## Sun Microsystems pasará a ser Java en el Nasdaq

25 ago 07 | 11:25 CET

*A partir del 27 de agosto, la compañía tecnológica cambiará su actual identificación bursátil, SUNW, por Java, su marca más reconocida en el mercado.*

Desde el próximo 27 de agosto, la identificación bursátil de la compañía tecnológica [Sun Microsystems](#) dejará de ser SUNW para pasar a ser **Java, su marca más reconocida en el mundo del software.**

En declaraciones reproducidas por [Europa Press](#), el presidente y consejero delegado de la empresa, Jonathan Schwartz, aseguró que esta modificación de las siglas de identificación bursátil “refleja una marca que todo el mercado puede identificar y supone un elemento importante del proceso de transformación de Sun a largo plazo”.

El directivo agregó que “Java está en todas partes, tocando de cerca a cualquiera que esté relacionado con Internet y es un símbolo de la capacidad de desarrollar, introducir y dar a conocer las novedades de Sun”.

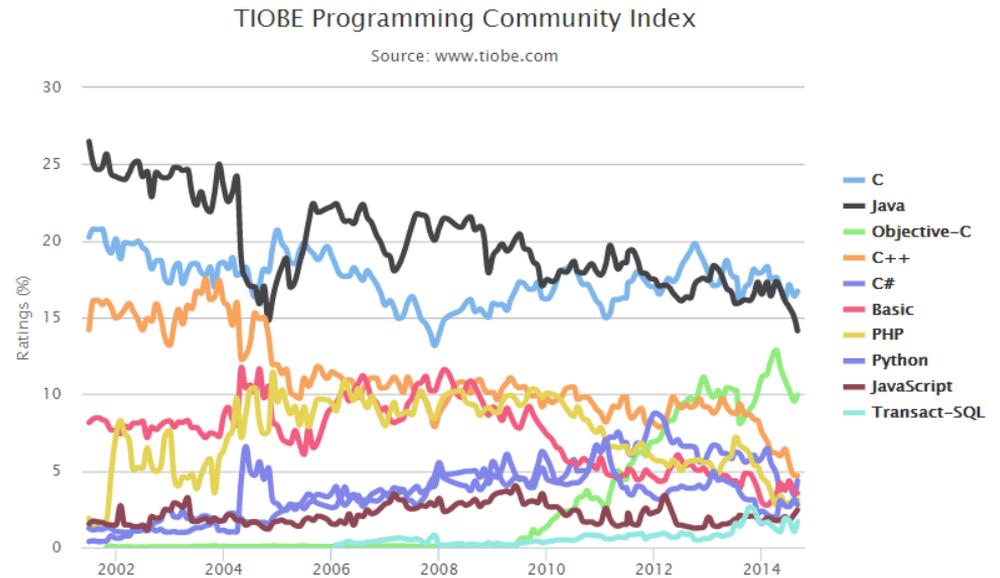
Cabe recordar que, según un informe de [Ovum](#) basado en estadísticas de mayo de este año, **existen 800 millones de ordenadores con software Java incorporado, 2.100 millones de dispositivos móviles para Java, 2.500 millones de tarjetas inteligentes y cerca de 180 operadores que ofrecen contenidos y servicios basados en esta tecnología.**

Curiosidad: Apple “relativiza” el boom de las “dotcom”



## Ordenación de lenguajes por presencia en la web

Sep 2014	Sep 2013	Change	Programming Language	Ratings	Change
1	1		C	16.721%	-0.25%
2	2		Java	14.140%	-2.01%
3	4	▲	Objective-C	9.935%	+1.37%
4	3	▼	C++	4.674%	-3.99%
5	6	▲	C#	4.352%	-1.21%
6	7	▲	Basic	3.547%	-1.29%
7	5	▼	PHP	3.121%	-3.31%
8	8		Python	2.782%	-0.39%
9	9		JavaScript	2.448%	+0.43%
10	10		Transact-SQL	1.675%	-0.32%
11	11		Visual Basic .NET	1.532%	-0.31%
12	12		Perl	1.369%	-0.32%
13	13		Ruby	1.281%	-0.10%
14	-	▲	Visual Basic	1.272%	+1.27%
15	14	▼	Delphi/Object Pascal	1.157%	+0.26%
16	26	▲	F#	0.990%	+0.49%
17	15	▼	Pascal	0.893%	+0.01%
18	-	▲	Swift	0.852%	+0.85%
19	19		MATLAB	0.818%	+0.18%
20	17	▼	PL/SQL	0.809%	+0.13%



<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

**¿QUÉ “CLASE” DE LENGUAJE ES JAVA?**

Del concepto de calculo efectivo a una

# Clasificación general de lenguajes

Euclides (Método axiomático), Aristóteles(Lógica formal), Muhammad ibn Musa Al'Khowarizmi (Algoritmo)...

1928 David Hilbert (Frege Russel Whitehead)

1931 Kurt Gödel

1938

Máquina de Turing



Alan Turing

## Lenguajes



Cálculo Lambda

Alonzo Church

*Sin olvidar las funciones recursivas de Herbrand-Gödel"*

Imperativos

- Fortran
- Cobol
- Pascal
- C
- ...

Funcionales

- Lisp
- Scheme
- ML
- Hope
- CLOS
- Ocaml
- ...
- Haskell
- Clojure

Lógicos

- Prolog
- ...



O/B objetos

- Object Pascal
- C++
- Javascript
- Java
- ...

Frameworks

- Ruby on Rails
- ...

- Javascript
- Scala
- Java 8

```
let rec long = function
| [] -> 0
|x::xs -> 1 + long xs;;

let rec ordenar = function
| [] -> []
|x::xs -> insertar x (ordenar xs)
and insertar e = function
| [] -> [e]
|x::xs -> if x > e
then e::xs
else x::(insertar e xs);;
```

Ejemplo OCaml

## Alan Turing - Wikipedia, la enciclopedia libre

[es.wikipedia.org/wiki/Alan\\_Turing](https://es.wikipedia.org/wiki/Alan_Turing) ▼

Alan Mathison **Turing**, OBE (Paddington, Londres, 23 de junio de 1912 - Wilmslow, Cheshire, 7 de junio de 1954), fue un matemático, lógico, científico de la ...

### Máquina de Turing

Una máquina de Turing es un dispositivo que manipula ...

### Enigma

Enigma era el nombre de una máquina que disponía de un ...

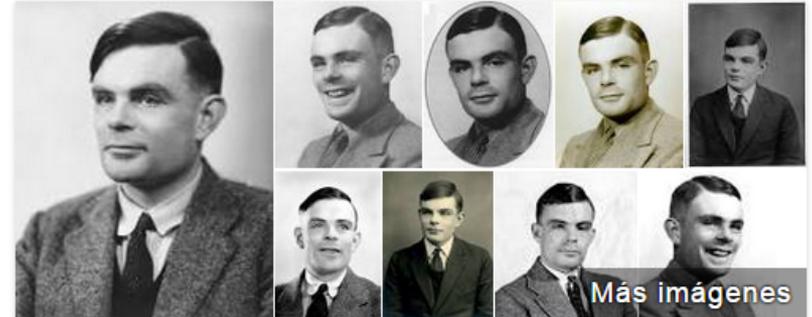
[Más resultados de wikipedia.org](#) »

### Tesis de Church-Turing

En teoría de la computabilidad, la tesis de Church-Turing formula ...

### Problema de la parada

El problema de la parada o problema de la detención para ...



## Alan Turing

Matemático

Alan Mathison Turing, OBE, fue un matemático, lógico, científico de la computación, criptógrafo y filósofo británico. Es considerado uno de los padres de la ciencia de la computación siendo el precursor de la informática moderna. [Wikipedia](#)

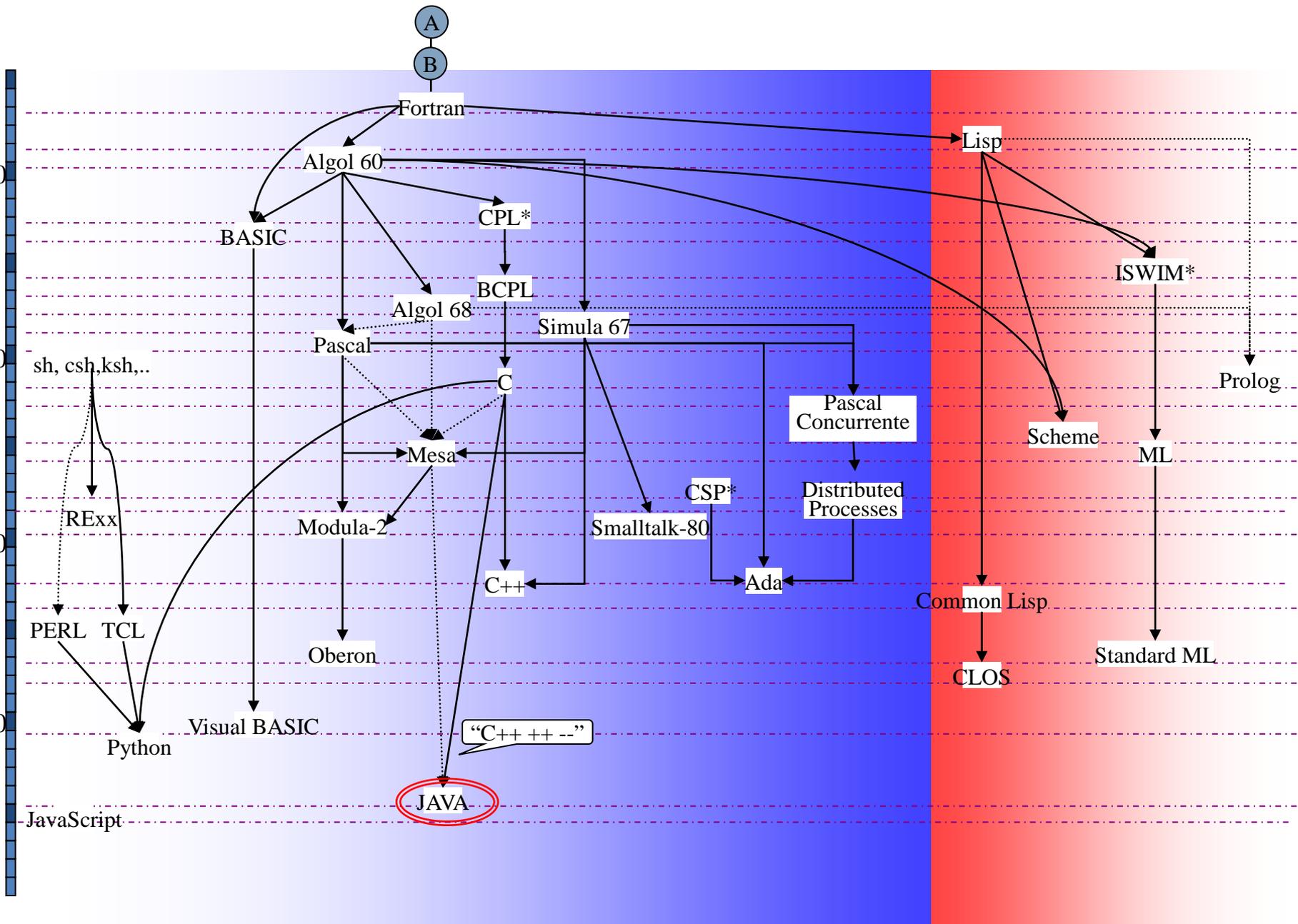
**Fecha de nacimiento:** 23 de junio de 1912, Maida Vale, Londres, Reino Unido

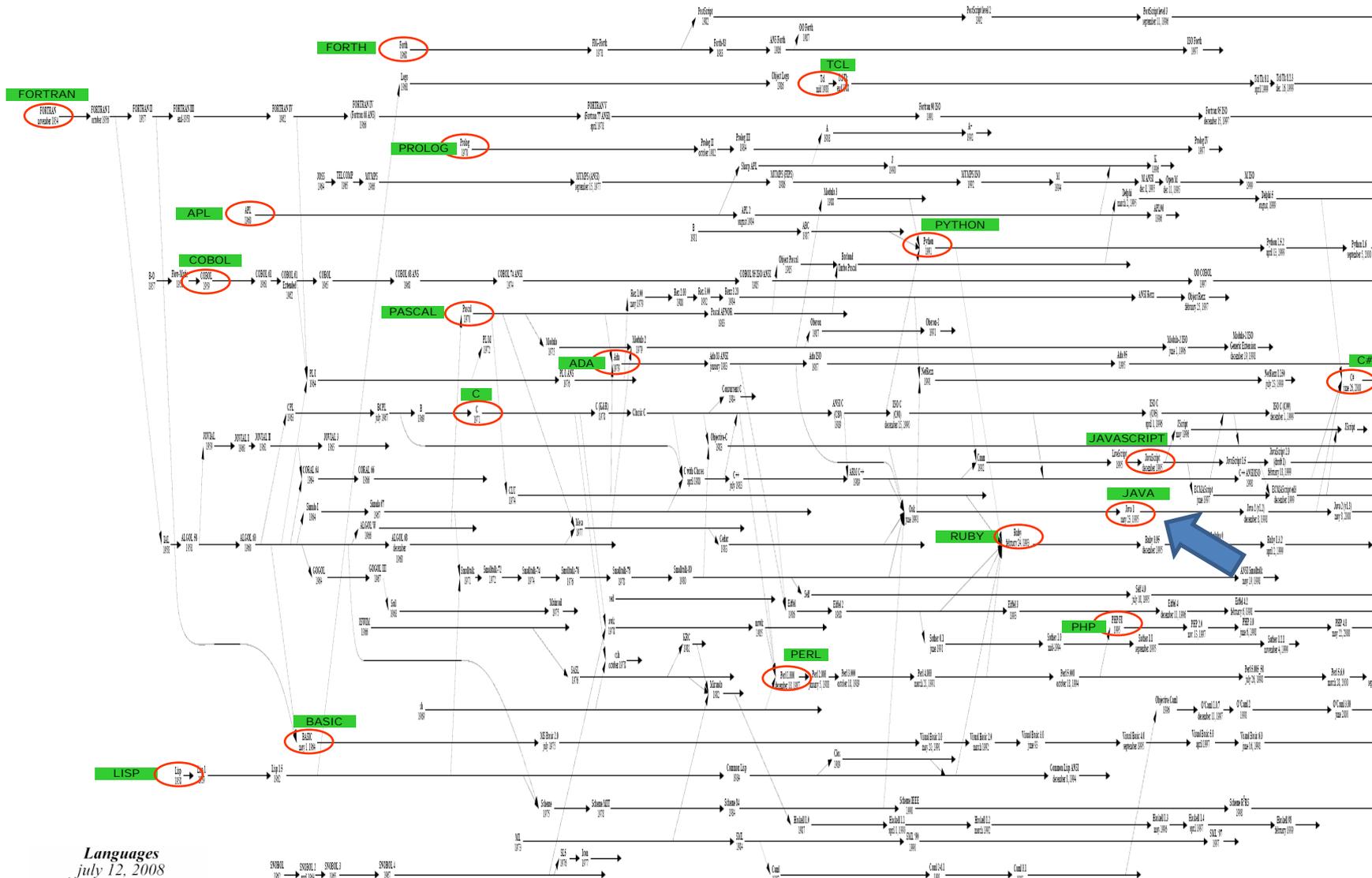
**Fecha de la muerte:** 7 de junio de 1954, Wilmslow, Reino Unido

**Libros:** [Computing machinery and intelligence](#), [¿Puede pensar una máquina?](#)

**Padres:** [Julius Mathison Turing](#), [Ethel Sara Stoney](#)

**Hermanos:** [John Turing](#)





Languages  
July 12, 2008

© Eric Lévénez 1999-2008

< <http://www.levenez.com/lang/> >

<http://www.levenez.com/lang/>

<http://gtts.ehu.es/German>

# ¿ES JAVA LENTO?

Es ya una cuestión muy antigua y resuelta, pero sirve para ilustrar aspectos de la ingeniería del software: programar no es sólo escribir programas

# ¿Java para cómputo intensivo?

Esta página contiene apreciaciones discutibles.  
(pero las conclusiones son VERDAD) 

(X=matemáticas, física, ingeniería...)

Tradicionalmente, en computación para X, se ha buscado la “velocidad”.

Esta visión X=cómputo intensivo puede ser cierta pero quizás parcialmente.

En todo caso MUCHÍSIMAS de las necesidades de computación “al límite” de hace unos años, hoy en día son livianas o “razonables”. (mi conjetura: El mundo de lo “intratable” por “impotencia computacional” se ha reducido enormemente)

Esta ¿obsesión?/¿necesidad? justificaba el inmovilismo de las X frente a nuevos lenguajes (debería llevarles a programar directamente los microprocesadores en su lenguaje ensamblador, pero curiosamente no se daba esto).

Resultado: FORTRAN es la referencia, y el razonable paso a C ¿se dio?.



La Web Resultados 1 - 10 de aproximadamente 122.000 de **mathematics "in FORTRAN" program.** (0,22 segundos)

La Web Resultados 1 - 10 de aproximadamente 1.630.000 de **mathematics "in C" program.** (0,30 segundos)

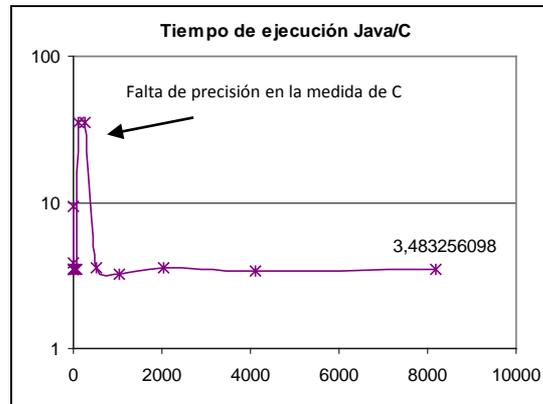
(comparación inválida sin duda. La presencia “arrasadora” de C junto a la “inteligencia” de Google (que utiliza el sinónimo “math” en la búsqueda) potencian el segundo resultado.

A Java se le ha “acusado” desde un principio de ser LENTO.

- Al principio era cierto.
  - Relación 4/1 frente a C
  - Razón principal: lenguaje interpretado
  - Otras razones: recogida de basuras, mecanismos de seguridad, etc.
- Desde hace unos años es comparable a C, dependiendo de en qué tareas. Ciertamente no es el mejor caso el del cómputo intensivo (estimaciones de un estudio particular 2004).
  - En gráficos bate a C
  - Relación media: 1.4/1 frente a C si excluimos gráficos
  - En cálculo intensivo la diferencia es más acusada
- Actualmente Java es más rápido que C en muchas tareas (particularmente gráficos) y similar en los peores casos (cómputo intensivo)
  - Máquina HotSpot
- Java (los lenguajes sobre máquinas virtuales en general) serán los más rápidos en el futuro (ante cómputos complejos, no para el caso de algoritmos muy “cerrados”)
  - El futuro de las máquinas virtuales

Experimento a partir del código tomado de <http://husnusensoy.blogspot.com/2006/06/c-vs-java-in-number-crunching.html>

- Comparación del tiempo de ejecución



La relación de tiempo de ejecución es del orden de 3,5 a favor de C

- Comparación del tiempo de preparación del experimento

JAVA:

- copiar, pegar, compilar, ejecutar y **listo en unos segundos.**

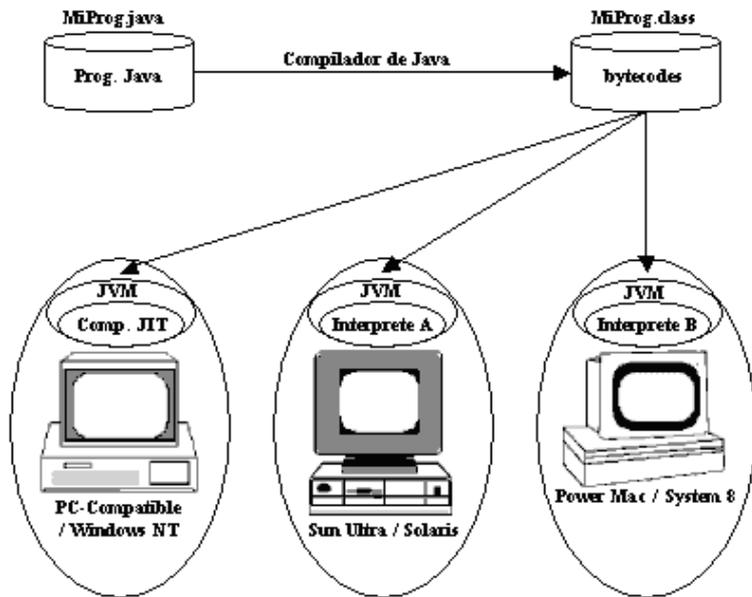
C:

- copiar, pegar, compilar, errores... (no coincide exactamente el lenguaje)
- corregir fuente, compilar, ejecutar, errores... (la arquitectura de la máquina no es la adecuada)
- corregir fuente, compilar, ejecutar, se observar falta de resolución de la función "time",
- ir a la bibliografía para resolver el tema, no encontrar solución...
- replantear con iteraciones para obtener tiempos mayores...
- cambiar fuente compilar, ejecutar... errores de apuntadores (falta de práctica de un "ex" de C)
- corregir fuente, compilar, ejecutar... errores de violación de segmentos
- corregir fuente, compilar, ejecutar y... **listo en una hora.**

La relación de tiempo de preparación ha sido de 120 a favor de Java

# LA MÁQUINA VIRTUAL

# "WRITE ONCE, RUN ANYWHERE"



## Proprietary/closed source implementations

- \* Hewlett-Packard's Java for HP-UX, OpenVMS, Tru64 and Reliant (Tandem) UNIX platforms
  - \* J9 VM from IBM, for AIX, Linux, MVS, OS/400, Pocket PC, z/OS
  - \* Mac OS Runtime for Java (MRJ) from Apple Inc.
  - \* JRockit from BEA Systems acquired by Oracle Corporation
  - \* Oracle JVM (also known as "JServer" and as "OJVM") from Oracle Corporation
  - \* Microsoft Java Virtual Machine (MS JVM) from Microsoft
  - \* PERC from Aonix is a real time Java for embedded
  - \* JBed from Esmertec is an embedded Java with multimedia capabilities
  - \* JBlend from Aplix is a Java ME implementation
  - \* Excelsior JET (with AOT compiler)
- ## Lesser-known proprietary JVMs
- \* Blackdown Java (port of Sun JVM)
  - \* CVM
  - \* Gemstone Gemfire JVM - modified for J2EE features
  - \* Golden Code Development (EComStation and OS/2 port of Java RTE and SDK for J2SE v1.4.1\_07)
  - \* Tao Group's intent
  - \* Novell, Inc.
  - \* NSIcom CrE-ME
  - \* HP ChaiVM and MicrochaiVM
  - \* MicroJVM from Industrial Software Technology (running of wide range of microcontrollers 8/16/32-bit)

## Free/open source implementations

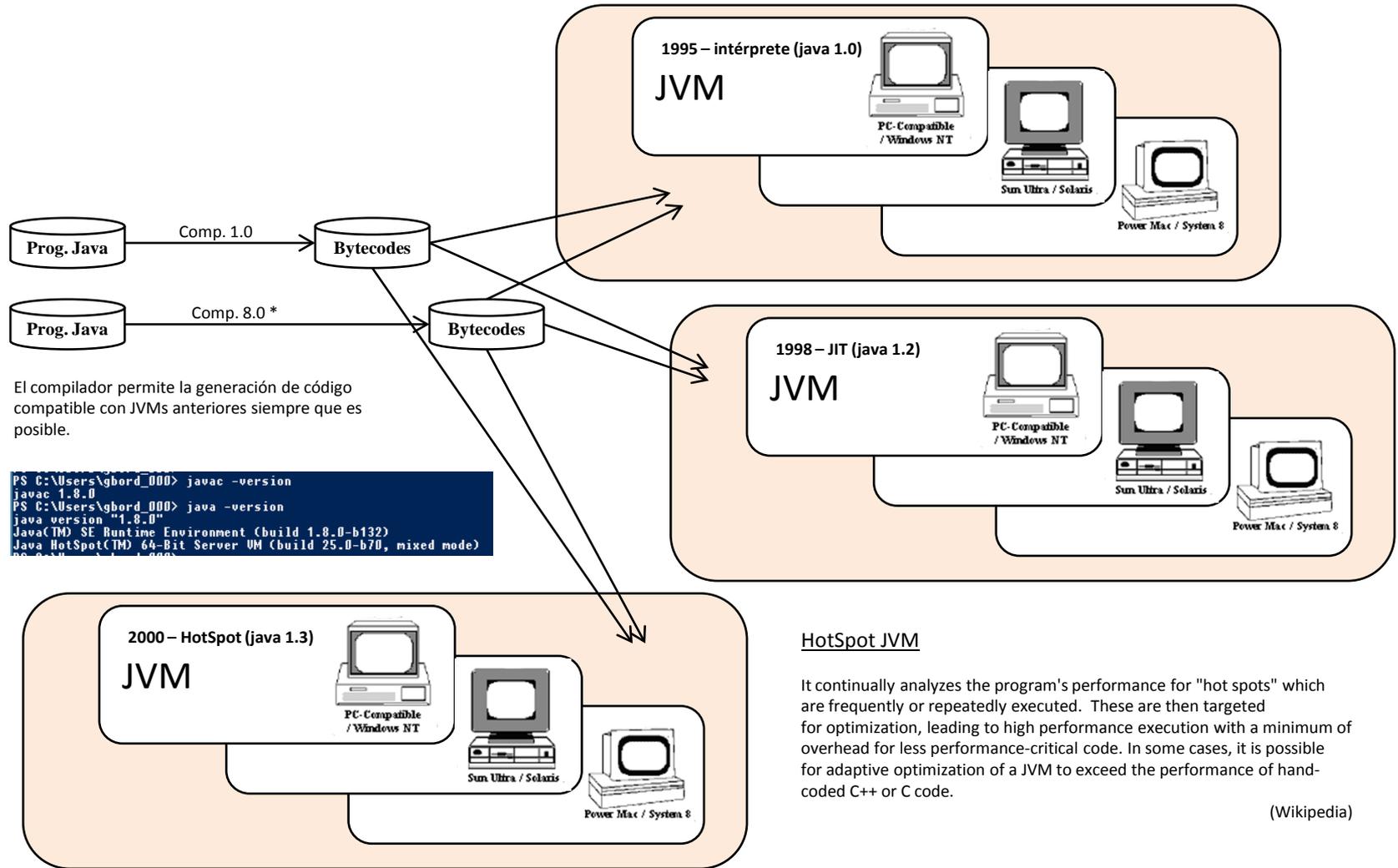
- |                  |             |                         |               |
|------------------|-------------|-------------------------|---------------|
| * AegisVM        | * JamVM     | * Juice                 | * Mika VM     |
| * Apache Harmony | * Jaos * JC | * Jupiter JVM           | * Mysaifu JVM |
| * CACAO          | * Jikes RVM | * JX (operating system) | * NanoVM      |
| * IcedTea        | * JNode     | * Kaffe                 | * SableVM     |
| * IKVM.NET       | * JOP       | * leJOS                 | * SuperWaba   |
| * Jamiga         |             |                         | * TinyVM      |
- \* JESSICA (Java-Enabled Single-System-Image Computing Architecture)
  - \* Squawk virtual machine (Sun JVM for embedded system and small devices)
  - \* Sun Microsystems' HotSpot
  - \* VMkit of Low Level Virtual Machine
  - \* Wonka VM
  - \* Xam



- Una idea novedosa, pero no del todo: cierta similitud con los lenguajes con código intermedio.
- Sí fue novedoso el enfoque de emulador de máquina (y la compilación JIT).
- Ventajas:
  - se pueden incluir con facilidad técnicas que en un diseño hardware pueden resultar prohibitivas por su complejidad técnica,
  - la posibilidad de evolución es mucho más sencilla al no requerir cambios de hardware
  - permite utilizar las "plataformas" existentes sin implicar una ruptura con los sistemas actuales (existe la máquina real pero...).
- el diseño es público y la "implementación" es privada [especificaciones técnicas que debe cumplir toda JVM.](#) )
  - Distintos comportamientos en términos de velocidad y uso de memoria

En la máquina Virtual está el “secreto”...

además aporta otras muchas ventajas.



```
PS C:\Users\gbord_000> javac -version
javac 1.8.0
PS C:\Users\gbord_000> java -version
java version "1.8.0"
Java(TM) SE Runtime Environment (build 1.8.0-b132)
Java HotSpot(TM) 64-Bit Server VM (build 25.0-b70, mixed mode)
```

### HotSpot JVM

It continually analyzes the program's performance for "hot spots" which are frequently or repeatedly executed. These are then targeted for optimization, leading to high performance execution with a minimum of overhead for less performance-critical code. In some cases, it is possible for adaptive optimization of a JVM to exceed the performance of hand-coded C++ or C code.

(Wikipedia)

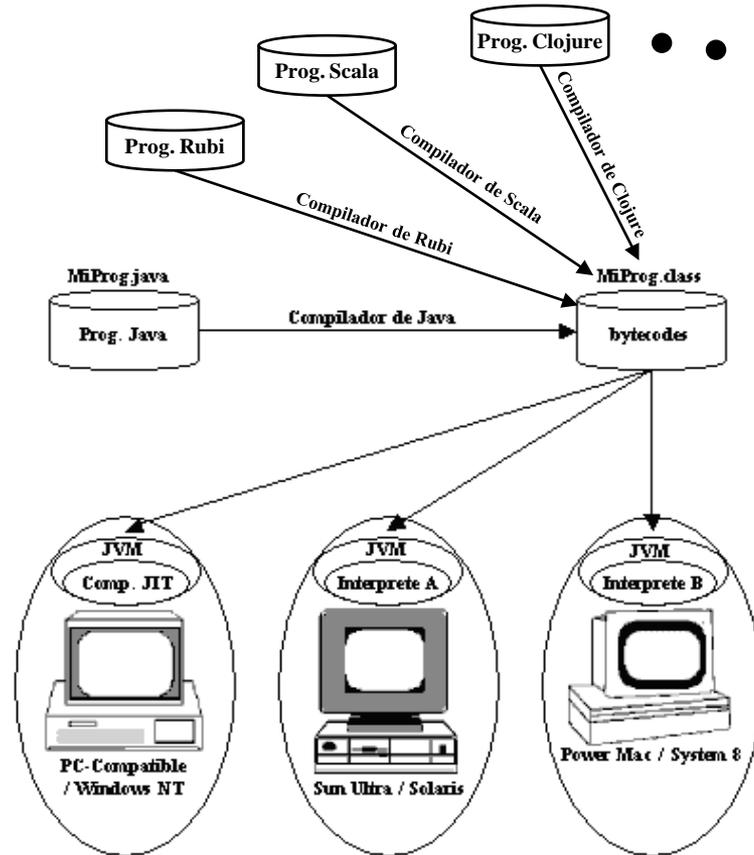
Otros cambios importantes:

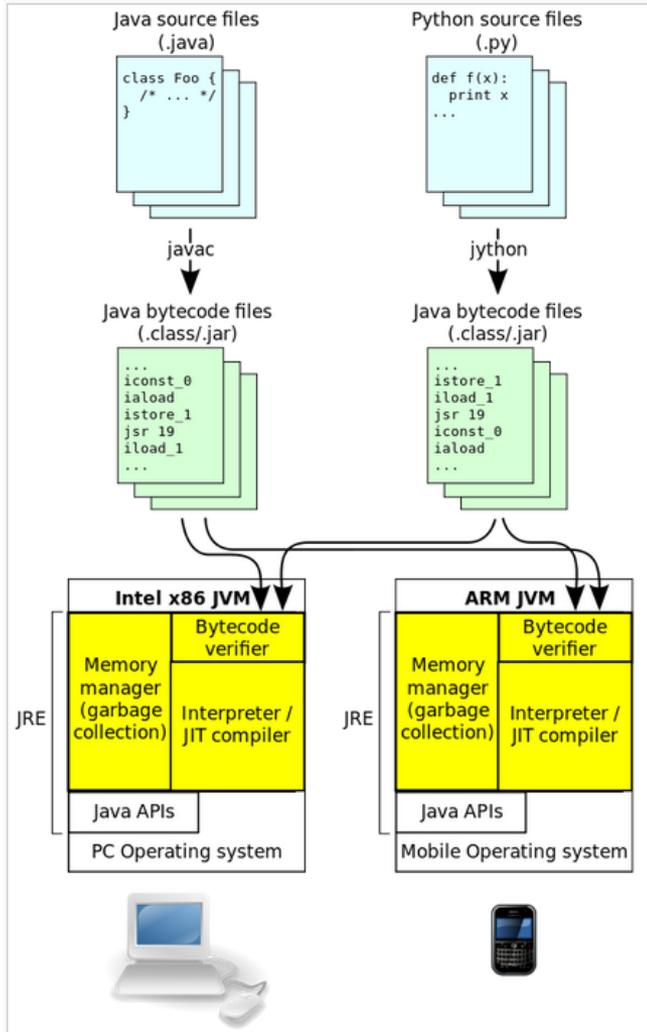
Java 1.5: colecciones, genéricos, enumeraciones...

Java 1.8 Versión actual (marzo 2014): programación funcional,...

En la máquina Virtual está el “secreto”...

y no debemos confundirla con el lenguaje.





Wikipedia: Java virtual machine (sep 2014)

**Versions of non-JVM languages**

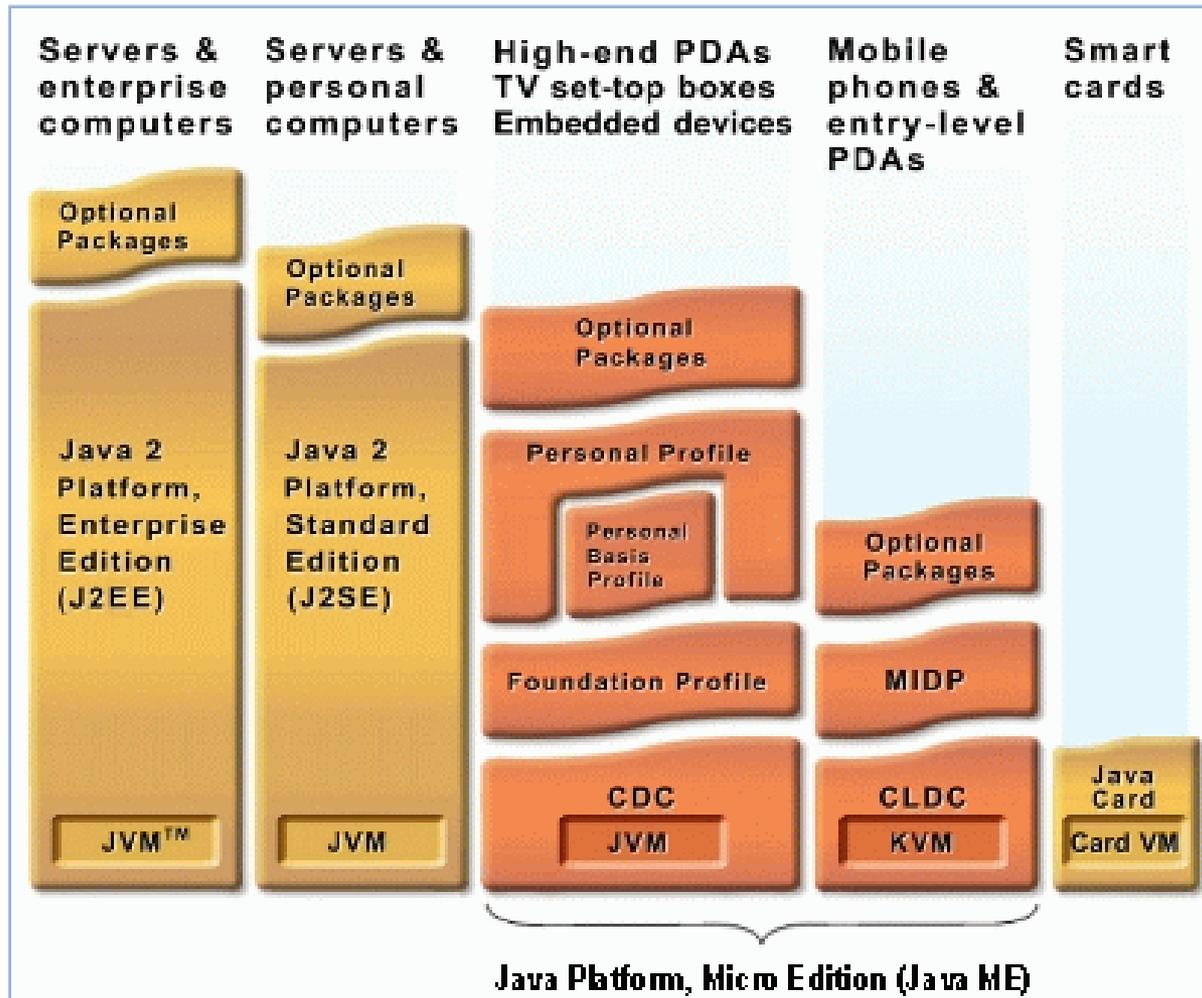
Language	On JVM
Erlang	Erjang
JavaScript	Rhino
Pascal	Free Pascal
PHP	Quercus
Python	Jython
REXX	NetRexx <sup>[3]</sup>
Ruby	JRuby
Tcl	Jacl

**Languages designed expressly for JVM**

Language
BBj
Clojure
Fantom
Groovy
MIDletPascal
Scala
Kawa

Wikipedia: Java virtual machine (sep 2014)

## Hay (al menos) tres “grados” de Máquinas Virtuales Java



... luego esta la de Android (Dalvik)

(ART para android L a finales de 2014)