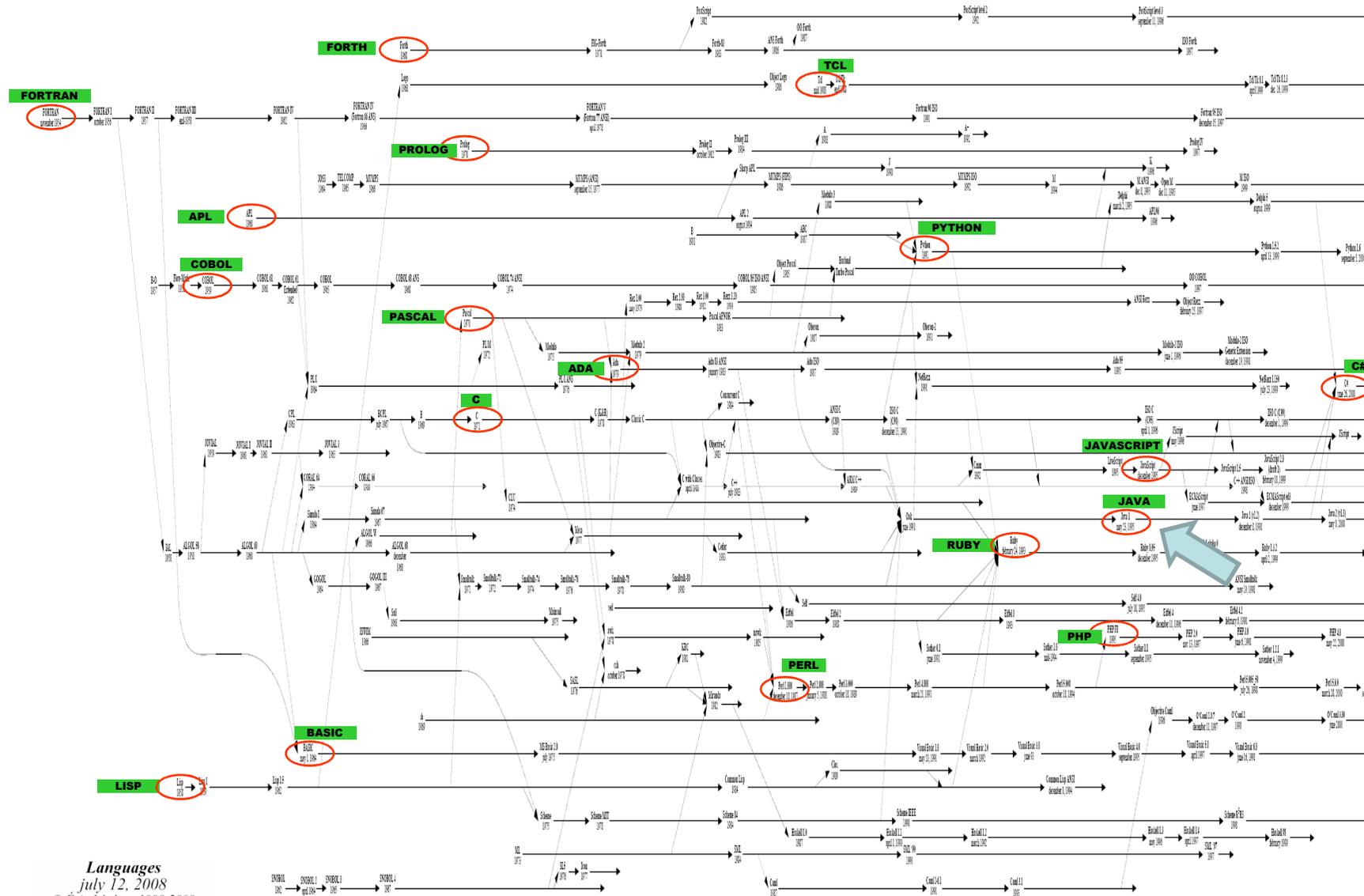


Digresión previa

¿Dónde está Java en el universo de los lenguajes?...



1954 1957 1960 1965 1970 1975 1980 1985 1990 1995 2000



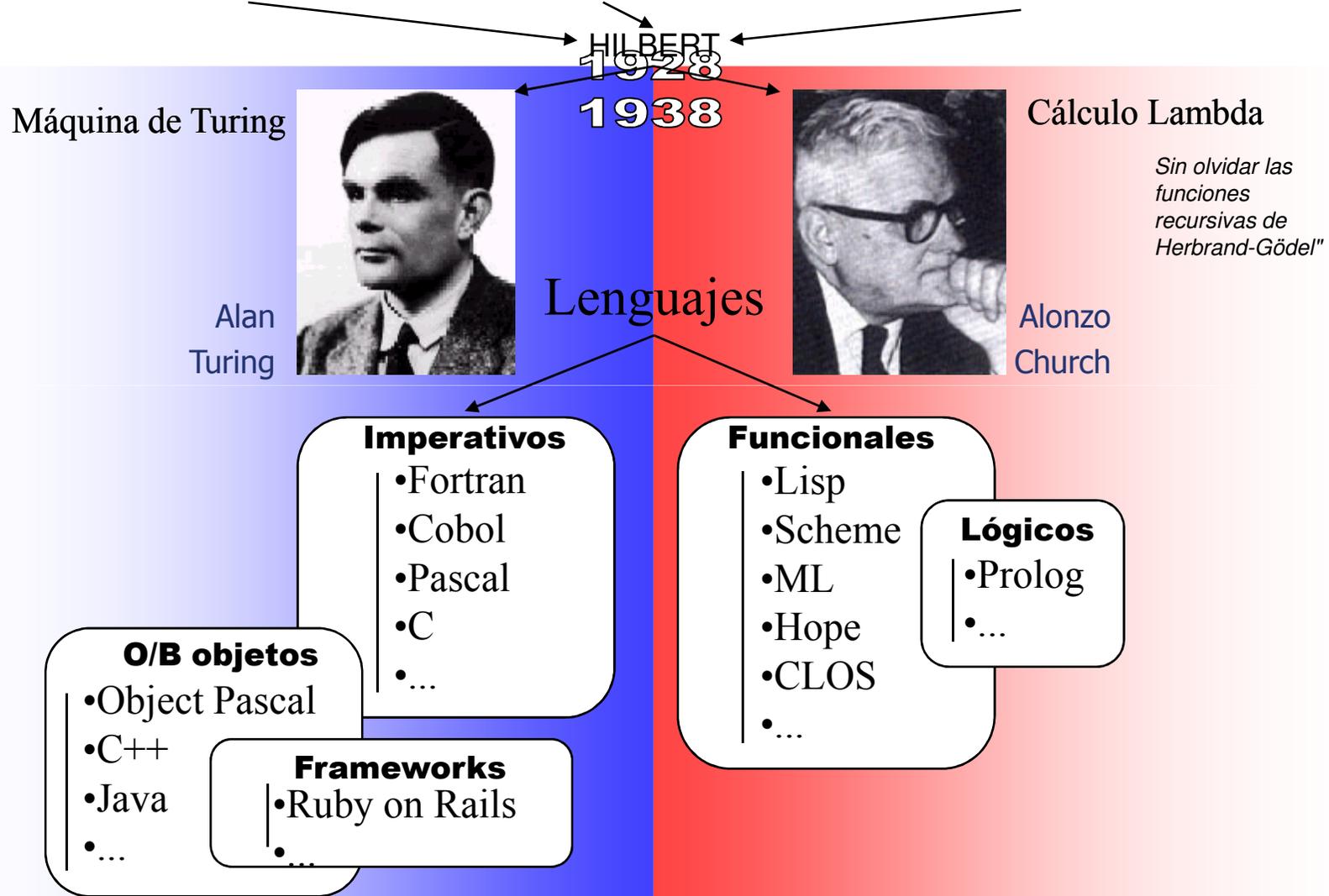
Languages
 July 12, 2008
 © Eric Lévesque 1999-2008
 <http://www.levenez.com/lang/>

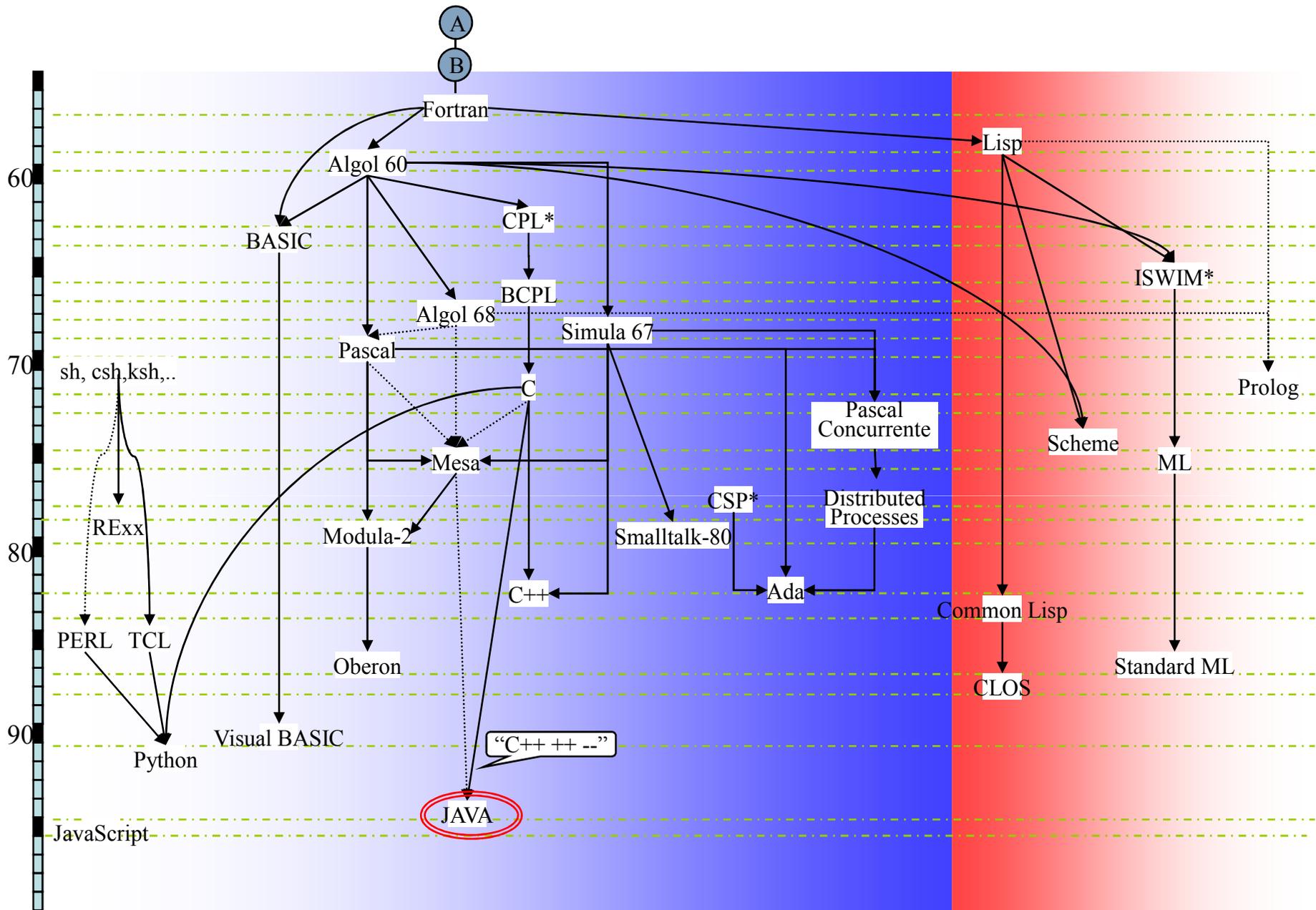
La Programación ES y VIENE DE las Matemáticas... (como parte de la computación)

Del concepto de calculabilidad efectiva a una

Clasificación general de lenguajes

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

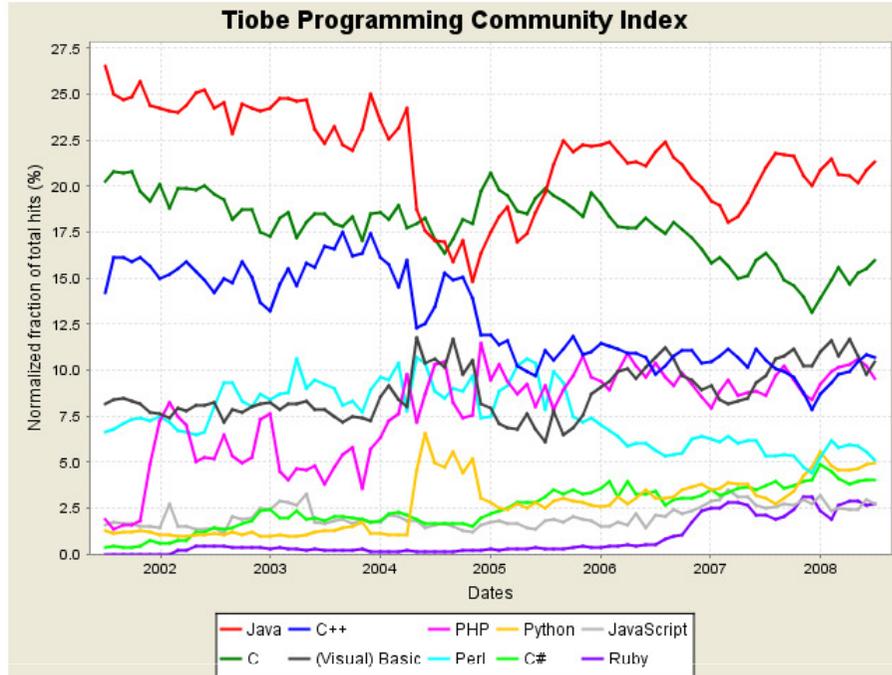




Ordenación de lenguajes por uso

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

Position Jul 2008	Position Jul 2007	Delta in Position	Programming Language	Ratings Jul 2008	Delta Jul 2007	Status
1	1	=	Java	21.345%	+0.33%	A
2	2	=	C	15.945%	-0.42%	A
3	3	=	C++	10.693%	+0.19%	A
4	4	=	(Visual) Basic	10.447%	+0.72%	A
5	5	=	PHP	9.525%	+0.87%	A
6	6	=	Perl	5.131%	-0.20%	A
7	8	↑	Python	4.973%	+1.95%	A
8	7	↓	C#	4.000%	+0.29%	A
9	9	=	JavaScript	2.757%	+0.24%	A
10	10	=	Ruby	2.735%	+0.64%	A
11	13	↑↑	Delphi	2.011%	+0.51%	A
12	14	↑↑	D	1.224%	-0.14%	A
13	11	↓↓	PL/SQL	0.736%	-1.23%	A
14	12	↓↓	SAS	0.629%	-1.23%	B
15	-	↑↑↑↑↑↑↑↑	PowerShell	0.436%	+0.44%	B
16	23	↑↑↑↑↑↑	Pascal	0.432%	-0.05%	B
17	17	=	COBOL	0.413%	-0.17%	B
18	18	=	Lua	0.408%	-0.17%	B
19	16	↓↓↓	Ada	0.407%	-0.20%	B
20	15	↓↓↓↓↓	Lisp/Scheme	0.384%	-0.30%	B

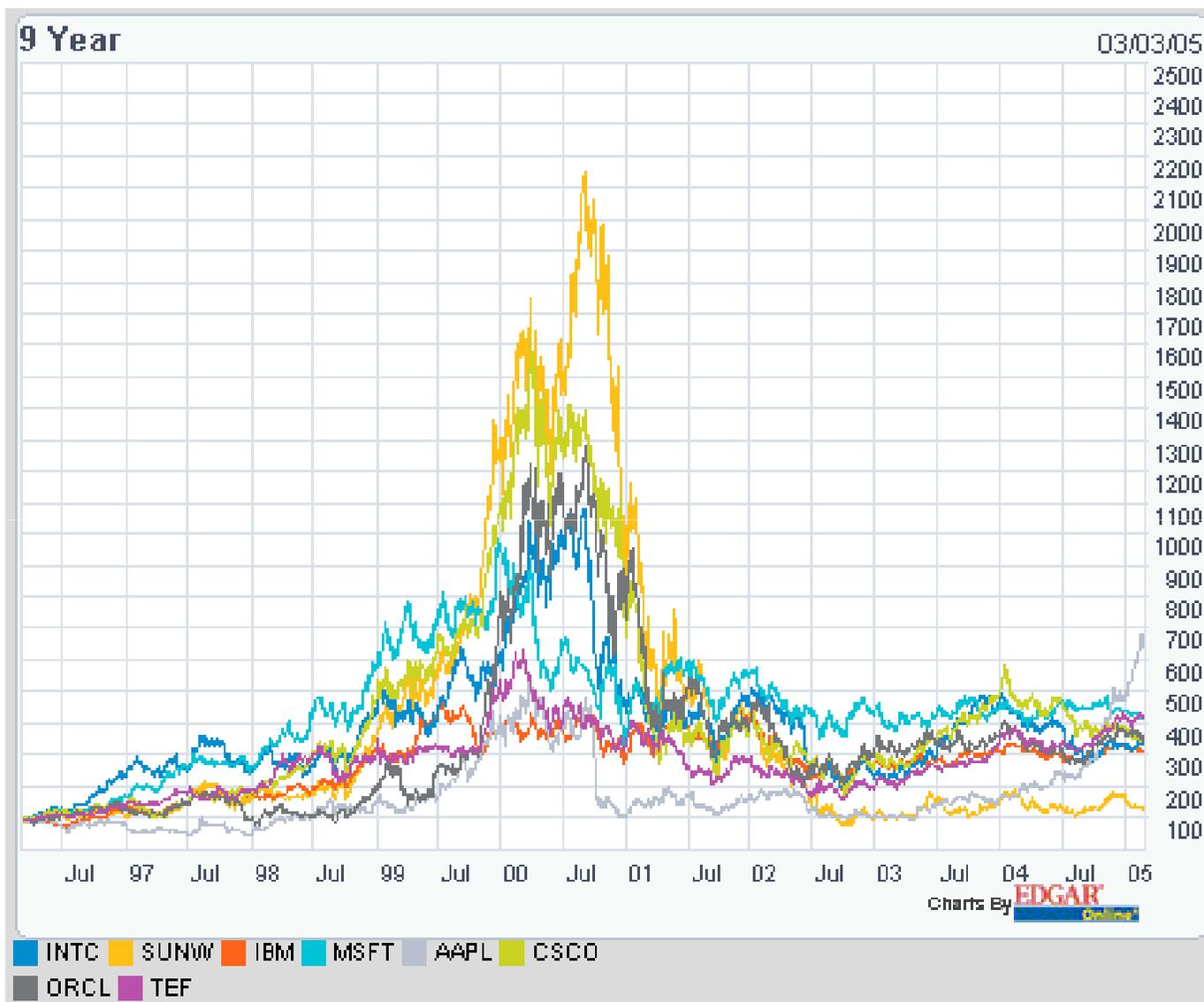


Position	Programming Language	Ratings	Position	Programming Language	Ratings
21	Logo	0.367%	36	Erlang	0.121%
22	ActionScript	0.363%	37	DC	0.118%
23	FoxProxBASE	0.357%	38	Groovy	0.117%
24	RPG (OS/400)	0.328%	39	Haskell	0.109%
25	Fortran	0.292%	40	ML	0.109%
26	ABAP	0.261%	41	Tcl/Tk	0.105%
27	MATLAB	0.253%	42	CL (OS/400)	0.101%
28	Prolog	0.221%	43	Forth	0.092%
29	Awk	0.200%	44	Lingo	0.089%
30	Transact-SQL	0.164%	45	REXX	0.088%
31	LabVIEW	0.159%	46	Caml	0.087%
32	Bourne shell	0.158%	47	Objective-C	0.085%
33	PL/I	0.152%	48	Natural	0.082%
34	Euphoria	0.151%	49	R	0.082%
35	Smalltalk	0.124%	50	APL	0.081%

21 a 50

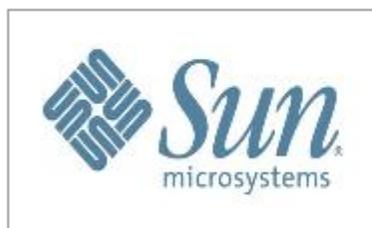
51 a 100

ABC, AD, Algol, Alpha, Applescript, AspectJ, Beta, Boo, C shell, cg, Ch, Clean, cT, Curl, Dylan, Eiffel, Factor, Felix, Focus, Icon, IDL, Inform, Io, MAD, Magic, Maple, Mathematica, Modula-2, MOO, MUMPS, Oberon, Occam, Oz, Pike, PILOT, Postscript, PowerBuilder, Progress, Q, REALbasic, Rebol, S-lang, Scala, SIGNAL, SPSS, TOM, VBScript, Verilog, VHDL, XSLT





Negocios y Mercados



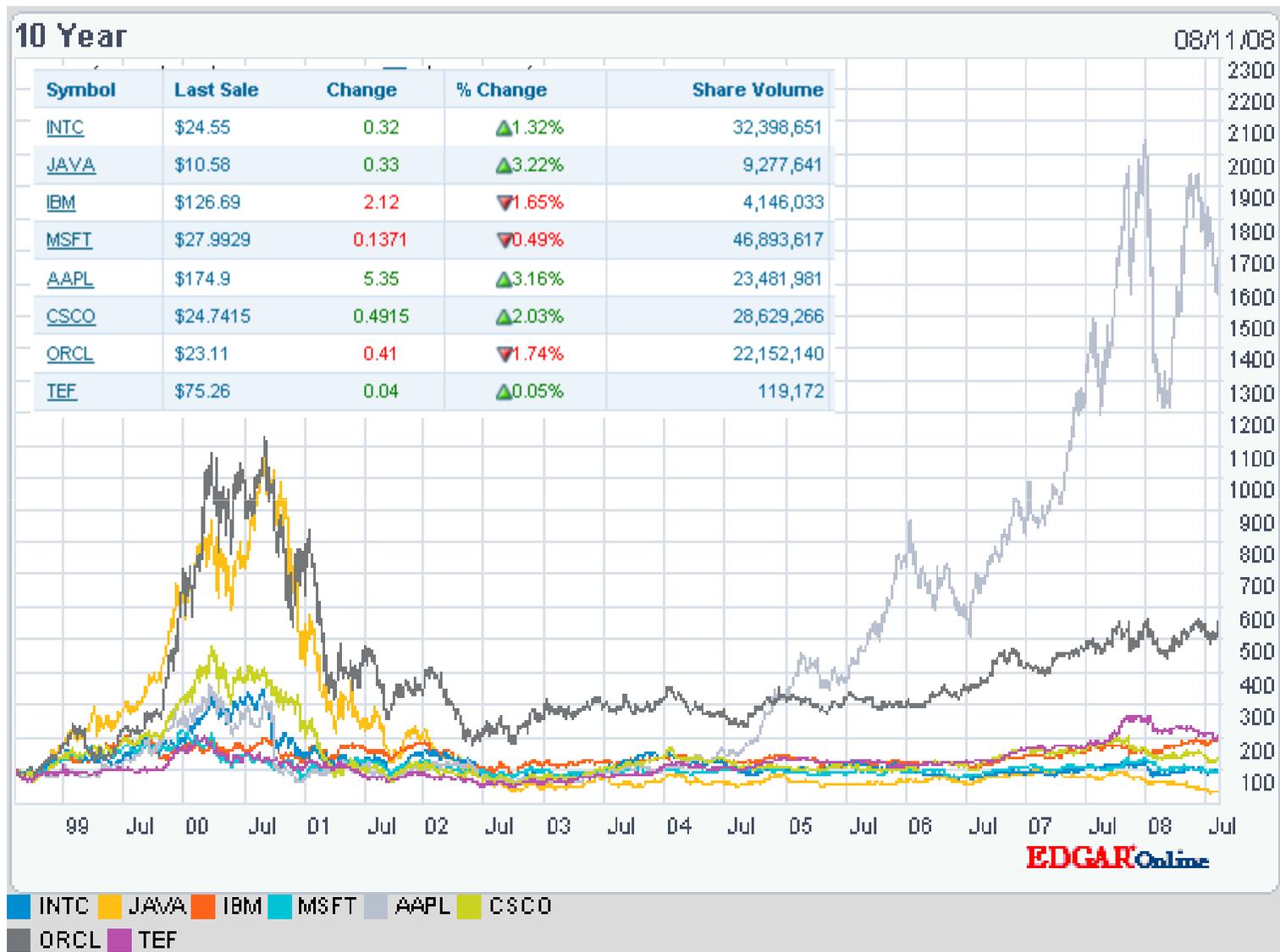
Sun Microsystems pasará a ser Java en el Nasdaq

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.**



1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. *Características y evolución.*

1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. *Características y evolución.*

Esta página contiene apreciaciones discutibles.

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

Esta visión matemáticas=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 los matemáticos 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. (desarrollo→)

- Al principio era cierto.
- Desde hace unos años es comparable a C, dependiendo de en qué tareas. Ciertamente no es el mejor caso el del cómputo intensivo.
- Actualmente Java es más rápido que C en muchas tareas (particularmente gráficos) y similar en los peores casos (cómputo intensivo)
- Java (los lenguajes sobre máquinas virtuales en general) serán los más rápidos en el futuro (ante cálculos complejos, no para el caso de algoritmos muy “cerrados”)

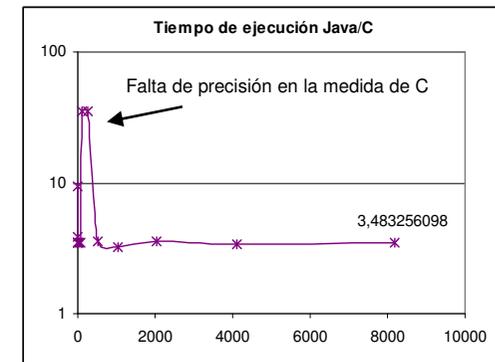
1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. Características y evolución.

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

Ejemplo peor caso Java vs. C (14ago08)
(cálculo de PI por MonteCarlo)

- La relación de tiempo de ejecución es del orden de 3,5 a favor de C
- La relación de tiempo de preparación del experimento a partir del código tomado de <http://husnusensoy.blogspot.com/2006/06/c-vs-java-in-number-crunching.html> ha sido del orden de 120 a favor de Java
 - JAVA: copiar, pegar, compilar, ejecutar y listo en unos segundos.
 - C: copiar, pegar, compilar, corregir fuente, compilar, ejecutar, observar falta de resolución de la función “time”, ir a la bibliografía para resolver el tema, no encontrar solución, reescribir con iteraciones para obtener tiempos mayores, compilar, resolver problemas de apuntadores, compilar, ejecutar, resolver problemas de violación de segmentos, compilar, ejecutar y listo en una hora a causa de las iteraciones en la búsqueda de precisión.



1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. **Características y evolución.**

1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. Características y evolución.

- Simple
- Orientado a objetos
- Distribuido
- Robusto
- Seguro
- Neutral respecto a la arquitectura
- Portable
- Interpretado
- Alto rendimiento
- Multitenhebrado
- Dinámico

	Java	SmallTalk	TCL	Perl	Shells	C	C++
Sencillo	■	■	■	■	■	■	■
Orientado a Objetos	■	■	■	■	■	■	■
Robusto	■	■	■	■	■	■	■
Seguro	■	■	■	■	■	■	■
Interpretado	■	■	■	■	■	■	■
Dinámico	■	■	■	■	■	■	■
Portable	■	■	■	■	■	■	■
Neutral	■	■	■	■	■	■	■
Hebras	■	■	■	■	■	■	■
Recolección de basura	■	■	■	■	■	■	■
Excepciones	■	■	■	■	■	■	■
"Performance"	Alta	Media	Baja	Media	Baja	Alta	Alta

1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. Características y evolución.

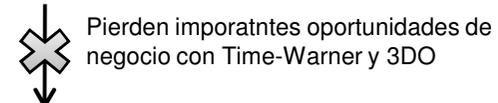


1970 - Bill Joy pretende diseñar un lenguaje combinando C y MESA
 1980 - Bill Joy pretende rehacer UNIX basándolo en objetos
 (Bill Joy fundó, junto a otras 3 personas, SUN Microsystems. Es vicepresidente)

1982: se crea SUN, que fabrica estaciones de trabajo con un UNIX propietario (Solaris)



1991 "Green Project" (secreto, 18 meses) / 1992 "FirstPerson Project" (abierto)
 SUN estudia su mercado y prevé un nicho importante: pretende introducir en el mercado un sistema Home Multimedia con Set Top Box (servicios externos: pay-per-view, etc) en red y con mando único. El software es un completo sistema operativo incluyendo elementos de desarrollo y su propio lenguaje "Oak" (a cargo de James Gosling).



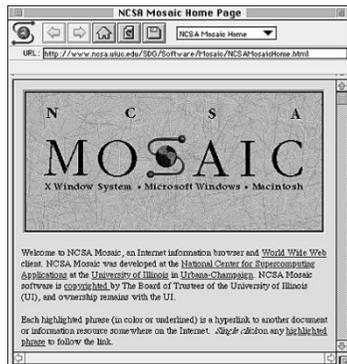
Pierden importantes oportunidades de negocio con Time-Warner y 3DO

SUN pretende introducir en el mercado aplicaciones desarrolladas con los recursos del proyecto FirstPerson



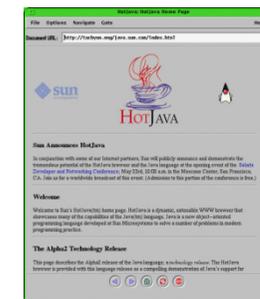
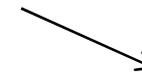
Disolución del equipo.

1992: Aparece el primer navegador www de gran difusión "Mosaic" (la idea original de http/html es de 1980 (Tim Berners Lee))



1994: Se abandona el objetivo de mercado anterior y se mira hacia Internet. John McNealy, en un fin de semana escribe un navegador semejante a Mosaic y todo el sistema del proyecto FirstPerson encaja a la perfección con la WWW

WebRunner



23 de mayo de 1995: se anuncia JAVA en la feria SunWorld

1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. **Características y evolución.**

- ***Simple***

- Es fácil de programar. Muy próximo a C/C++ sintácticamente pero omite elementos difíciles de entender o que propician los errores (no hay ficheros cabecera, ni aritmética de apuntadores, ni estructuras, ni uniones, ni sobrecarga de operadores, etc). Además el núcleo del lenguaje es pequeño.

- ***Orientado a objetos***

- Básicamente similar a C++ pero eliminando la herencia múltiple en favor de una mejor solución. Además se lleva a cabo chequeo fuerte de tipos.

1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. *Características y evolución.*

- ***Distribuido***

- Dispone de una extensa librería para tratar con protocolos TCP/IP (ftp, http, etc) y otros (UDP, etc.) Se utilizan objetos remotos con la misma facilidad que los locales y la utilización de "sockets" o la programación de CGI's es sencilla.

- ***Robusto***

- La robustez es un aspecto primordial en Java. Se revela en varios aspectos: eliminación del lenguaje de características propiciatorias de errores, máximo nivel de chequeos en tiempo de compilación, chequeos  de errores en tiempo de ejecución.

 Característica sólo posible mediante una Máquina Virtual, como es el caso de la JVM de la que se hablará más adelante

1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. **Características y evolución.**

- **Seguro**^{JVM}
 - Pensado para funcionar en entornos de red, se ha puesto el máximo énfasis en la seguridad. puede limitarse el acceso a zonas de memoria, no puede sobrepasarse el stack, puede limitarse el acceso a recursos locales, etc.
- **Neutral respecto a la arquitectura**^{JVM}
 - El compilador genera un código neutral, para toda máquina con la JVM (Java Virtual Machine). Este código es muy eficaz y puede traducirse "al vuelo" a código máquina con un compilador JIT (Just In Time).

^{JVM} Característica sólo posible mediante una Máquina Virtual, como es el caso de la JVM de la que se hablará más adelante

1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. *Características y evolución.*

- ***Portable***

- No hay características dependientes de la implantación (p.ej. Los enteros son siempre de 32 bits), el almacenamiento es siempre igual (no hay big/little endian), etc. Las librerías del sistema definen objetos portables (p.ej. “window”).

- ***Interpretado***

- El código puede ser interpretado, lo que hace el desarrollo más sencillo, ya que el “linkado” se simplifica.

1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. **Características y evolución.**

- ***Alto rendimiento***

- El hecho de interpretar un código intermedio supone una rebaja en la potencia computacional restante al proceso, pero el uso de un compilador JIT genera procesos de eficacia muy próxima a la del código "nativo". En este sentido se han ido consiguiendo importantes avances con el tiempo, de modo que las estrategias de compilación "al vuelo" y de optimización de código [p.ej. mediante "profiling"] dan lugar en muchas ocasiones a código más eficaz que el obtenido con compiladores a código nativo.

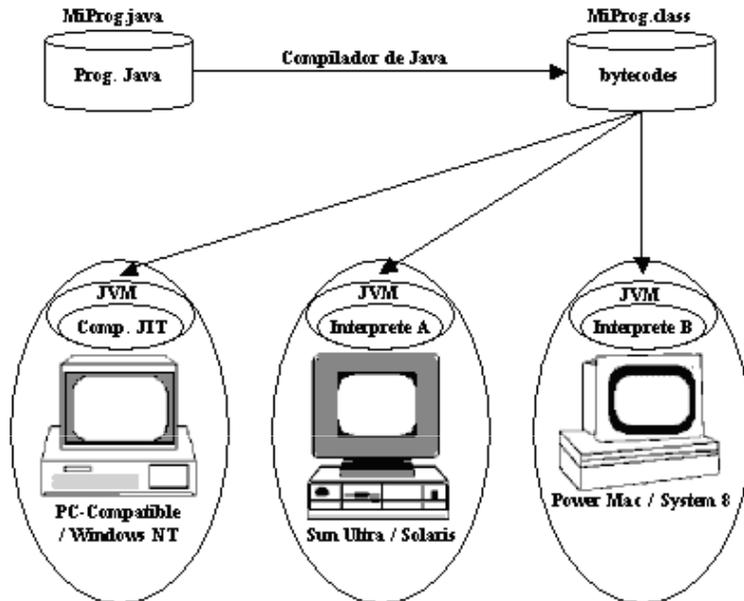
 Característica sólo posible mediante una Máquina Virtual, como es el caso de la JVM de la que se hablará más adelante

1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. *Características y evolución.*

- **Multitenhebrado** 
 - Los beneficios de esta característica son una mejor respuesta interactiva y mejor comportamiento en tiempo real.
- **Dinámico**
 - Diseñado para adaptarse a un entorno en evolución: pueden añadirse nuevos elementos a las librerías sin efecto en sus clientes, la capacidad de introspección de los programas permite acciones dinámicas, etc.

1. La situación de Java frente a otros lenguajes desde el punto de vista de su aplicación en matemáticas. Características y evolución.

"Write Once, Run Anywhere"



La máquina virtual.

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.
- * NSIScom 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 es novedoso el enfoque de emulador de máquina.
- 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