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

Del concepto de calculo efectivo a una

Clasificación general de lenguajes

Euclides (Método axiomático), Aristóteles(Lógica formal), Muhammad ibn Musa Al'Khowarizmi (Algorítmo)…

1928 David Hilbert (Frege Russel Whitehead)

1931 Kurt Gödel

1938

Máquina de Turing

Alan Turing

Cálculo Lambda

Alonzo Church

Sin olvidar las funciones recursivas de Herbrand-Gödel"

Lenguajes

Imperativos
- Fortran
- Cobol
- Pascal
- C
- …

O/B objetos
- Object Pascal
- C++
- Javascript
- Java
- Python
- …

Frameworks
- Ruby on Rails
- …

Funcionales
- Lisp
- Scheme
- ML
- Hope
- CLOS
- Ocaml
- …
- Haskell
- Clojure

Lógicos
- Prolog
- …

- Javascript
- Scala
- Java 8

! no hay asignación, (todo es inmutable) !

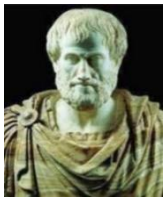
```
let rec long = function
                |[] -> 0
                |x::xs -> 1 + long xs;;

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

Ejemplo OCaml

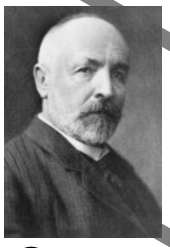
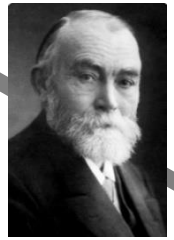
Prolog (1970 - 1997 (v.4))
es un lenguaje lógico-declarativo,
pero esa es "otra historia"

La formalización del conocimiento
El descubrimieto de sus límites
Y el Cálculo Efectivo




Aristóteles (384ac-322ac)
Euclides (325ac-265ac)

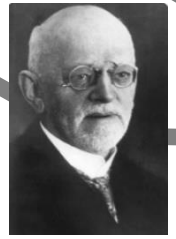
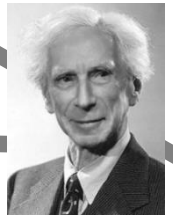
al-Khwārizmī
(780-850)

Frege (1848-1925)

Whitehead
(1861-1947)

Cantor
(1845-
1918)

Russell (1872-1970)

Zermelo
(1871-1953)

Fraenkel
(1891-1965)

Dedekind
(1831-1916)

Hilbert
(1862-1943)

Gödel (1906-1978)

Peano
(1858-1932)

Church (1903-1995)

Turing
(1912-1954)

Noether
(1882-1935)

Herbrand
(1908-1931)

Kleene
(1909-1994)

Mantengo esta página que tenía sentido cuando Alan Turing era desconocido

Alan **Turing** - Wikipedia, la enciclopedia libre

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 ...

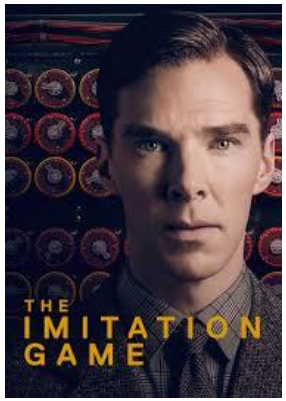
Máquina de Turing
Una máquina de Turing es un dispositivo que manipula ...

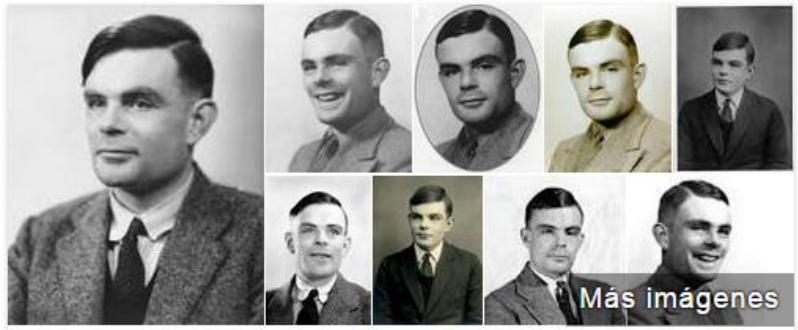
Tesis de Church-Turing
En teoría de la computabilidad, la tesis de Church-Turing formula ...

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

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

Más resultados de wikipedia.org »





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

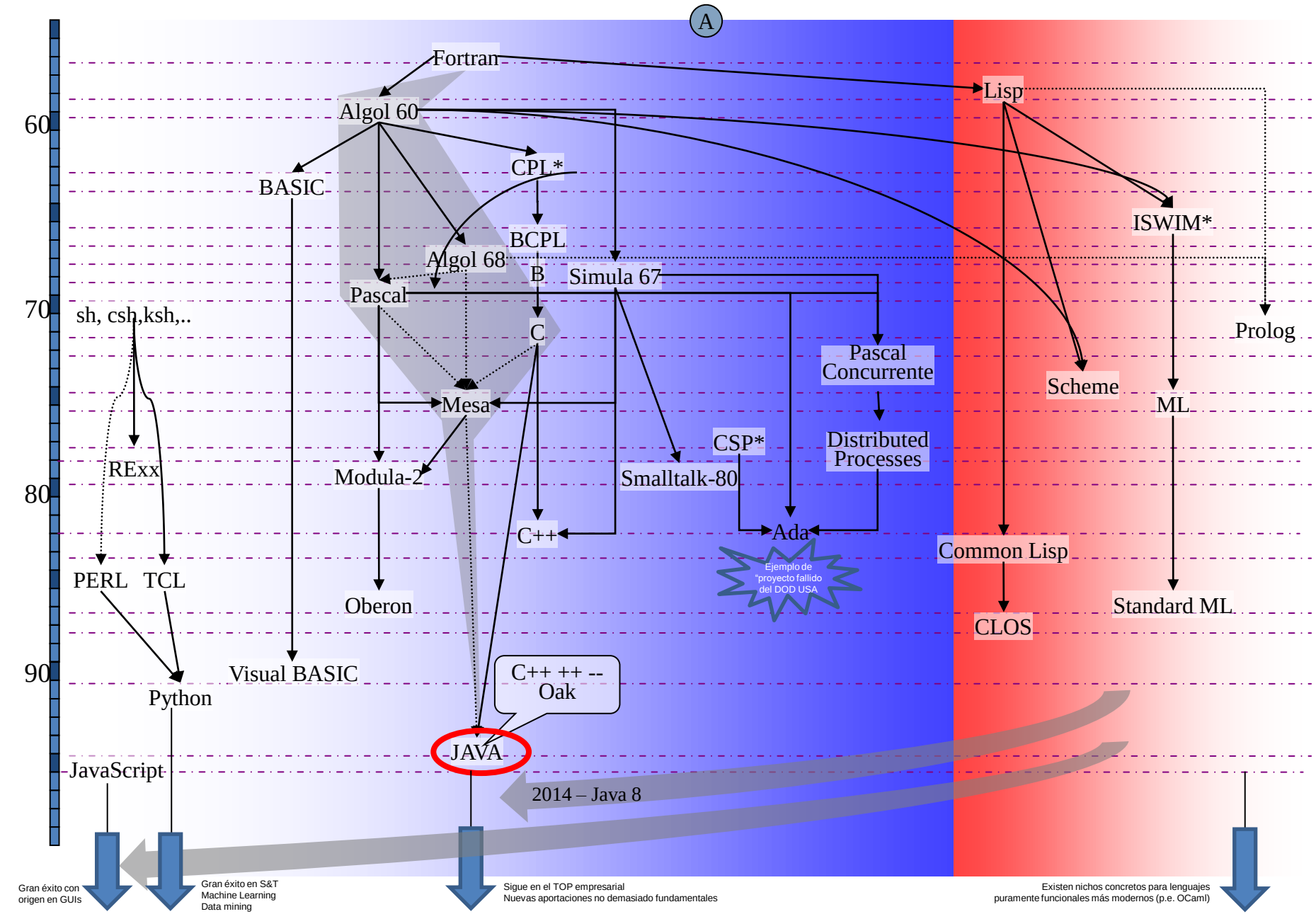
Fecha de nacimiento: 23 de junio de 1912, Maida Vale, Londres, Reino Unido

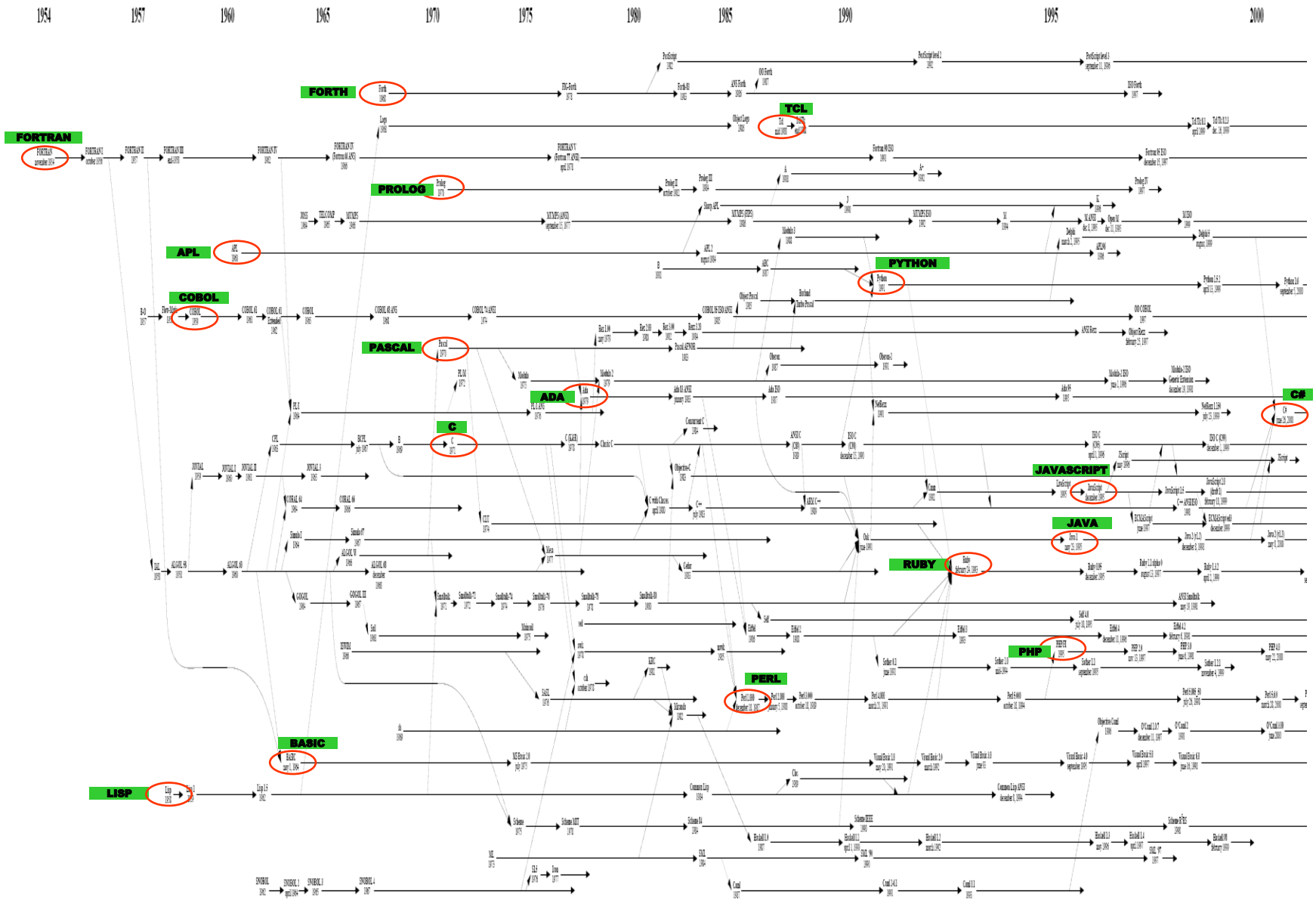
Fecha de la muerte: 7 de junio de 1954, Wilmslow, Reino Unido

Libros: Computing machinery and intelligence, ¿Puede pensar una máquina?

Padres: Julius Mathison Turing, Ethel Sara Stoney

Hermanos: John Turing

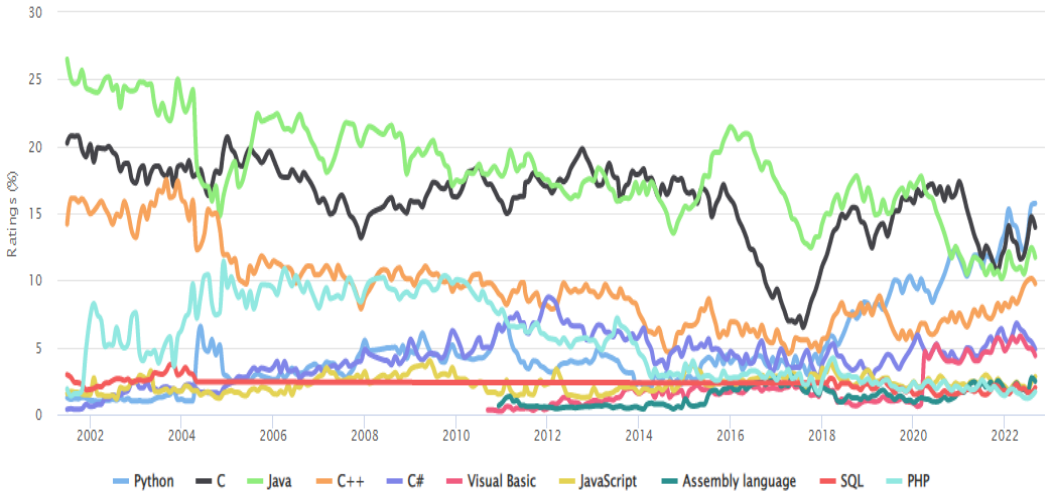




La página anterior es un extracto de este esquema de Éric Lévénez que se encuentra en http://www.levenez.com/lang/
Muestra las relaciones entre 50 lenguajes (con sus versiones). En la misma página hay referencias a compilaciones mucho más extensas (hasta 8945 lenguajes)

Ordenación de lenguajes por presencia en la web

Sep 2022	Sep 2021	Change	Programming Language	Ratings	Change
1	2	↑	Python	15.74%	+4.07%
2	1	↓	C	13.96%	+2.13%
3	3		Java	11.72%	+0.60%
4	4		C++	9.76%	+2.63%
5	5		C#	4.88%	-0.89%
6	6		Visual Basic	4.39%	-0.22%
7	7		JavaScript	2.82%	+0.27%
8	8		Assembly language	2.49%	+0.07%
9	10	↑	SQL	2.01%	+0.21%
10	9	↓	PHP	1.68%	-0.17%
11	24	⇈	Objective-C	1.49%	+0.86%
12	14	↑	Go	1.16%	+0.03%
13	20	⇈	Delphi/Object Pascal	1.09%	+0.32%
14	16	↑	MATLAB	1.06%	+0.04%
15	17	↑	Fortran	1.03%	+0.02%
16	15	↓	Swift	0.98%	-0.09%
17	11	⇊	Classic Visual Basic	0.98%	-0.55%
18	18		R	0.95%	-0.02%
19	19		Perl	0.72%	-0.06%
20	13	⇊	Ruby	0.66%	-0.62%



Position	Programming Language	Ratings	Position	Programming Language	Ratings
21	Julia	0.61%	36	PowerShell	0.21%
22	Scratch	0.57%	37	ABAP	0.21%
23	SAS	0.56%	38	Awk	0.20%
24	(Visual) FoxPro	0.52%	39	LabVIEW	0.20%
25	COBOL	0.52%	40	TypeScript	0.20%
26	Rust	0.51%	41	Groovy	0.18%
27	Prolog	0.45%	42	Erlang	0.15%
28	Ada	0.43%	43	Haskell	0.15%
29	Lua	0.41%	44	cg	0.15%
30	Lisp	0.37%	45	Transact-SQL	0.15%
31	PL/SQL	0.36%	46	Bash	0.14%
32	Dart	0.34%	47	Clojure	0.14%
33	Scala	0.33%	48	Apex	0.14%
34	Kotlin	0.31%	49	ActionScript	0.13%
35	D	0.27%	50	Elixir	0.13%

https://www.tiobe.com/tiobe-index/

51 a 100

ABC, Algol, Alice, AspectJ, B4X, Bourne shell, C shell, CFML, CHILL, CL (OS/400), Clipper, CLIPS, Crystal, EXEC, Icon, IDL, J#, Ladder Logic, Lasso, Logo, ML, MOO, MQL5, NATURAL, Nim, NXT-G, OCaml, Occam, OpenCL, PL/I, Q, Racket, Raku, REXX, Ring, RPG, Scheme, Simulink, Slate, Solidity, SPARK, SPSS, Stata, Tcl, Vala/Genie, VBScript, Verilog, VHDL, X++, Xojo

¿CÓMO DEBE SER UN LENGUAJE DE PROGRAMACIÓN?... PREMISAS DE JAVA

El origen (demasido remoto) de Java

Sun Microsystems

Logo used from the 1990s until acquisition by Oracle

Type	Public
Traded as	Nasdaq: SUNW Nasdaq: JAVA
Industry	Information technology
Founded	February 24, 1982; 39 years ago
Founders	Scott McNealy Vinod Khosla Andy Bechtolsheim Bill Joy
Defunct	January 27, 2010; 11 years ago
Fate	Acquired by Oracle
Headquarters	Menlo Park, California, U.S.
Products	Servers Workstations Storage Services
Owner	Oracle Corporation
Number of employees	38,600 (near peak, 2006)[1]
Website	www.sun.com See Archived 4 January 2010 at the Wayback Machine.

1970 Bill Joy vicepresidente de Sun Microsystems pretende diseñar un lenguaje combinando las mejores característicasde C y MESA.
1980 Bill Joy intenta reescribir UNIX concluyendo que C++ era inadecuado.

El auténtico origen

1991 (enero)
Stealth Project → Green project.
Objetivo: un sistema operativo para electrónica de consumo "inteligente" funcionando en red y controlada y programada con un "handheld".

- Bill Joy, dirección
- **James Gosling,** lenguaje de programación (C++ ++ --, Oak)
- Mike Sheradin, desarrollo de negocio
- Patrick McNaughton, Entorno gráfico
- Ed Frank (abril), hardware

James Gosling

James Gosling en 2005

Información personal	
Nombre de nacimiento	James Arthur Gosling
Nacimiento	19 de mayo de 1955 (66 años) Calgary, Alberta, Canadá
Residencia	Área de la Bahía de San Francisco
Nacionalidad	Canadiense
Familia	
Hijos	Kate y Kelsey
Educación	
Educado en	Universidad Carnegie Mellon Universidad de Calgary
Supervisor doctoral	Bob Sproull y Raj Reddy
Información profesional	
Ocupación	Científico de la computación
Conocido por	Creador del lenguaje de programación Java.
Empleador	Sun Microsystems (1984-2010) Google (2010- 2011) Liquid Robotics (2011 - 2017) Amazon Web Services (2017 - presente)
Obras notables	*Java*
Miembro de	Association for Computing Machinery
Distinciones	Oficial de la Orden de Canadá (2007)

CARACTERÍSTICAS

JAVA: características por diseño

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

La mayoría de estas características son incuestionables hoy en día, pero su planteamiento a principios de los 90 fue un hito.

(veremos que si alguna se cuestiona, no lo es de un modo fundamental)

Simple

- Fácil de programar: tan próximo al C++ como es posible, omitiendo elementos difíciles de entender o que propician los errores (no hay ficheros cabecera, aritmética de apuntadores, estructuras, uniones, sobrecarga de operadores, etc.)
- Pequeño (intérprete básico 40Kb, con librerias básicas y hebras 175Kb).

Orientado a objetos

- Básicamente similar a C++
  - Se elimina la herencia múltiple mediante una mejor solución.
  - Chequeo “fuerte” de tipos

Distribuido

- Disponer de una librería extensa para tratar con protocolos TCP/IP (ftp, http,...)
  - Se utilizan objetos remotos con la misma facilidad que los locales
  - La apertura 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

Seguro

- 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

- 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)

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)
- Las librerias del sistema definen objetos portables:
  - p.ej. “window”

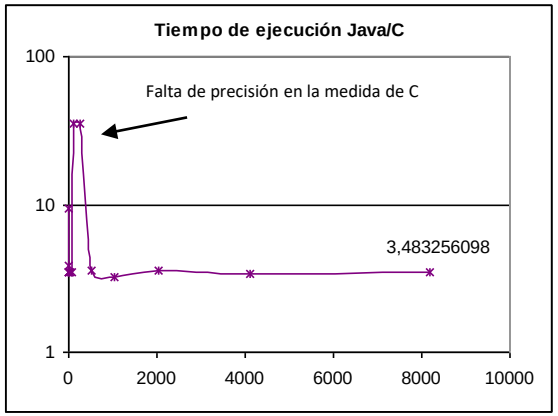
Sobre neutralidad respecto a la plataforma y portabilidad...

Una prueba de Java en cómputo intensivo

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

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**

Programar no es sólo escribir programas.

Interpretado

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

Alto rendimiento

- El hecho de interpretar un código intermedio supone una rebaja en la potencia computacional, pero el uso de un compilador JIT genera procesos de eficacia muy próxima a la del código “nativo”

Multienhebrado

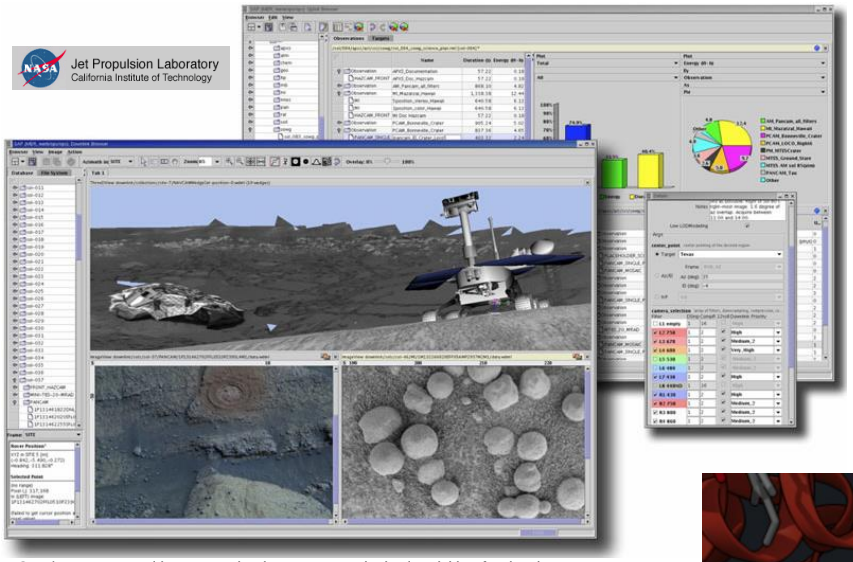
- Los beneficios de esta característica son una mejor respuesta interactiva y mejor comportamiento en tiempo real.

Dinámico

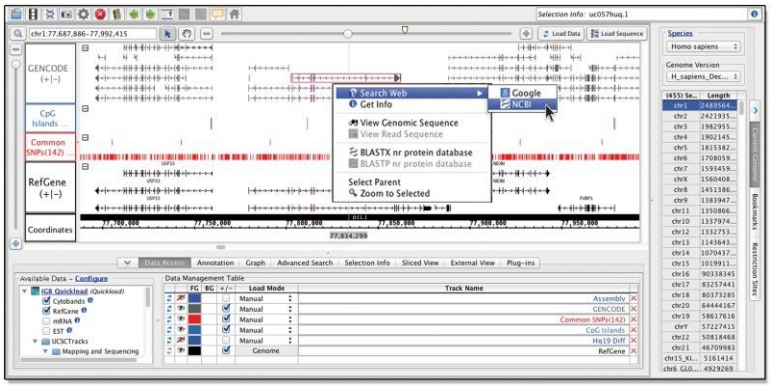
- Diseñado para adaptarse a un entorno en evolución.
  - Pueden añadirse nuevos elementos a las librerías sin efecto en sus clientes
  - La capacidad de introspección de los programas permite acciones dinámicas
  - etc.

ADENDUM:
¿SE USA JAVA EN CIENCIA Y TECNOLOGÍA? ¿ES INEFICIENTE?

Si bien Python, tiene una gran presencia en C&T (gracias inicialmente al éxito de Google, hoy en día domina la escena de la Minería de Datos y el Aprendizaje Automático), Java es utilizado en muchos otros campos dentro de la C&T.



SAP/Maestro, used by MER scientists to create desired activities for the day.



Integrated genome browser. Bioinformatics. 2016 Jul 15; 32(14): 2089–2095.

¿Java para cómputo intensivo?

Esta página es muy antigua y tenía el siguiente comentario:

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

El tiempo lo ha confirmado.

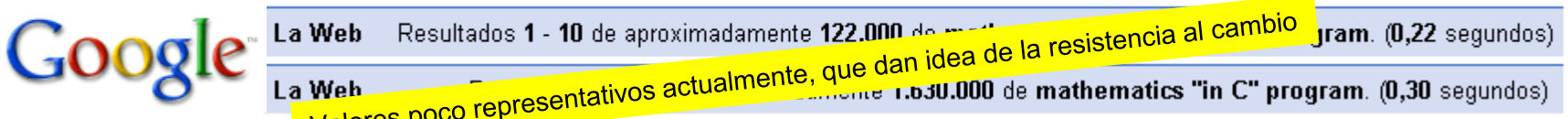
(CYT=matemáticas, física, ingeniería…)
Tradicionalmente, en computación para CYT, se ha buscado la "velocidad".

Esta visión CYT=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 CYT frente a nuevos lenguajes (debería llevarles a programar directamente los microprocesadores en su lenguaje ensamblador, pero curiosamente no se daba esto).

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



Valores poco representativos actualmente, que dan idea de la resistencia al cambio

(comparación i… 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 "acusó" 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 ya muchos 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
- Los lenguajes sobre máquinas virtuales (Java en particular) serán los más rápidos en el futuro ante cómputos complejos (no para el caso de algoritmos muy "cerrados")

ADENDUM:
SUN MICROSYSTEMS… IN MEMORIAM

Sun Microsystems

(1982 -2009)

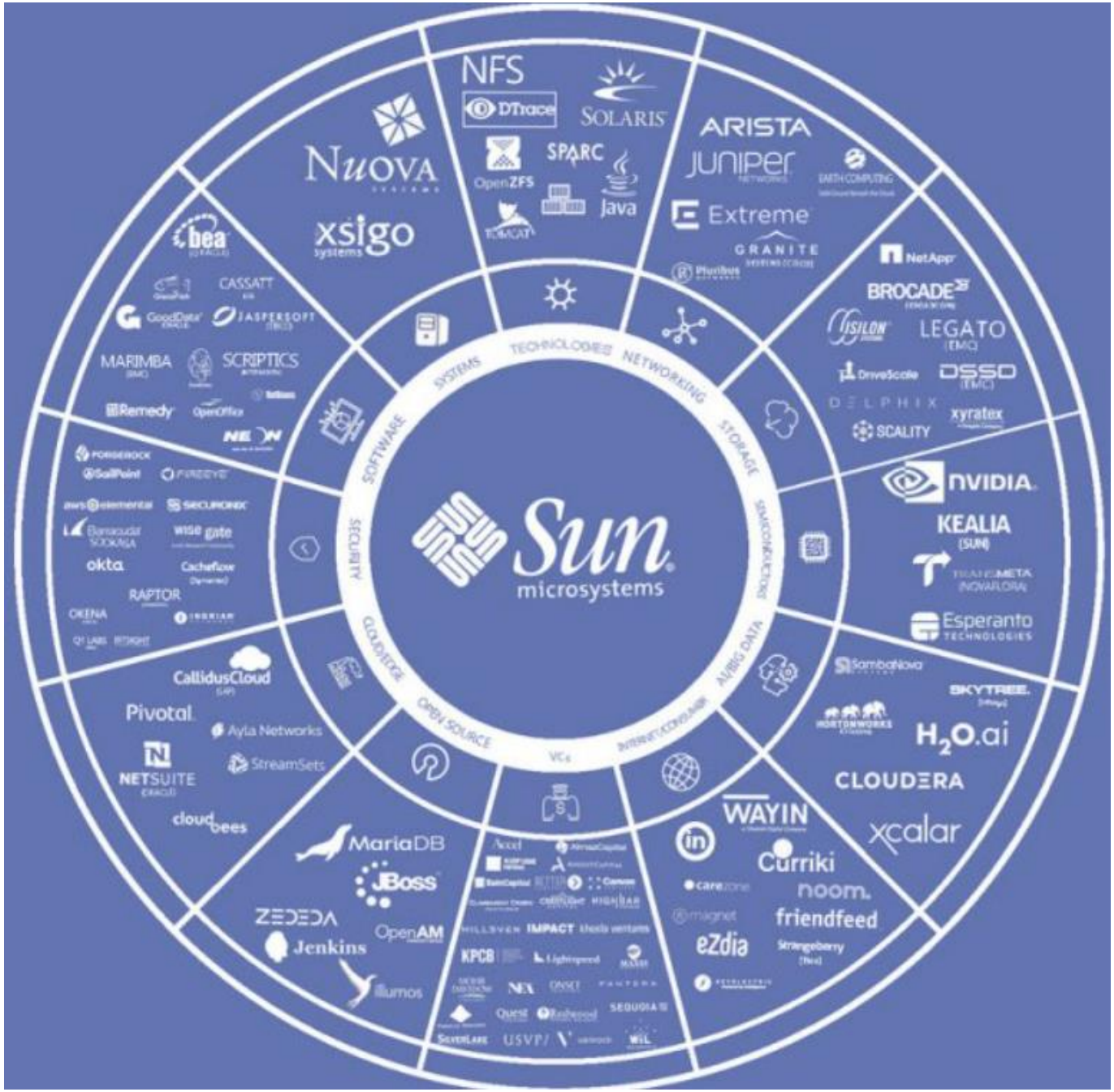
Fue comprada por Oracle en abril de 2009 por 7400 millones de dólares.

A principios del nuevo milenio la imagen de Sun era fantástica. Su larga experiencia en el desarrollo y uso de soluciones UNIX hizo que poco a poco fueran **contribuyendo más y más a proyectos Open Source de relevancia** o convirtió sus propios desarrollos a proyectos libres mediante la adopción de licencias Open Source.

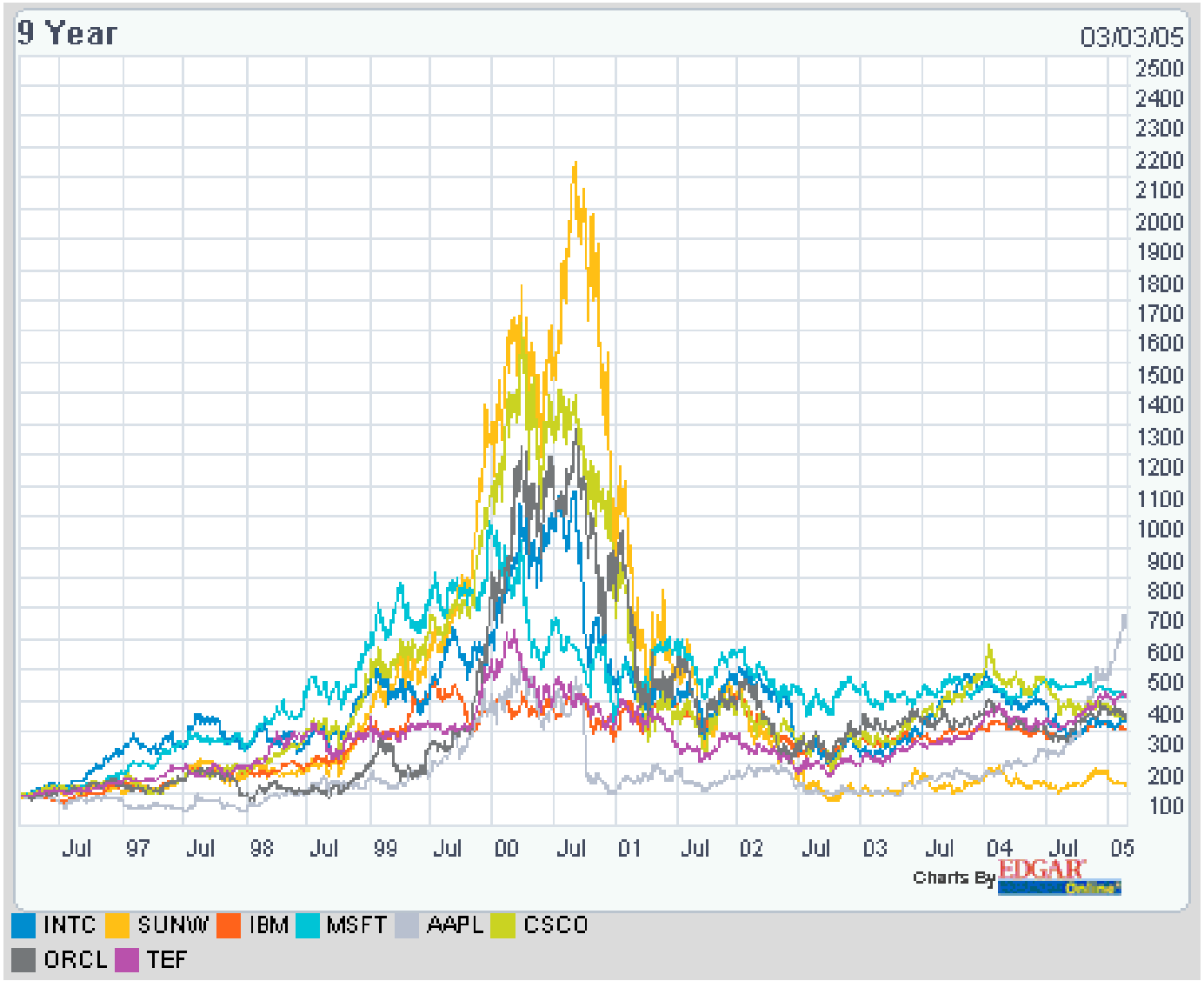
Java es el mejor ejemplo de ese amor al Open Source. Esta plataforma que revolucionó el mundo de la programación a principios de los 90 acabaría adoptando la licencia GPLv2 en noviembre de 2006, pero hay otros muchos ejemplos de esa filosofía.

Texto extraído de https://www.xataka.com/historia-tecnologica/asi-es-como-oracle-ha-sometido-a-sun-microsystems-a-una-muerte-larga-y-agonica

Q. E. P. D.



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

siliconnews.es
La actualidad empresarial y de las nuevas tecnologías

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.**

El (catastrófico) desarrollo del “Green Project”

1991 (agosto) *7 con interfaz de usuario gráfico mostrado a Joy y McNealy
1992 (septiembre) sistema completo mostrado a Joy y McNealy (sistema operativo, lenguaje, toolkit, interface, plataforma hardware en 18 meses)
1992 (septiembre) proyecto FirstPerson como reorientación tras el fracaso como negocio del anterior
1993 Concurso de Time-Warner para set-top-box de video bajo demanda concedido a SGI
1993 Negociación con 3DO para un S.O. Basado en Oak para su set-top-box abandonada.
1994 Cancelación de proyecto propio de televisión interactiva.
1994 FirstPerson reconducido a aplicaciones estandar en Oak.
1994 FirstPerson cancelado y disuelto

El nuevo origen de Java

1994 (junio) Bill Joy proyecto LiveOak “big small operating system”
1994 (julio) McNaughton escribe un navegador para internet con LiveOak

! Las características del lenguaje encajan con los requerimientos de Internet !

1994 (septiembre) Naughton y Jonathan Payne desarrollan WebRunner (luego HotJava)
1994 (octubre) HotJava es estable y se muestra a los ejecutivos de Sun que deciden soportar Java en relación con la WWW.
1995 Sun anuncia el Java en la SunWorld ´95; Netscape anuncia inmediatamente que lo soportará en su navegador. Algo más tarde Microsoft hace lo mismo.