

Principios de Lenguajes de Programación

Unidad 1

Muy Breve Introducción a la **Historia y Evolución** de los Lenguajes de Programación



Área Fundamentos Teóricos
Departamento de Teoría de la Computación
Facultad de Informática
Universidad Nacional del Comahue



Indice


- Historia temprana de los LdP
- Fortran, Algol, otros lenguajes
- Los 70s: simplicidad y abstracción
- Los 80s: modularidad y objetos
- Los 90s: tecnologías maduras



Historia temprana de los LdP


- Programas como Algoritmo
 - Mucho más anterior
 - 1840: Ada Lovelace “programó” la máquina analítica de Babbage
 - 600 AC: matemáticas se expresaban “programando”: descripción de “cómo” computar

PROGRAMMING LANGUAGES ENABLE USERS TO WRITE PROGRAMS FOR SPECIFIC COMPUTATIONS/ALGORITHMS



1843:
Ada Lovelace credited with first computer programming language; wrote an algorithm for the Analytical Engine (early mechanical computer)

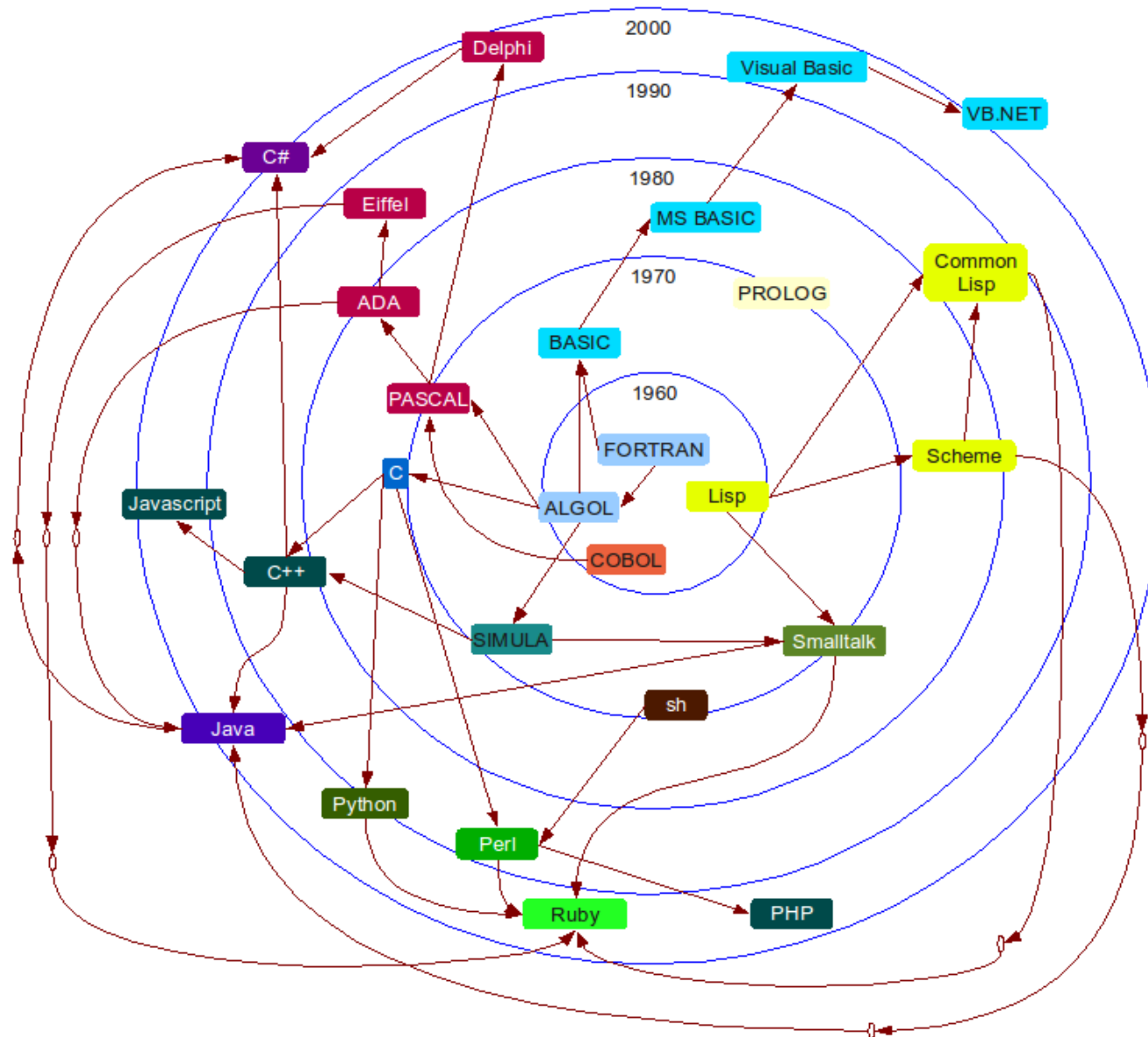
THERE ARE 1.2M+ COMPUTER PROGRAMMERS AND SOFTWARE DEVELOPERS IN THE US



~12,500



Historia: árbol genealógico de los LdP



FORTRAN

- John Backus
 - Líder del grupo en IBM que desarrolló el lenguaje y su compilador
- Herramienta para el cálculo científico
 - (FORMula TRANslation)
- Objetivo Principal: Eficiencia
 - Sintaxis limitada por el entorno
 - Implementación limitada por eficiencia
- En uso aún hoy en día
 - Fortran I, II, III, IV, Fortran66, Fortran77, Fortran90, Fortran95, Fortran 2003, Fortran2008



FORTRAN

- Implementa efectivamente características de los lenguajes de Alto Nivel
 - Variables
 - Bucles / Ciclos
 - Procedimientos
 - Etiquetas y Saltos, etc.
- Versiones antiguas: características únicas que se mantuvieron por compatibilidad
- Muy usado en aplicaciones de ingeniería que requieren mucho cálculo y manipulación de arreglos y matrices



ALGOL

- Primer lenguaje de programación con
 - Estructuras de bloque
 - Recursión
 - Definición *formal*
- No está en uso actualmente
- Es ancestro de la mayoría de los lenguajes actuales
- Diseño innovador:
 - Frente al resto de sus congéneres
 - La innovación más importante en la historia de los lenguajes de programación



Otros lenguajes de la época

- Algol60 – 1958 a 1960 (Backus)
 - “ALGOritmic Language”
 - Parecido a C pero distinto en arreglos
 - “Vive” en C, Java, C++, etc.
- LISP – 1958 a 1960 (McCarthy, Stanford)
 - Procesamiento de Listas
 - En Scheme, Common Lisp, etc.
- COBOL: 1959 a 1960 (Grace Hopper)
 - Orientado a los negocios
 - Sintaxis no matemática



Otros lenguajes de la época...

- PL/I: 1963 a 1964
 - El lenguaje universal
 - Con intencionalidad, pero el lenguaje es muy grande, adelantado a su tiempo, FRACASO
- Algol68: triunfo teórico
 - Muy elegante, ortogonal
 - Muy difícil de implementar
 - Descripción formal muy compleja, difícil de entender
 - DESASTRE práctico
 - Un lenguaje muy interesante



Otros lenguajes de la época...

- BASIC 1964
 - Permite expresar ideas elementales en las computadoras más simples
 - Muy fácil de aprender/propósito general
 - Limitado, pero el “primer” lenguaje de muchos
 - Presente en versiones actuales
- Simula67: primer lenguaje orientado a objetos
 - Muy adelantado a su tiempo
 - Uso para simulación y programación concurrente
 - Específico e ineficiente



Los 70s: simplicidad y abstracción

- C (Ritchie, 1972), Pascal (Wirth, 1971)
 - Más sencillos que los lenguajes de los 60s
 - Añade estructuras de datos (como COBOL)
 - Estructuras de control de Algol60/68
 - Éxito universitario
- Otros lenguajes TDA
 - CLU, Euclid, Mesa (1974-1979)
 - Formas de programar que descompone en unidades independientes
- Scheme (Sussman, Steele, 1975)
 - LISP reglado



Los 80s: modularidad y objetos

- Modula-2 (Wirth, 1982)
 - Continuación (no tan exitosa) de Pascal (fuera de tiempo)
 - Permite concurrencia
- Smalltalk80 (Alan Kay)
 - Avance en POO
- C++ (Stroustrup, 1980)
 - Extensión OO de lenguaje imperativo
 - Demuestra que OO puede ser eficiente
 - Sintaxis completa, semántica dificultosa



Los 80s ...

- ADA (Ichbiah, 1980)
 - Intento práctico de lenguaje universal
 - Más exitoso que PL/I
 - No existen dialectos: completamente estándar
 - Soporta concurrencia y fuertes verificaciones
 - Muy burocrático para los programadores
 - Muy bien “bancado”



Los 90s: tecnologías maduras

- Necesidad creciente de librerías
 - API (Application Programming Interface)
- JAVA (Gosling, 1995)
 - Actualización de C++
 - Primer lenguaje con API ya desarrollada
 - Estandar de Facto?
- C#
 - Muy utilizado, parte del entorno .NET
 - Basado en C++, Java y Delphi



Los 90s: tecnologías maduras

- Muchas implementaciones de programación de sistemas:
 - Perl, Tcl, Javascript, VBScript, Python, PHP
- Los lenguajes declarativos se hacen oír
 - ML (Milner, 1978-1988)
 - Haskell (Hudack, Peyton-Jones, Wadler, 1989-1998)



Los 2000s...

- Conviven lenguajes de programación históricos con modernos
- Híbridos y multiparadigmas
- De Propósito Específico
- Lenguajes de Marcas y metalenguajes
- Programación Visual
- ¿nuevos lenguajes?
- ¿nuevos paradigmas?



Diversidad

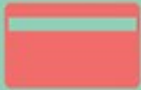


- Muchos lenguajes y muchas implementaciones

What is the BEST programming language?

To my knowledge there is no programming language called BEST



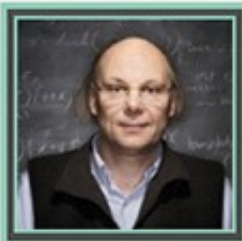




Lenguajes históricos en uso

1957-1959	<p>FORTRAN (FORMULA TRANSLATION), LISP (LIST PROCESSOR), AND COBOL (COMMON BUSINESS-ORIENTED LANGUAGE)</p> <p>Considered the oldest languages that are still used today. High-level languages created for scientific, mathematical, and business computing.</p>	<p>PRIMARY USES</p> <p>Supercomputing applications, AI development, business software</p>	<p>USED BY</p> <p>NASA, credit cards, ATMs</p> <p>NASA</p> 	<p>FUN FACT</p> <p>Action movie The Terminator used samples of Cobol source code for the text shown in the Terminator's vision display.</p>
1970	<p>PASCAL (AFTER FRENCH MATHEMATICIAN/PHYSICIST BLAISE PASCAL)</p> <p>High-level. For teaching structured programming and data structuring. Commercial versions widely used throughout the '80s.</p>	<p>CREATOR</p> <p>NIKLAUS WIRTH</p> 	<p>PRIMARY USES</p> <p>Teaching programming</p> <p>Object Pascal, a derivative, is commonly used for Windows application development</p>	<p>USED BY</p> <p>Apple Lisa (1983), Skype</p> 







Lenguajes históricos en uso

1972	<p>C (BASED ON AN EARLIER LANGUAGE CALLED "B")</p> <p>General-purpose, low-level. Created for Unix systems. Second most popular language (behind Java). Many leading languages are derivatives, including C#, Java, JavaScript, Perl, PHP, and Python.</p>	<p>CREATOR</p> <p>DENNIS RITCHIE Bell Labs</p> 	<p>PRIMARY USES</p> <p>Cross-platform programming, system programming, Unix programming, computer game development</p>	<p>USED BY</p> <p>Unix (rewritten in C in 1973), early WWW servers and clients</p> 
1983	<p>C++ (FORMERLY "C WITH CLASSES"; ++ IS THE INCREMENT OPERATOR IN "C")</p> <p>Intermediate-level, object-oriented. An extension of C, with enhancements such as classes, virtual functions, and templates.</p>	<p>CREATOR</p> <p>BJARNE STROUSTRUP Bell Labs</p> 	<p>PRIMARY USES</p> <p>Commercial application development, embedded software, server/client applications, video games</p>	<p>USED BY</p> <p>Adobe, Google Chrome, Mozilla Firefox, Microsoft Internet Explorer</p>  



Lenguajes históricos en uso

1983	<p>OBJECTIVE-C (OBJECT-ORIENTED EXTENSION OF "C")</p> <p>General-purpose, high-level. Expanded on C, adding message-passing functionality based on Smalltalk language.</p>	<p>CREATOR</p> <p>BRAD COX AND TOM LOVE Stepstone</p> 	<p>PRIMARY USES</p> <p>Apple programming</p>	<p>USED BY</p> <p>Apple's OS X and iOS operating systems</p> 
1987	<p>PERL ("PEARL" WAS ALREADY TAKEN)</p> <p>General-purpose, high-level. Created for report processing on Unix systems. Today it's known for high power and versatility.</p>	<p>CREATOR</p> <p>LARRY WALL Unisys</p> 	<p>PRIMARY USES</p> <p>CGI, database applications, system administration, network programming, graphics programming</p>	<p>USED BY</p> <p>IMDb, Amazon, Priceline, Ticketmaster</p> 









Lenguajes históricos en uso

1991	<p>PYTHON (FOR BRITISH COMEDY TROUPE MONTY PYTHON)</p> <p>General-purpose, high-level. Created to support a variety of programming styles and be fun to use. Tutorials, sample code, and instructions often contain Monty Python references.</p>	<p>CREATOR</p> <p>GUIDO VAN ROSSUM CWI</p> 	<p>PRIMARY USES</p> <p>Web applications, software development, information security</p>	<p>USED BY</p> <p>Google, Yahoo, Spotify</p> 
1993	<p>RUBY (THE BIRTHSTONE OF ONE OF THE CREATOR'S COLLABORATORS)</p> <p>General-purpose, high-level. A teaching language influenced by Perl, Ada, Lisp, Smalltalk, etc. Designed for productive and enjoyable programming.</p>	<p>CREATOR</p> <p>YUKIHIRO MATSUMOTO</p> 	<p>PRIMARY USES</p> <p>Web application development, Ruby on Rails</p>	<p>USED BY</p> <p>Twitter, Hulu, Groupon</p> 






Lenguajes históricos en uso

1995	<p>JAVA (FOR THE AMOUNT OF COFFEE CONSUMED WHILE DEVELOPING THE LANGUAGE)</p> <p>General-purpose, high-level. Made for an interactive TV project. Cross-platform functionality. Currently the world's most popular programming language.</p>	<p>CREATOR</p> <p>JAMES GOSLING Sun Microsystems</p> 	<p>PRIMARY USES</p> <p>Network programming, web application development, software development, Graphical User Interface development</p>	<p>USED BY</p> <p>Android OS/apps</p> 
1995	<p>PHP (FORMERLY "PERSONAL HOME PAGE," NOW IT STANDS FOR "HYPERTEXT PREPROCESSOR")</p> <p>Open-source, general-purpose. For building dynamic web pages. Most widely used open-source software by enterprises.</p>	<p>CREATOR</p> <p>RASMUS LERDORF</p> 	<p>PRIMARY USES</p> <p>Building/maintaining dynamic web pages, server-side development</p>	<p>USED BY</p> <p>Facebook, Wikipedia, Digg, WordPress, Joomla</p>   



Lenguajes históricos en uso

1995	<p>JAVASCRIPT (FINAL CHOICE AFTER "MOCHA" AND "LIVESCRIPT")</p> <p>High-level. Created to extend web page functionality. Used by dynamic web pages for form submission/validation, interactivity, animations, user activity tracking, etc.</p>	<p>CREATOR</p> <p>BRENDAN EICH Netscape</p> 	<p>PRIMARY USES</p> <p>Dynamic web development, PDF documents, web browsers, desktop widgets</p>	<p>USED BY</p> <p>Gmail, Adobe Photoshop, Mozilla Firefox</p>  
------	---	---	---	--

