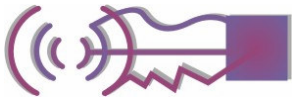
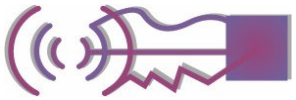


EVOLUCION HISTORICA DE LOS LENGUAJES DE PROGRAMACION

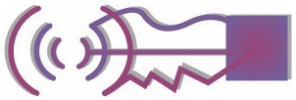
contribución de los lenguajes

motivación – herencia - caracteristicas



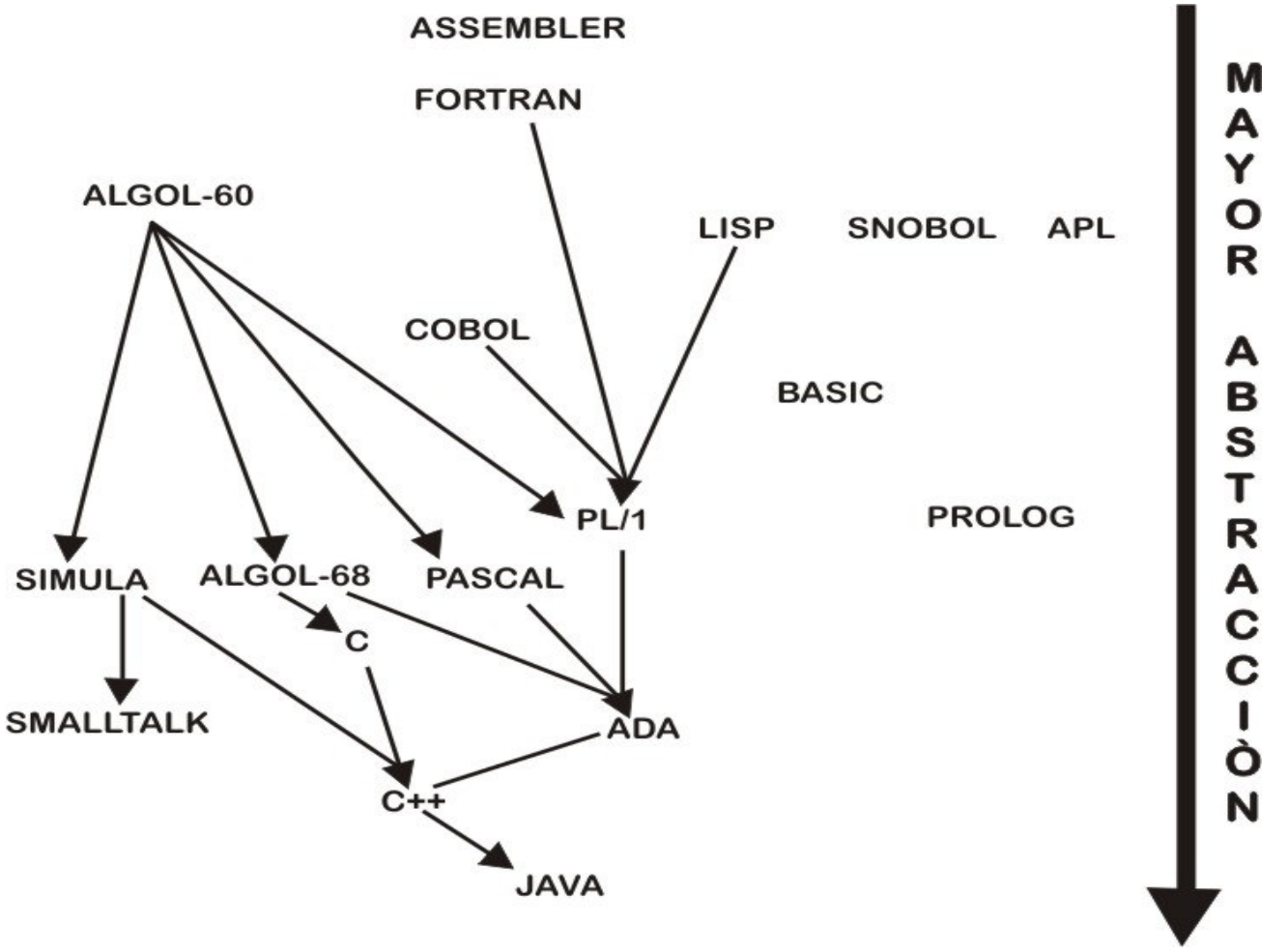
Puntos de vista

- **Programador**
- **Diseñador**
- **Implementador**



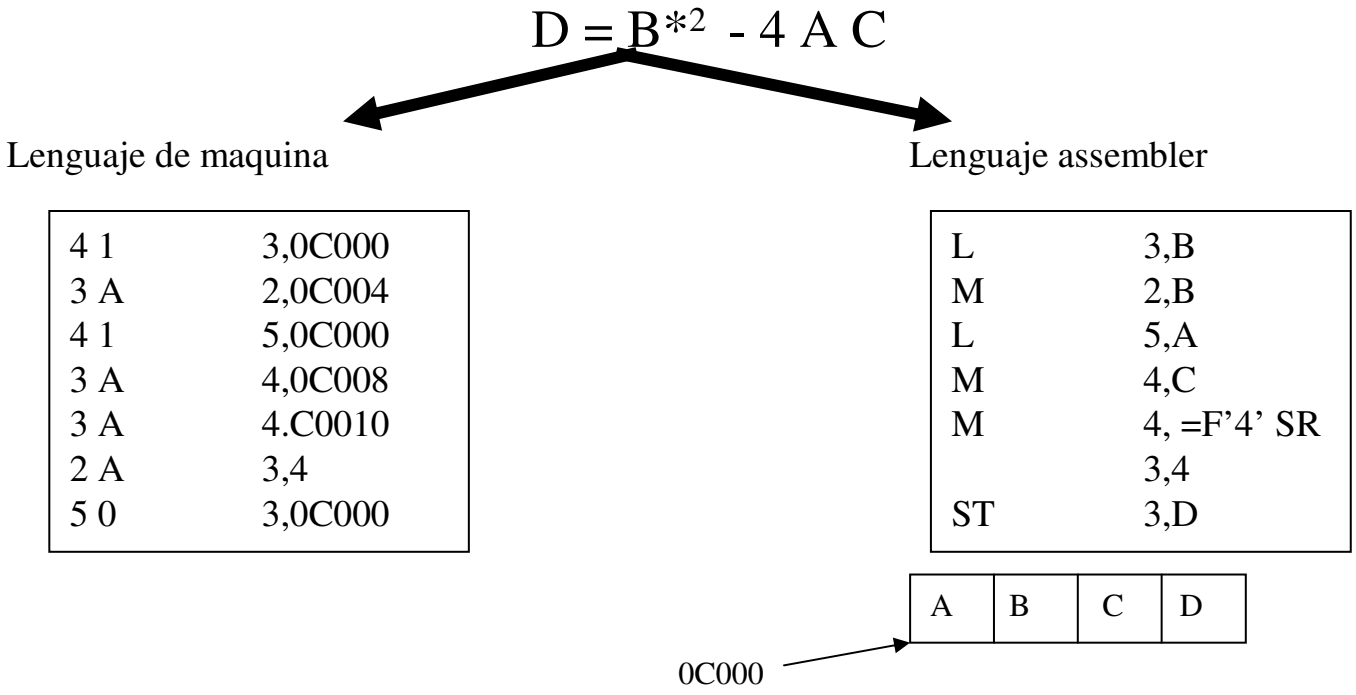
Usuario del lenguaje

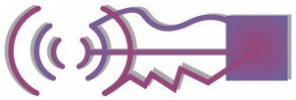
- **proceso de desarrollo de software**
- **comunicación de resultados**
- **mantenimiento y confiabilidad**


ASSEMBLER
FORTRAN
ALGOL-60
LISP
SNOBOL
APL
COBOL
BASIC
PL/1
PROLOG
SIMULA
ALGOL-68
PASCAL
C
SMALLTALK
ADA
C++
JAVA
MAYOR ABSTRACCIÒN

Assembler

Lenguaje ensamblador o simbólico

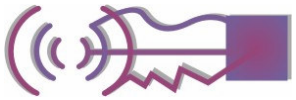




PRIMEROS PASOS

usuario – máquina → expresividad - eficiencia

- FORTRAN
- ALGOL-60
- COBOL



Fortran (54)

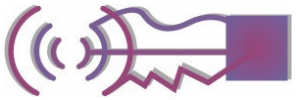
Backus - **For**mula **Trans**lantion

Primer lenguaje de alto nivel

- Objetivo: cálculos numéricos

- Notación de expresión
- Modularidad
- Ambiente global

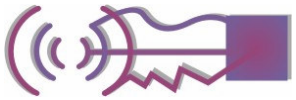
Fortan II–FortranIV(66)-Fortran77–Fortran90



Algol – 60 (60)

Comité europeo(Naum): **L**enguaje**Algo**ritmico

- Objetivo: independiente de la máquina sin E/S
- Estructura de bloques
- Recursión
- Estructura de datos: concepto de tipo
- Definido con notación BNF
- Compilador muy potente
- El lenguaje académico

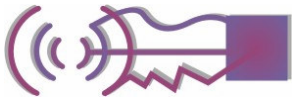


Cobol (59)

Comite - **Co**mmon **B**usiness **L**anguage

Objetivo: aplicaciones comerciales

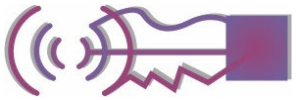
- descripción de programa en lenguaje natural.
- Archivos de datos
- Tipos tructurados
- Facilidades de impresión de informes.
- Conectores de nombres: NOM_APE,



LENGUAJES NO CONVENCIONALES

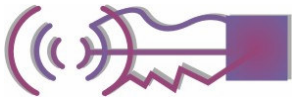
Fuertes consumidores de recursos, se alejan del hardware subyacente.

- LISP
- SNOBOL
- APL



Lisp (58)
Mc Carthy (IBM)- **Lis**t **P**rocesing

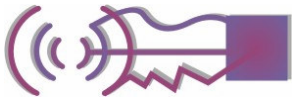
- Objetivo: procesamiento de listas de datos simbólicos
- Lenguaje funcional aplicado a la IA
- Uniformidad de código y datos
- Algoritmos recursivos que manipulan datos dinámicos
- Garbage collector



Snobol (64)

Laboratorios Bell - **S**tri**n**g **O**riented Sim**bo**lic **L**anguajes

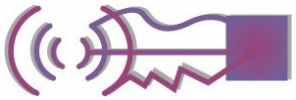
- Objetivo: manipulación de Strings
- Procesamiento de texto en lenguaje natural
- Pattern matching: reconocimiento de patrones



APL (60)

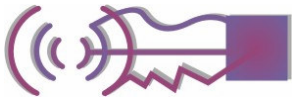
Iverson (IBM) - **A** **P**rogramming **L**anguage

- Objetivo: manejo de arreglos
- Características dinámicas
- Interactivo
- Encriptado: símbolos especiales
- Libera al usuario de tratar elemento a elemento



INTEGRACION – **PL/1** (64)
IBM – **P**rogrammin **L**anguaje **1**

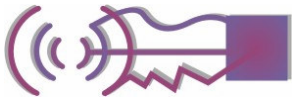
- Objetivo: integrar todos los conceptos de los lenguajes existentes a la fecha:
  - Fortran
  - ALGOL-60
  - Cobol
  - LISP
- manejo de excepciones
- punteros
- facilidades de multitasking



SIGUIENTE PASO

- SIMULA
- ALGOL-68
- PASCAL

- BASIC



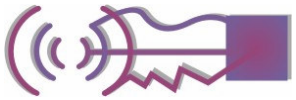
Simula (67)

Iverson noruego-**Simu**lation **La**nguage

- Objetivo: simulación
- Corrutinas: ejecución paralela
- Constructor **class**:

ABSTRACCION

  - modularización
  - herencia - jerarquía

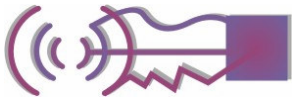


Algol 68
Comité

- Objetivo: principio de **ortogonalidad**

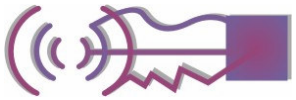
A partir de elementos básicos componer libremente con efectos predecibles

- Reporte: especificacion formal del lenguaje
- Falta de compromiso con el mundo
- Utilizado en las universidades



Pascal (71)
Wirth – x Blaise Pascal

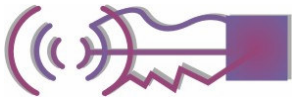
- Objetivo: enseñanza de la programación
- Ideal para implementar los principios de la programación estructurada
- Simple y expresivo
- Permite programar disciplinadamente
- Versiones posteriores permiten la construcción de TADs y la modularización.



BASIC (64)

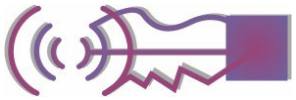
Beginners All purpouse Symbolic Instrucccion Code

- Objetivo: el tiempo del usuario es mas importante que el tiempo de máquina
- sintáxis algebraica
- pocas estructuras de datos
- simple fácil y eficientes implementaciones
- muy popular entre los usuarios de PCs y los principiantes.
- estilo imperativo y altamente interactivo



Evolución de estructuras de control

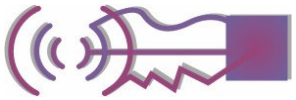
- FORTRAN II IF (4*X) 3,4,5
- FORTRAN IV IF (X.EQ.Y) 8
- ALGOL-60 *if* cond *then* accion1
- *else* accion2
- ALGOL-68 w := if x:=x+y, x>z then
  z:=z+1 else x:x+1, z fi



EXPERIMENTOS

Fines específicos

- C
- PROLOG

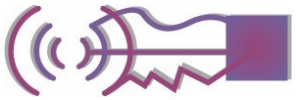


C (71)

Laboratorios Bell – Ritchie y Thompson

- Objetivo: implementar el sistema operativo UNIX
- rico en operadores
- rico en tipos de datos
- Transportable

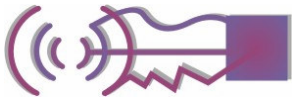
Demostró que era posible implementar un sistema operativo en alto nivel



Prolog (72)

Universidad de Marsella- Programación en Lógica

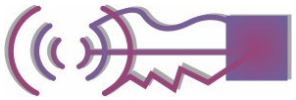
- Objetivo: para aplicaciones de IA. Estilo declarativo
- Calculo proposicional
- Objetos y relaciones de inferencia
- Intenta alejarse de los conceptos de la máquina de Von Newman



SISTEMAS Y PARADIGMAS

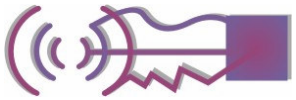
Premisa: construir sof confiable y mantenible

- SMALLTALK
- ADA
- C++
- JAVA



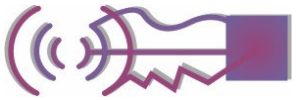
Smalltalk (71)
Xerox (Alan Kay)

- Objetivo: lenguaje orientado a objetos
- Lenguaje de programación y entorno de desarrollo
- Sistema: un conjunto de objetos que se comunican entre sí mediante mensajes
- Abstraccion de datos: objetos
- Herencia, Polimorfismo y binding dinámico



ADA (82)
Comité DoD – Ada Byron

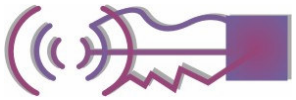
- Objetivo: integrar el estado del arte en lenguajes de programación. Abstracción a nivel de sistemas
- Unidades genéricas
- Tiempo real
- Concurrencia
- Abstracción de datos
- Manejo de excepciones
- Constructor principal: el package



C++ (86)

L. Bell (Stroustrup) – Superador de C

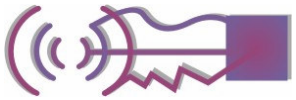
- Objetivo:
  - C + clases
  - manteniendo eficiencia
- Hibrido: soporta la programación procedural y la orientación a objetos
- Propósito general



Java (95)
Sun (J. Gosling)

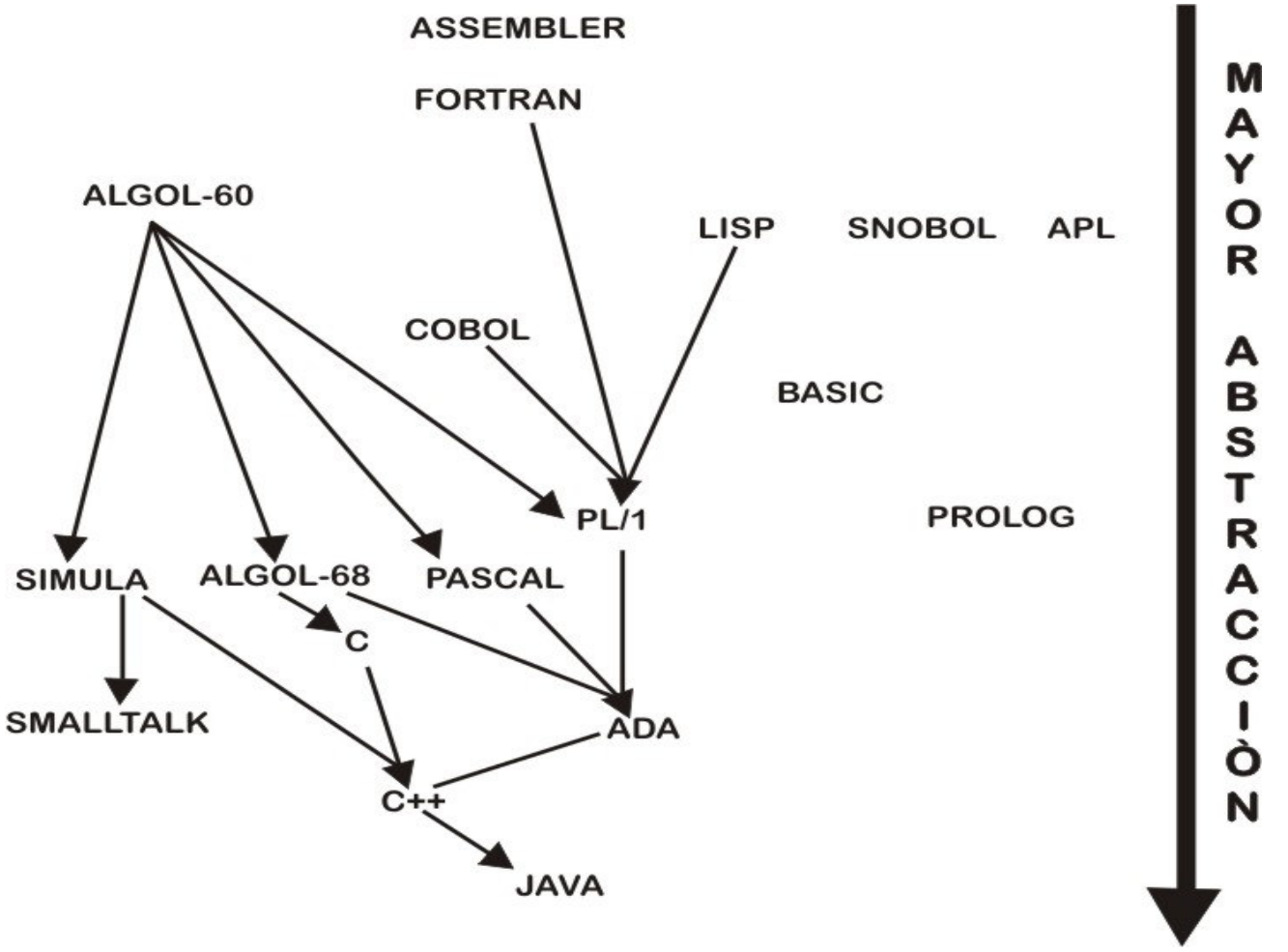
- Objetivo: para electrodomésticos: “no debe colgarse”
- Simple y OO
- Arquitectura neutral y portable: bytecode-movilidad
- Robusto
- Interpretado y dinámico
- Seguro: bien definido, sin punteros, fuertemente tipado, recolector de residuos

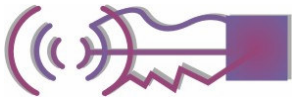
El lenguaje de las redes - (IBM-Oracle-Netscape)



Evolución de datos

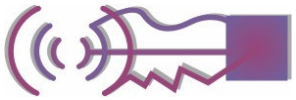
- FORTRAN Tipos predefinidos
- Algol-60 Tipos definidos por el usuario
- ModulaII,ADA,C++ TADs : class, módulo, paquete
- Smalltalk,C++, Java OBJETOS


ASSEMBLER
FORTRAN
ALGOL-60
LISP
SNOBOL
APL
COBOL
BASIC
PL/1
PROLOG
SIMULA
ALGOL-68
PASCAL
C
SMALLTALK
ADA
C++
JAVA
MAYOR ABSTRACCIÒN



EVOLUCION

- ESTANDARIZACION
- OPORTUNIDAD
- CONFORMIDAD
- OBSOLESCENCIA



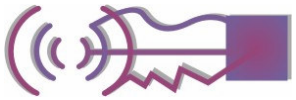
Estandarización

Sentencia valida en esta versión?

- consultar el manual
- probarla en la máquina
- | leer la definición estándar

- PATENTADOS
- POR CONSENSO

ANSI ISO IEEE



Oportunidad

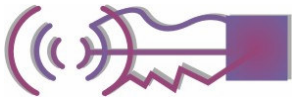
- Cuando estandarizar?

  Cuando su uso va en aumento y antes de que aparezcan muchas versiones

Conformidad

Adherir a un estándar

- compilador
- programa



Obsolescencia

- Cuando pierde vigencia un estándar
- Característica obsoleta
- Característica desaprobada