

*Home Page*

*Title Page*

*Contents*



*Page 1 of 25*

*Go Back*

*Full Screen*

*Close*

*Quit*

# Evolución de los Lenguajes de Programación: Una mirada desde la Investigación

Escuela de Ingeniería de Sistemas y Computación



Profesor: Juan Francisco Diaz  
[jdiaz@eisc.univalle.edu.co](mailto:jdiaz@eisc.univalle.edu.co)

20 de septiembre de 2002

Home Page

Title Page

Contents



Page 2 of 25

Go Back

Full Screen

Close

Quit

# Plan de la conferencia

- Un poco de historia
- La perspectiva de los paradigmas
- Retos de investigación
- CRISOL: Una apuesta a la investigación
- Referencias para estar al día

*Home Page*

*Title Page*

*Contents*



*Page 3 of 25*

*Go Back*

*Full Screen*

*Close*

*Quit*

# Un poco de historia

- Temas relacionados:
  - Lenguaje
  - Matemáticas
  - Formalismos de cálculo
- Lenguajes de Programación

Home Page

Title Page

Contents



Page 4 of 25

Go Back

Full Screen

Close

Quit

# Un poco de historia: Lenguaje (1)

- Lenguaje ↔ Comunicación
- Lenguaje de Programación ↔ Comunicación con una máquina
- Lenguaje hablado vs Lenguaje escrito
- Lenguaje escrito → Formalismos → Lenguajes formales

Home Page

Title Page

Contents



Page 5 of 25

Go Back

Full Screen

Close

Quit

## Un poco de historia: Lenguaje (2)

- Sistemas de escritura independientes del medio usado para representarla:
  - Sumerios, Egipcios (A.C.)
  - Mayas (D.C)
- Evolución de los alfabetos: Pictogramas → Ideogramas → Fonogramas → Consonantes → Cons. + Vocales

Home Page

Title Page

Contents



Page 6 of 25

Go Back

Full Screen

Close

Quit

# Un poco de historia: Matemáticas

- Algoritmos: Babilonios (base 60, A.C.), Griegos (Euclides, A.C.), Arabes (Al-Khorazmi, base 10, D.C.)
- Leibniz: Lenguajes formales
- Rusell: Lógica y teoría de conjuntos (Fundamentos de los Tipos)
- Hilbert: Construcción de pruebas ( $\rightarrow$  Construcción de programas)
- Tarski: Semántica de la Lógica
- Gödel: Incompletitud de la aritmética

Home Page

Title Page

Contents



Page 7 of 25

Go Back

Full Screen

Close

Quit

# Un poco de historia: Formalismos de cálculo

- Turing: Máquinas formales
- Church: Cálculo Lambda
- Zuse: Plan Calculus
- Goldstine, Von Neumann: Diagramas de flujo

Home Page

Title Page

Contents



Page 8 of 25

Go Back

Full Screen

Close

Quit

# Un poco de historia: Lenguajes de Programación (1)

- Primeras máquinas → Primeros lenguajes de programación, orientados por las máquinas:
  - Algebraic Interpreter: MIT
  - A-2: Univac
- Se advierten los primeros grandes requerimientos: Independencia de las máquinas, Alto Nivel

# Un poco de historia: Lenguajes de Programación (2)

- Los pioneros:

- Fortran (1953): Estructuras de control
- LISP (1958): Manejo automático de la memoria, Inteligencia Artificial, Recursión
- ALGOL (1960): BNF, Estructuras de bloques
- ALGOL-68: Computación paralela, tipos, coerción
- Simula (1960,67): Lenguaje con propósito específico (simulación de procesos), clases de objetos.
- Cobol (1968,74,85): Lenguaje con propósito específico (Procesamiento de grandes cantidades de datos)
- Otros: Apl,Snobol

Home Page

Title Page

Contents



Page 9 of 25

Go Back

Full Screen

Close

Quit

# Un poco de historia: Lenguajes de Programación (3)

## ■ La evolución:

- PL I (1960): reemplazar Fortran y Cobol, bloques, recursión, excepciones, facilidad de depuración. Prematuro para la época.
- Pascal (1971); Disciplina, enseñanza, refinamiento paso a paso, programación estructurada.
- C (1972): Competidor de Pascal, Cobol y Fortran. Se impone cuando el procesamiento gráfico y en red se hace indispensable.
- Prolog (1972): El poder de la deducción al servicio de la programación, IA, capacidad de paralelización, la apuesta japonesa.
- BASIC (1978): fácil de aprender, rápido de compilar, fácil de depurar.
- Scheme, ML, CAML, Haskell (1980's): lenguajes funcionales, compilación eficiente.

Home Page

Title Page

Contents

◀

▶

◀

▶

Page 10 of 25

Go Back

Full Screen

Close

Quit

Home Page

Title Page

Contents



Page 11 of 25

Go Back

Full Screen

Close

Quit

# Un poco de historia: Lenguajes de Programación (4)

- La Orientación Objeto:
  - Smalltalk (1971): tres evoluciones 72,76 (Xerox), 80 (Apple,DEC,HP,Tecktronix)
  - ADA (1973): Muy alto nivel combinando programación concurrente y en tiempo real; paquetes y mecanismos de excepciones.
  - C++ (1984): Evolución de C, con soporte a mecanismos orientados a objetos.
  - OCAML (1990's): Evolución de CAML, OO.
  - JAVA (1990's): Máquina virtual OO, concurrencia, internet, portabilidad.

Home Page

Title Page

Contents



Page 12 of 25

Go Back

Full Screen

Close

Quit

## Un poco de historia: Lenguajes de Programación (5)

### ■ Otros:

- Oz, Mozart (1990's) : Restricciones, concurrencia, multiparadigma.
- Janus, Cordial (1990's): Programación visual.
- Visual XX (1990's): Interfaces visuales para la programación textual

*Home Page*

*Title Page*

*Contents*



*Page 13 of 25*

*Go Back*

*Full Screen*

*Close*

*Quit*

# La perspectiva de los paradigmas

- Imperativo
- Funcional
- Lógico
- Orientado a Objetos
- Por Restricciones
- Visual
- Concurrente

Home Page

Title Page

Contents



Page 14 of 25

Go Back

Full Screen

Close

Quit

# La perspectiva de los paradigmas: Imperativo

- Orientado por la máquina
- Alto nivel
- Esencial: Asignación y secuenciación
- Programa  $\equiv$  Secuencia de instrucciones
- Ejemplos: Fortran, Algol, Basic, C, Pascal

Home Page

Title Page

Contents



Page 15 of 25

Go Back

Full Screen

Close

Quit

# La perspectiva de los paradigmas: Funcional

- Basado en el cálculo  $\lambda$ .
- Manejo implícito de la memoria.
- Esencial: concepto de función.
- Funciones son ciudadanos de primera clase.
- Programa  $\equiv$  Conjunto de funciones + aplicación
- Ejemplos: LISP, Scheme, ML.

Home Page

Title Page

Contents



Page 16 of 25

Go Back

Full Screen

Close

Quit

# La perspectiva de los paradigmas: Lógico

- Basado en el cálculo de predicados.
- Mecanismo de demostración automática de teoremas.
- Esencial: Concepto de deducción lógica.
- Programa  $\equiv$  Conjunto de axiomas y un objetivo.
- Ejemplos: Prolog I, II, III.

Home Page

Title Page

Contents



Page 17 of 25

Go Back

Full Screen

Close

Quit

# La perspectiva de los paradigmas: Orientado a Objetos

- Basado en la hipótesis: Mundo real  $\equiv$  objetos + interacción
- Basado en el concepto computacional de objeto.
- Esencial: conceptos de objeto, herencia y mensaje.
- Programa  $\equiv$  Conjunto de objetos + mensajes
- Ejemplos: Smalltalk, Java, C<sup>++</sup>, Obliq, Dylan, CLOS, Squeak, ...

Home Page

Title Page

Contents



Page 18 of 25

Go Back

Full Screen

Close

Quit

# La perspectiva de los paradigmas: Por Restricciones

- Basado en el concepto de predicado o relación lógica.
- Esencial: Relación de vinculación o consecuencia lógica.
- Esencial: Búsqueda en árboles y reducción de dominios.
- Programa  $\equiv$  Conjunto de Relaciones entre variables + Estrategia de exploración
- Ejemplos: CLP, Mozart.

Home Page

Title Page

Contents



Page 19 of 25

Go Back

Full Screen

Close

Quit

# La perspectiva de los paradigmas: Visual

- Basado en comunicación gráfica.
- Muy alto nivel.
- Esencial: Todo elemento del lenguaje es un dibujo.
- Programa  $\equiv$  Conjunto de dibujos.
- Ejemplos: Janus, Patchwork, Cordial.

Home Page

Title Page

Contents



Page 20 of 25

Go Back

Full Screen

Close

Quit

# La perspectiva de los paradigmas: Concurrente

- Basado en cálculos de procesos (Cálculo ' $\pi$ ').
- Esencial: Mecanismos de comunicación entre procesos.
- Programa  $\equiv$  Conjunto de procesos.
- Ejemplos: PICT

*Home Page*

*Title Page*

*Contents*



*Page 21 of 25*

*Go Back*

*Full Screen*

*Close*

*Quit*

# Retos de investigación (1)

- En general:
  - Enfoques basados en componentes
  - Interoperabilidad
  - Dinamicidad de los lenguajes vs. seguridad y corrección
  - Fundamentación en teoría de la concurrencia
  - ALto desempeño

Home Page

Title Page

Contents



Page 22 of 25

Go Back

Full Screen

Close

Quit

## Retos de investigación (2)

- En Programación OO:
  - Integración Análisis y Diseño con Implementación
  - Lenguajes basados en prototipos vs. basados en clases
  - Concurrencia y distribución integrada a lo OO.
  - Integración con otros paradigmas: restricciones y funcional.
  - Cómo enseñar a programar?
  - Tipos y programación de alto nivel.

Home Page

Title Page

Contents



Page 23 of 25

Go Back

Full Screen

Close

Quit

## Retos de investigación (3)

- En Programación Visual:
  - Apropiación de lo visual: qué si representar visualmente, qué no?
  - Escalabilidad: Cómo construir y comprender grandes programas visuales?
  - Transparencia de la interfaz: manipulación de programas visuales tan natural y transparente como en los programas textuales.

Home Page

Title Page

Contents



Page 24 of 25

Go Back

Full Screen

Close

Quit

# CRISOL: Una apuesta a la investigación

- *CRISOL*:<sup>a</sup> Laboratorio de Investigación en Tecnologías de Programación Concurrente de Restricciones para el Parque Tecnológico de Software de Cali.
- Alianza Univalle, Javeriana, Parque.
- El objetivo del proyecto es el de propiciar el desarrollo de software innovador en Colombia a través de la apropiación por el *Parque tecnológico de software de Cali* de tecnologías no tradicionales, en particular la programación concurrente por restricciones, mediante el desarrollo de herramientas facilitadoras del proceso de construcción, depuración y puesta en marcha del software basado en esa tecnología.
- Antecedentes: *PATHOS*.

---

<sup>a</sup>Constraints Research and Innovation for Software Solutions

*Home Page*

*Title Page*

*Contents*



*Page 25 of 25*

*Go Back*

*Full Screen*

*Close*

*Quit*

# Referencias para estar al día

- The Study of programming languages, R. Stansifer, Prentice Hall
- Transactions on Programming Languages and Systems (TOPLAS)
- Navegar por [portal.acm.org](http://portal.acm.org): Proceedings SIGPLAN