

Universidad Tecnológica Nacional
Facultad Regional Resistencia

Paradigmas de Programación

Introducción

Paradigmas

- ❑ Un paradigma está constituido por los supuestos teóricos generales, las leyes y las técnicas para su aplicación que adoptan los miembros de una determinada comunidad científica.
- ❑ Es un modelo o esquema fundamental que organiza nuestras opiniones con respecto a algún tema en particular.
- ❑ Los paradigmas establecen límites, para resolver problemas dentro de estos, y de esta forma mejorar o proporcionar nuevas soluciones, ya que estos filtran experiencias, se ajustan a los límites, percepciones o creencias.

Paradigmas

Antonio, padre de Martín, un niño de 8 años, sale manejando desde su casa. Martín, va con él. En el camino se produce un terrible accidente. Un camión, que venía de frente, se sale de su sector de la autopista y embiste de frente al auto de Antonio.

El impacto mata instantáneamente a Antonio, pero Martín sigue con vida. Una ambulancia lo traslada al hospital.

Ni bien llega, los médicos de guardia comienzan a tratar al nene con mucha dedicación pero, luego de charlar entre ellos y estabilizarle las condiciones vitales, deciden que no pueden resolver el problema de Martín. Necesitan consultar. Luego de las consultas pertinentes, se comunican con una eminencia en el tema.

Los médicos del lugar le presentan el caso y esperan ansiosos su opinión. Finalmente, uno de ellos es el primero en hablar: "¿Está usted en condiciones de tratar al nene?", obtiene la siguiente respuesta: "¡Cómo no lo voy a tratar si es mi hijo!".



Pensemos la manera que esto tenga sentido.

Paradigmas de Programación

- ❑ Representan un enfoque particular o filosofía para la construcción del software.
- ❑ *Programming paradigms are heuristics used for algorithmic problem solving. A programming paradigm formulates a solution for a given problem by breaking the solution down to specific building blocks and defining the relationship among them.*

Paradigma	Bloques de Construcción	Relaciones entre los bloques de construcción
Funcional	Funciones	Composicion
Procedural	Procedimientos o Secuencia de Comandos	Jerarquía
Orientado a Objetos	Clases	Herencia

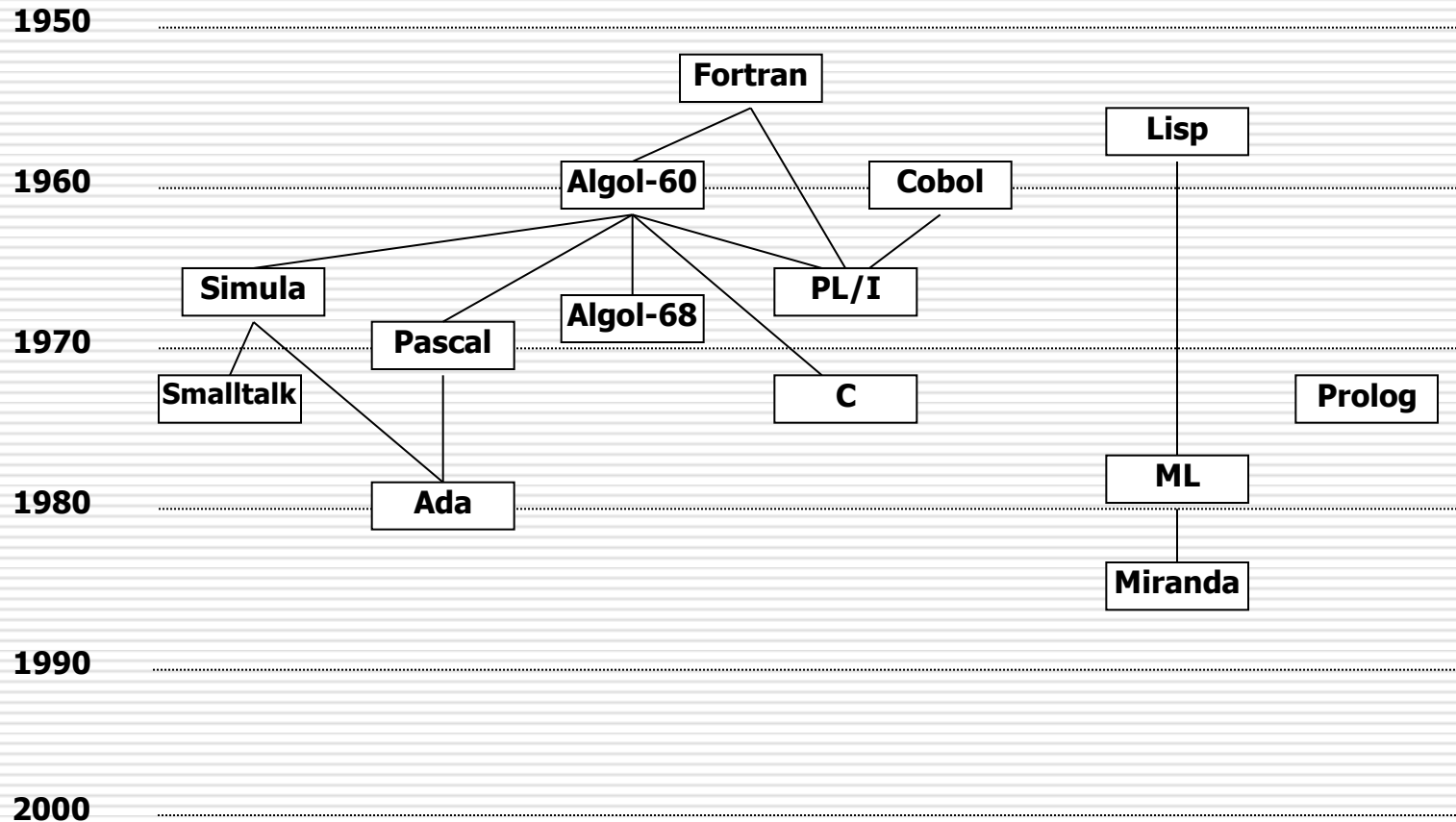
Paradigmas de Programación

- ❑ El paradigma **imperativo** es considerado el más común y está representado, por ejemplo, C, Pascal, COBOL. (*Procedural*: cuando usa procedimientos o funciones)
- ❑ El paradigma **orientado a objetos**. Un lenguaje completamente orientado a objetos es Smalltalk.
- ❑ El paradigma **funcional** está representado por la familia de lenguajes LISP. Otro podría ser Haskell.
- ❑ El paradigma **lógico**, un ejemplo es PROLOG.

Un poco de Historia

- ❑ Los lenguajes de programación de nuestros días fueron desarrollados a principios de los años 50. Numerosos conceptos fueron inventados, probados e incorporados a distintos lenguajes.
- ❑ No son un producto terminado sino que nuevos y revolucionarios conceptos y paradigmas están siendo desarrollados.

Un poco de Historia 2



Paradigma Imperativo

- ❑ Fortran introdujo las expresiones simbólicas y subprogramas con parámetros.
- ❑ COBOL introdujo el concepto de descripción de datos.
- ❑ El Algol-60 introdujo el concepto de estructura de bloques, variables, procedimientos, etc.
- ❑ Influenció a numerosos lenguajes posteriores tan fuertemente que fueron llamados lenguajes estilo Algol.
- ❑ Pascal fue el lenguaje estilo Algol que se hizo más popular porque es simple, sistemático e implementado eficientemente.

Paradigma Imperativo 2

- ❑ Describe la programación en términos del estado del programa y sentencias que cambian dicho estado.
- ❑ Los programas imperativos son un conjunto de instrucciones que le indican al computador cómo realizar una tarea.
- ❑ Las recetas y las listas de revisión de procesos, a pesar de no ser programas de computadora, son también conceptos familiares similares en estilo a la programación imperativa; cada paso es una instrucción.
- ❑ La programación imperativa recibe este nombre porque está basada en comandos que actualizan variables contenidas en almacenamientos.

Paradigma Imperativo 3

- ❑ Tiene estrecha relación para modelar el mundo real.
- ❑ Se caracteriza esencialmente porque las instrucciones que lo componen se ejecutan secuencialmente en un orden preestablecido.
- ❑ La unidad base de ejecución es el *programa* o conjunto de instrucciones ejecutables, que se divide en una serie de módulos o rutinas distribuidos de acuerdo con una organización jerárquica.
- ❑ En este tipo de programación los datos están desorganizados, son meros apéndices de los programas y tan solo proporcionan valores que sirven para realizar cálculos.

Paradigma Imperativo 4

- ❑ Uno de los programas situados en la raíz de la jerarquía recibe el nombre de *programa principal* y los demás se conocen con el nombre de *subprogramas*, *subrutinas*, *funciones* o *procedimientos*.
- ❑ En cuanto a los datos, dependiendo del lenguaje que se trate pueden ser *globales*, accesibles por todos los programas de la jerarquía, o *locales*, utilizables solo por el programa al que pertenece.
- ❑ Dentro de este tipo de programación podemos encontrar dos divisiones:
 - *programación estructurada* o programación sin GOTO: presenta bastantes ventajas en cuanto a la facilidad de modularización, documentación y mantenimiento de los programas;
 - *programación no estructurada* que permiten realizar saltos incondicionales en la ejecución del programa mediante instrucciones GOTO.

Paradigma Orientado a Objetos

- ❑ Es el paradigma que está tomando mayor auge en los últimos años.
- ❑ La programación orientada a objetos deriva de conceptos introducidos por Simula, otro lenguaje del estilo Algol.
- ❑ Smalltalk está basado en clases de objetos, un objeto viene a ser como una variable que puede accederse sólo a través de operaciones asociadas con él.

Paradigma Orientado a Objetos 2

- ❑ Los objetos que son entidades que combinan estado (es decir, datos) y comportamiento (esto es, procedimientos o métodos).
- ❑ La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que se comunican entre ellos para realizar tareas.
- ❑ Lenguajes procedurales:
 - escriben funciones y después les pasan datos.
- ❑ Lenguajes orientados a objetos
 - definen objetos con datos y métodos y después envían mensajes a los objetos diciendo que realicen esos métodos en sí mismos.



Paradigma Funcional

- ❑ Están basados en el cálculo lambda (o λ -cálculo), de Alonso Church.
- ❑ LISP fue el primer lenguaje basado enteramente en el Cálculo Lambda.
- ❑ Lisp en su forma pura, está basado enteramente en funciones sobre listas y árboles.
- ❑ Los lenguajes funcionales modernos, tratan a las funciones como valores de primera clase e incorporan también un sistema de tipos avanzado.



Paradigma Funcional 2

- ❑ La secuencia de computaciones llevadas a cabo por el programa se regiría única y exclusivamente por la reescritura de definiciones más amplias a otras cada vez más concretas y definidas.
- ❑ Los programas escritos en un lenguaje funcional están constituidos únicamente por definiciones de funciones
- ❑ Se verifican ciertas propiedades como la transparencia referencial (el significado de una expresión depende únicamente del significado de sus subexpresiones).
- ❑ No existe asignaciones de variables.
- ❑ Falta de construcciones estructuradas como la secuencia o la iteración (solo recursión)



Paradigma Funcional 3

- Entre los lenguajes funcionales puros:
 - Haskell
 - Miranda

- Los lenguajes funcionales híbridos:
 - Lisp
 - Scheme
 - Ocaml
 - Standard ML

(estos dos últimos, descendientes del lenguaje ML).

Paradigma Lógico

- ❑ Un lenguaje de programación lógica está basado en un subconjunto de la lógica matemática.
- ❑ La computadora es programada para inferir relaciones entre valores más que calcular valores de salida a partir de valores de entrada.
- ❑ PROLOG hizo a la programación lógica popular y es más bien débil e ineficiente, sin embargo fue modificado para agregarle características no lógicas para hacerlo más usable como lenguaje de programación.

Paradigma Lógico 2

- ❑ La mayoría de estos lenguajes se basan en la lógica de predicados.
- ❑ Se definen propiedades de los objetos del dominio de referencia y relaciones entre ellos a través de fórmulas, las cuales usan normalmente el conector implicación.

Paradigma Orientado a Eventos

- ❑ La programación dirigida por eventos es un paradigma de programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema o que ellos mismos provoquen.
- ❑ Será el propio usuario -o lo que sea que esté accionando el programa- el que dirija el flujo del mismo.
- ❑ Se llevarán a cabo las inicializaciones y demás código inicial y a continuación el programa quedará bloqueado hasta que se produzca algún evento.
 - Clic del mouse.
 - Llegada de un mensaje por la red.