



PUC Minas

SEMINÁRIOS DE LP

PROLOG

Outubro

Novembro, 2022

Conteúdo do Seminário

1 - INTRODUÇÃO

*O que é Prolog
Quem foi Alain Colmerauer*

2 - PREDICADO

*Sintaxe, Semântica
Regras, Consultas e Fatos*

3 - HISTÓRIA

*PROLOG I, II, III e IV
TURBO PROLOG
VISUAL PROLOG*

4 - CASES/APLICAÇÃO

*IBM
SISTEMAS
BOEING*

5- PRÁTICA

*Linguagens Similares
Exemplos de Algoritmos*

6 - CONCLUSÃO

*Conclusão
Referencias
Material de apoio*

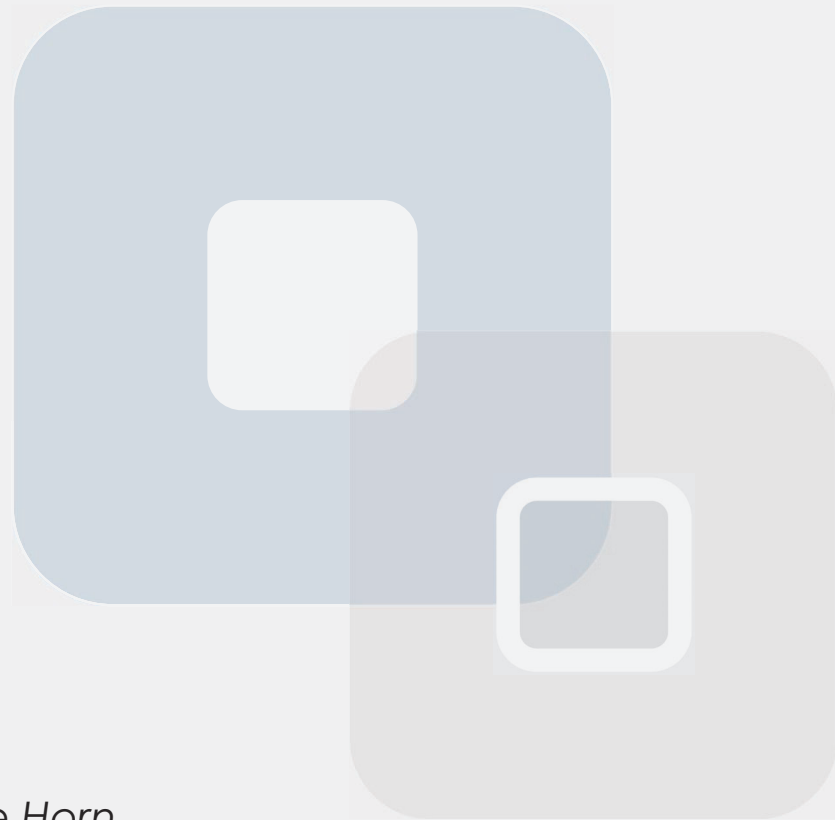
DISTRIBUIÇÃO



PROLOG: O que é?

PROgramation et LOgique

- *Linguística computacional*
- *Paradigma lógico e declarativo*
- *Aplicações de inteligência artificial*
- *Origina-se do teorema da Clausulas de Horn
no formato **H:- B1, ..., Bn.***





Alain Colmerauer

Alain Marie Albert Colmerauer, nascido em na comuna de Carcassonne, no 24 de janeiro de 1941, foi um cientista da computação francês. Teve outras ocupações tais como engenheiro e professor universitário.



FATOS E OS TIPOS DE DADOS

FATOS

São formados por predicados que contém argumentos/objetos cuja instrução é finalizada com o caractere ponto.

Ex:

gato(tom).

tom é um gato?

? - **gato**(tom).

yes.

que coisas (conhecidas) são gatos?

? - **gato**(X).

X = tom;

yes.

TIPOS DE DADOS

- **ÁTOMOS**

- gustavo valadares_castro
- marco rodrigo_costa

- **NÚMEROS**

- 1958 1.958

- **VARIÁVEIS**

- Nome Aluno_de_Graduacao
- Nome Professor_Universitário

- **STRINGS**

- 'Tom é um gato e Jerry um rato'

REGRAS E AS CONSULTAS

REGRAS

luz(acesa) :- interruptor(ligado).

Os caracteres “:” e “–” significam “se”,

avo(X,Z) :- pai(X,Y), pai(Y,Z).

REGRAS RECURSIVAS:

ancestral(X,Y) :- mãe(X,Y).

ancestral(X,Y) :- pai(X,Y).

ancestral(X,Y) :- mãe(X,Z),ancestral(Z,Y).

ancestral(X,Y) :- pai(X,Z),ancestral(Z,Y).

AVALIAÇÃO

- irmaos(X,Y) :- filho(X,Z), filho(Y,Z).

filho(X,Y) :- pai(Y,X).

filho(X,Y) :- mae(Y,X).

mae(marcia, ana).

pai(tomas, ana).

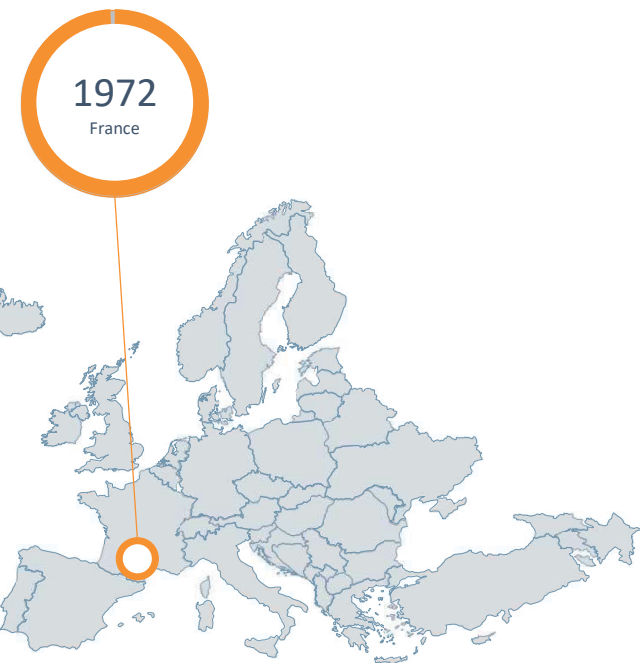
pai(tomas, erica).

pai(marcos, tomas).

?- irmaos(ana, erica).

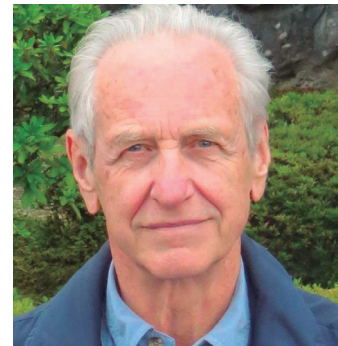
yes.

CONHECENDO A EQUIPE



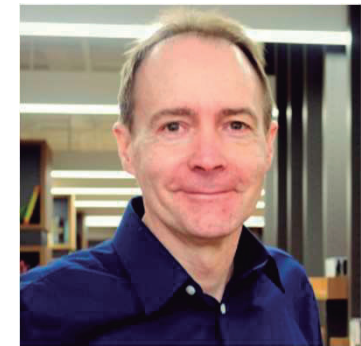
Alain Colmerauer

Cientista da Computação
Professor e Pesquisador
Criador da Linguagem



Robert Kowalski

Professor de Lógica
Cientista da Computação
Matemático



Phillipe Roussel

Cientista da Computação
Professor e Pesquisador
Criador da Linguagem

Principais contribuintes para a criação da Linguagem

VERSÕES PURAS PROLOG

VERSÕES COMO **TURBO PROLOG** E **VISUAL PROLOG** SÃO TIDAS COMO NÃO PURAS COMO O PROLOG "CLÁSSICO"

Implementações - 1967

- W-Grammar
- Q-System
- Homem Máquina
- Linguagem Natural
- Prolog I

PROLOG I e II

1973 surge Prolog I, onde havia aplicação nas áreas de computação simbólica, solução de teoremas e, também, é criada a empresa PrologIA para comercializa-lo.

PROLOG III

O algoritmo do tipo Simplex é implementado nesta versão

PROLOG IV

Ofereceu uma biblioteca composta por um conjunto de mais de restrições, que conta com listas, booleanos, inteiros e reais

LINGUAGEM “SIMILAR”

PROLOG

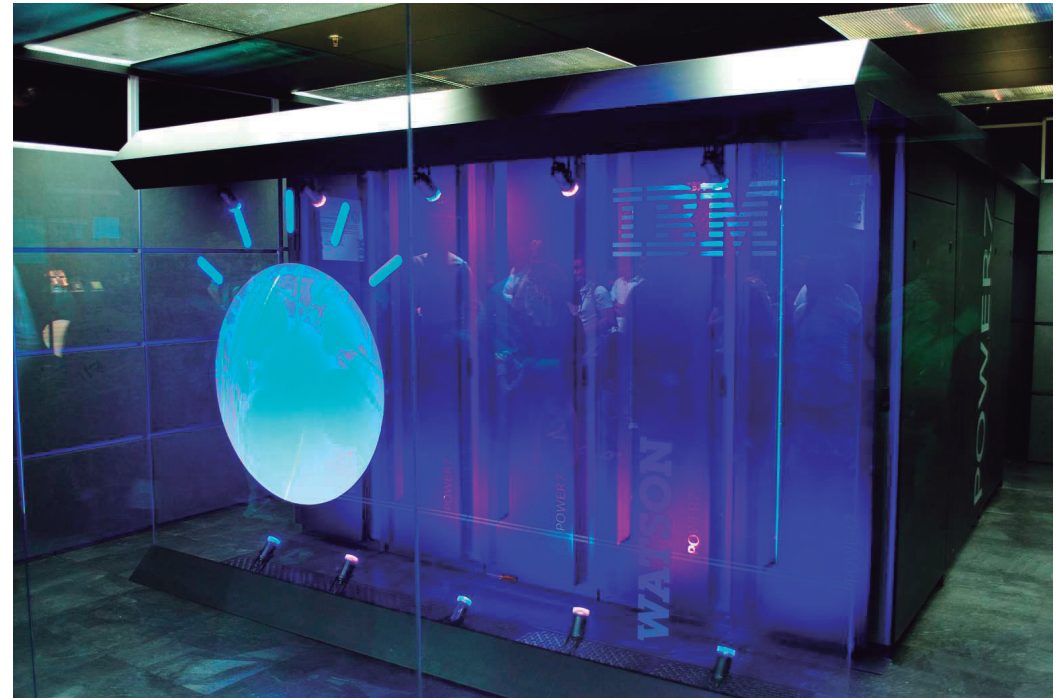
Prolog suporta
paradigma de
programação
lógica.

LISP

Lisp suporta
paradigmas
funcionais,
procedurais,
reflexivos;

CASE E APLICAÇÃO

- **SISTEMAS ESPECIALISTAS**
 - Auxilio nas tomadas de decisões
- **IBM**
 - IBM WATSON
- **BOEING**
 - Controle e trafego aéreo
- **SWI-PROLOG**
 - Ferece uma ambiente Prolog



IBM WATSON

EXEMPLOS DE CÓDIGOS I

```
?- write('Hello World!'), nl.  
Hello World!  
true.  
  
?-
```

Hello World - PROLOG

```
(defparameter *httpd*  
  (hunchentoot:start  
    (make-instance 'hunchentoot:acceptor  
                   :port 8080)))  
(princ "Hunchentoot started on port "  
(princ *httpd-port*)  
(terpri)  
  
(hunchentoot:define-easy-handler (hello-world (:uri "/hello"))  
  ()  
  (with-html-output (*standard-output* nil :indent t)  
    (:html  
      (:head  
        (:title "Hello World"))  
      (:body  
        (:p "Hello world!..."))))))
```

Hello World - LISP

EXEMPLOS DE CÓDIGOS II

```
void bubbleSort( int a[], int n)
{
    int i,j,temp; // for a={1,2,3,4,5} n is 5
    n = n - 1;    // bcz otherwise it will get out of index
    for(i=0; i<n; i++)
    {
        for(j=0; j<n-i; j++)
        {
            if(a[j]>a[j+1])
            {
                temp = a[j+1];
                a[j+1] = a[j];
                a[j] = temp;
            }
        }
    }
}
```

BubbleSort - C

```
bubblesort ( List, SortedList) :-
    swap ( List, List1 ), ! ,
    bubblesort ( List1, SortedList) .
bubblesort ( List, List).

swap ( [ X, Y | Rest ], [ Y, X | Rest ] ) :-
    X > Y, ! .
swap ( [ Z | Rest ], [ Z | Rest1 ] ) : -
    swap (Rest, Rest1 ).
```

Bubble Sort - Prolog

CONCLUSÃO



Hernane Velozo

Engenharia de Computação
2º/2022 - PUC Minas



Artigo escrito, PDF e referências

INTRODUÇÃO AO PROLOG



VIA: IA Expert Academy

O que os seres humanos precisam não é utopia, mas entropia, uma cidade real que podem construir, um lugar que satisfaça o sonhador e seja aceitável para o cientista, um lugar onde as projeções do artista e do construtor se fundem.

Constantinos A. Doxiadis

REFERÊNCIAS

BRATKO, Ivan. **Prolog Programming for Artificial Intelligence** - (2nd edition). Addison-Wesley, 1993.

STERLING, L. & **SHAPIRO**, E. **The Art of Prolog: advanced programming techniques** - (2nd edition). MIT Press, 1994.

HOGGER, Christopher John. **Essentials of Logic Programming**. Oxford University Press, 1990.

TOWNSEND, Carl. **Advanced techniques in Turbo Prolog** Sybex, 1987.

KOWALSKI, Robert A. **Lógica, Programación e Inteligencia Artificial** Díaz de Santos, 1986.

CLOCKSIN, William. Clause, and Effect. **Prolog Programming for the Working Programmer**. Springer-Verlag, 1997.

AMBLE, Tore. **Logic Programming and Knowledge Engineering** Ed. Addison-Wesley, 1987.

ZHANG, J. and Grant. **An Automatic Difference-list Transformation Algorithm for Prolog** P. W., 1988.