



# PROLOG

Vitor R. Di Toro - 983

Erick Ribeiro - 1389

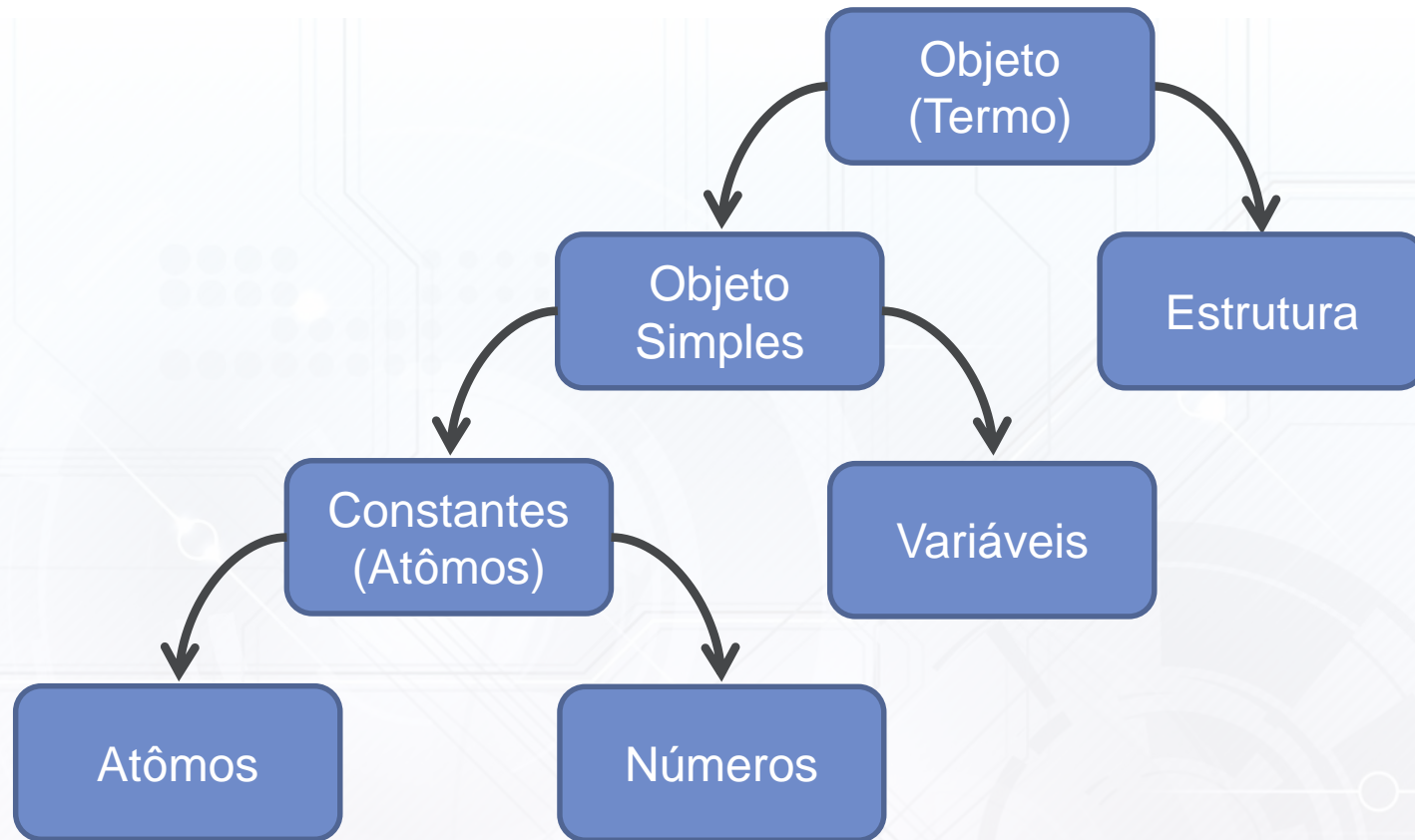
Lucas Teixeira - 709

# CARACTERISTICAS

- **Criada por:** Alain Colmerauer e Robert Kowalski em 1972;
- **Nome:** Abreviação de "**PRO**grammation en **LOG**ique";
- **Paradigma:** Lógico, declarativo;
- **Compiladores:** GNU PROLOG e SWI-PROLOG;
- **Aplicações:** AI & Linguística Computacional;



# OBJETOS



# ÁTOMOS

- Átomo: sequência constituída de letras, números e underscore, iniciada por uma letra minúscula, que representa as constantes de texto;
- Se um átomo não alfanumérico é necessário, pode-se usar qualquer sequência entre aspas simples;
- **Exemplos:** edson1847, albert\_einstein, 'seu sexo?', 'masculino';

# VARIAVEIS

- No Prolog uma variável não é um contêiner cujo valor pode ser atribuído, como ocorre nas outras linguagens;
- Declaração feita da mesma forma que átomos, porém iniciada com letra maiusculo ou underscore;
- **Exemplos:** X, Nome, Professor\_do\_Inatel, \_ (*variável anônima*)
- **Variável Anônima:** Quando uma variável aparece em uma única clausa, não é necessário dar um nome a ela.

**Sem Variável Anônima:** temfilho(X) :- progenitor(X,Y).

**Com Variável Anônima:** temfilho(X) :- progenitor(X,\_).



# VARIÁVEIS (Exemplo)

- **Fatos Conhecidos:**

gosta(joao,flores);  
gosta(joao,maria);  
gosta(paulo,maria);

- **Questão:**

**?** - gosta(joão,X)

- **Resultado:**

A busca é realizada na ordem em que os fatos foram inseridos, gerando a seguinte saída:

**X = flores ;**  
**X = maria.**

# NÚMEROS & OPERADORES

- Em Prolog, os números podem ser **Interiros** ou **Reais**;
- Os operadores que a linguagem aceita são de dois tipos:

Operadores Aritiméticos	
Adição	<b>+</b>
Subtração	<b>-</b>
Multiplicação	<b>*</b>
Divisão	<b>/</b>
Divisão Inteira	<b>//</b>
Resto da Divisão	<b>mod</b>
Potênciação	<b>**</b>
Atribuição	<b>is</b>

Operadores Relacionais	
$X > Y$	X é maior que Y
$X < Y$	X é menor que Y
$X \geq Y$	X é maior ou igual a Y
$X \leq Y$	X é menor ou igual a Y
$X =:= Y$	X é igual a Y
$X = Y$	X unifica com Y
$X \neq Y$	X é diferente de Y

# NÚMEROS & OPERADORES (Exemplo)

- O operador **=** tenta unificar apenas

?-  $X = 1 + 2$ .

**$X = 1 + 2$**

- O operador **is** força a avaliação aritmética

?-  $X \text{ is } 1 + 2$ .

**$X = 3$**

- Se a variável à esquerda do operador **is** já estiver instanciada, Prolog apenas compara o valor da variável com o resultado da expressão à direita do operador **is**

?-  $X = 3, X \text{ is } 1 + 2$ .

**$X = 3$**

?-  $X = 5, X \text{ is } 1 + 2$ .

**false**



# QUESTÕES

- A sintaxe das questões varia de acordo com os compiladores;
- **Exemplo:**  
    ?- amiga(joana,maria).
- Quando uma questão é feita, o Prolog realiza uma busca na sua base de conhecimento;
- O retorno de uma questão é sempre "TRUE" ou "FALSE", conforme a exista ou não um fato igual a questão;

# LISTAS

- Uma lista não é um tipo de dados à parte, mas sim definida por uma construção recursiva (usando o termo '.'):
- O primeiro elemento, chamado cabeça, é H, que é seguida pelo conteúdo do restante da lista, T, também chamado de cauda.

- **Exemplo 1:**

?- Lista = .(brasil, .(uruguai, .(argentina, .(paraguai, []))).  
Lista = [brasil, uruguai, argentina, paraguai].

- **Exemplo 2:** A lista [1, 2, 3], pode ser representada como:

.(1, .(2, .(3, .(4, [])))) .

Ou com um atalho sintático:

[H|T].

# REGRAS

- As **Regras** são um exemplo de **Termo Composto**;
- Em comparação, **Regras** são como **funções**;

- **Exemplos:**

**% Fatos**

gerou(maria, antonio).  
gerou(joao, antonio).

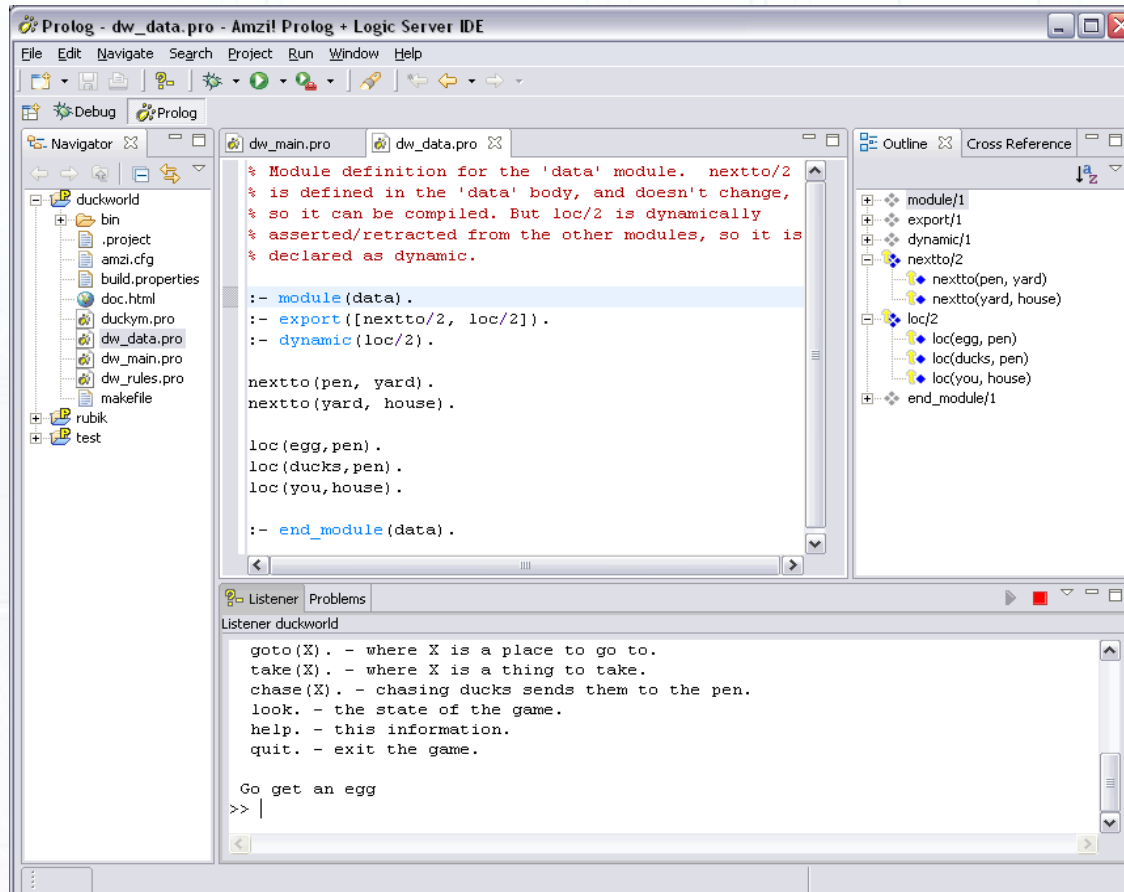
**% Regras**

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



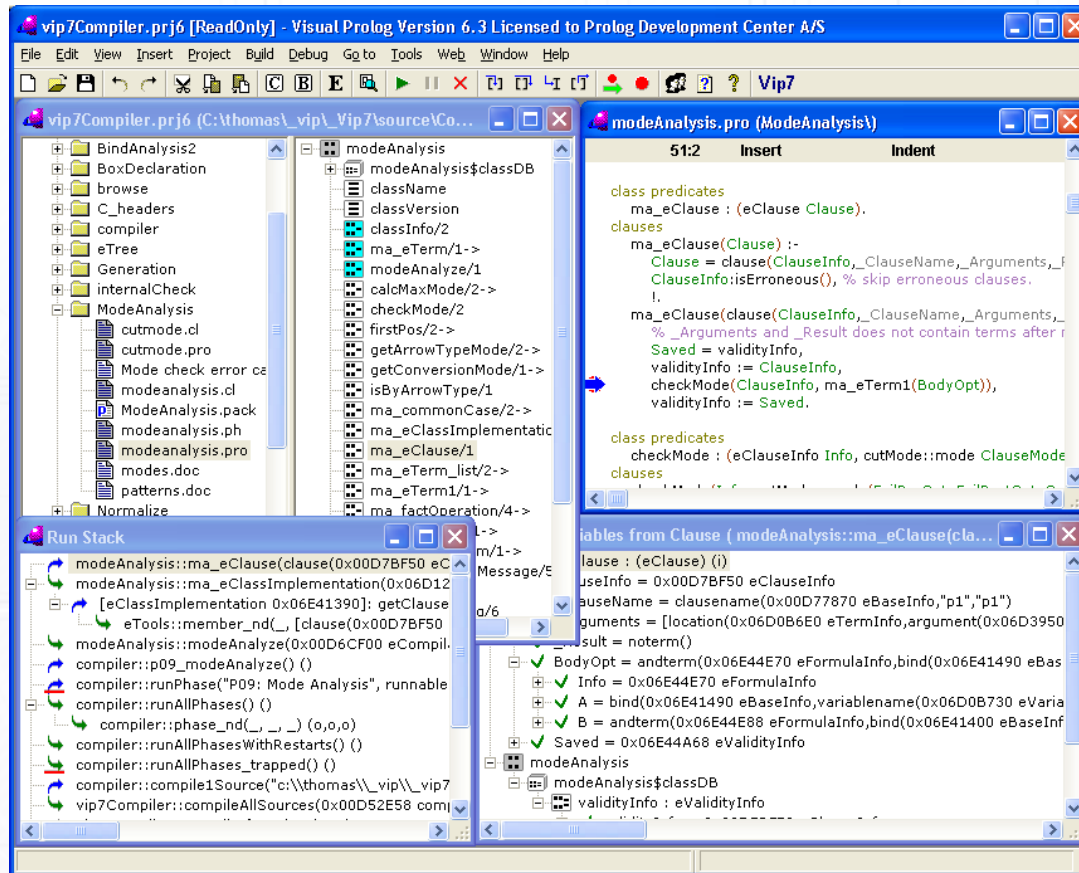
# IDEs

## ○ Amzi! IDE



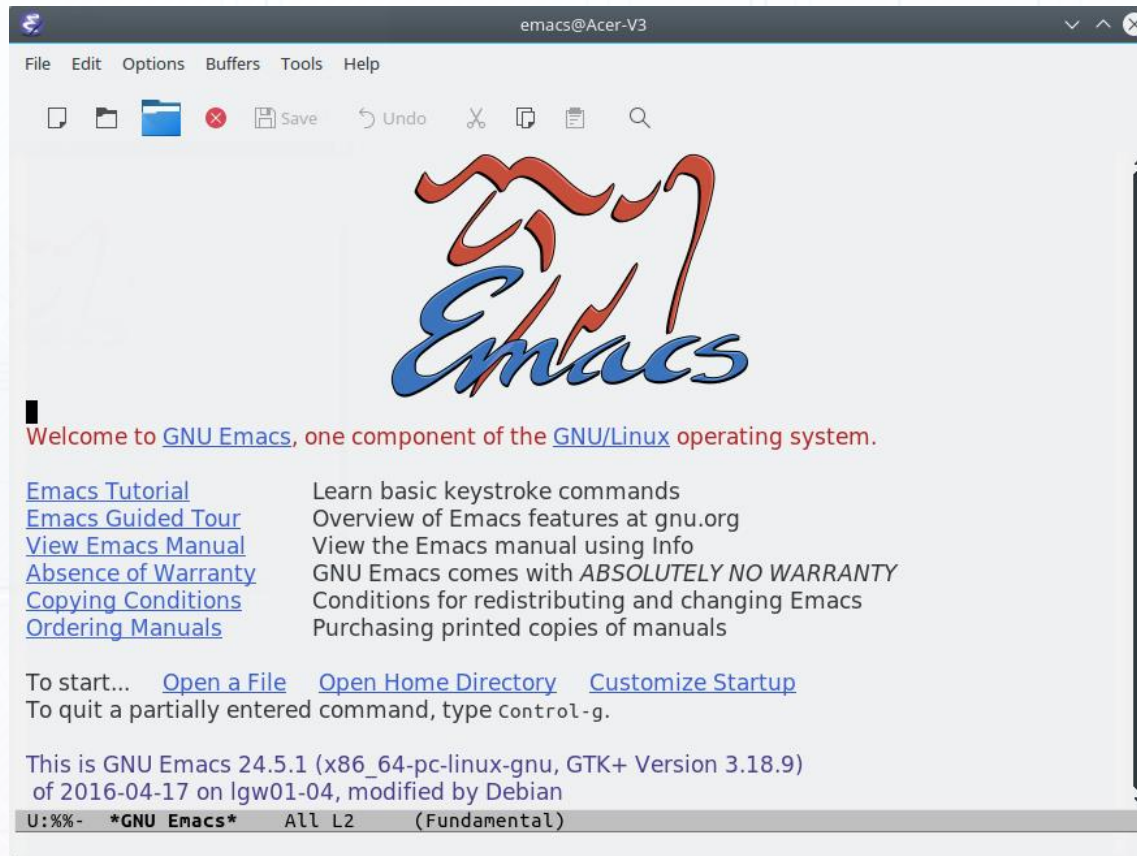
# IDEs

## Visual Prolog



# IDEs

## ○ GNU EMACS



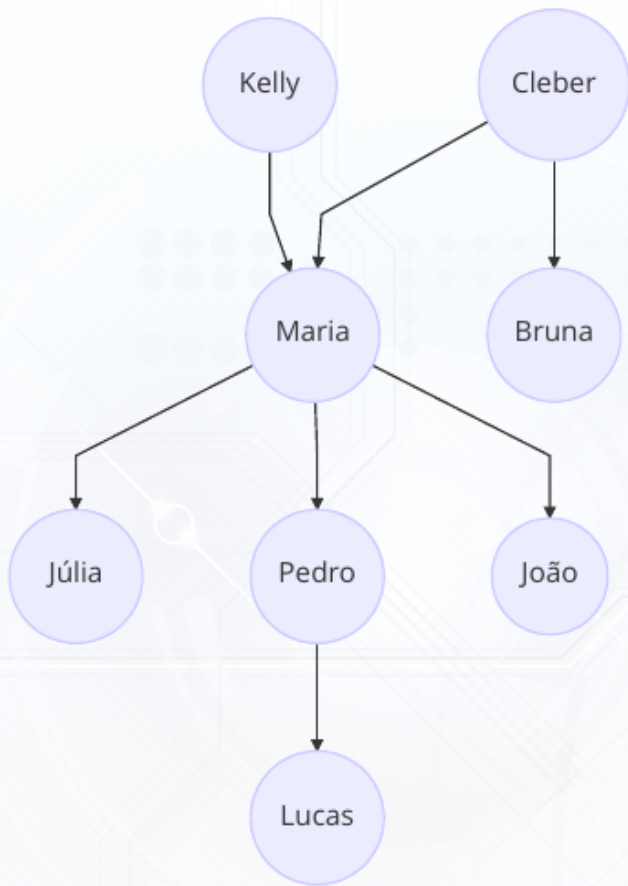
**Stallman**



**Approves**



# EXEMPLO PRATICO



```
vitor@Acer-V3 ~/git/SeminarioEC004-Prolog
Exemplo 1 x Exemplo 2
vitor@Acer-V3 ~/git/SeminarioEC004-Prolog 82x21

?- consult(exemplo2).
true.

?- ['exemplo2'].
true.

?- filho_masculino(X,maria).
X = pedro ;
false.

?- consult(exemplo2).
true.

?- ['exemplo2'].
true.

?- filho_masculino(X,maria).
X = pedro ;
X = joao.

vitor@Acer-V3 ~/git/SeminarioEC004-Prolog 82x21
Fatos ou Base de Conhecimento
gerou(kelly,maria).
gerou(cleber,maria).
gerou(maria,julia).
gerou(maria,pedro).
gerou(maria,joao).
gerou(pedro,lucas).

feminino(kelly).
feminino(maria).
feminino(julia).
feminino(bruna).
masculino(cleber).
masculino(pedro).
masculino(lucas).
masculino(joao).

% Regras
filho(X, Y) :-
    gerou(Y,X).

1,1 Topo
```