



**UNIVERSIDADE FEDERAL DO CEARÁ**  
**CAMPUS DE QUIXADÁ**

Trabalho final da disciplina de Lógica para Computação

**Tema: Mitologia Nórdica**

Equipe:

Gildard C.A. Myehouenou

Gabriel de Souza Lins

# Introdução

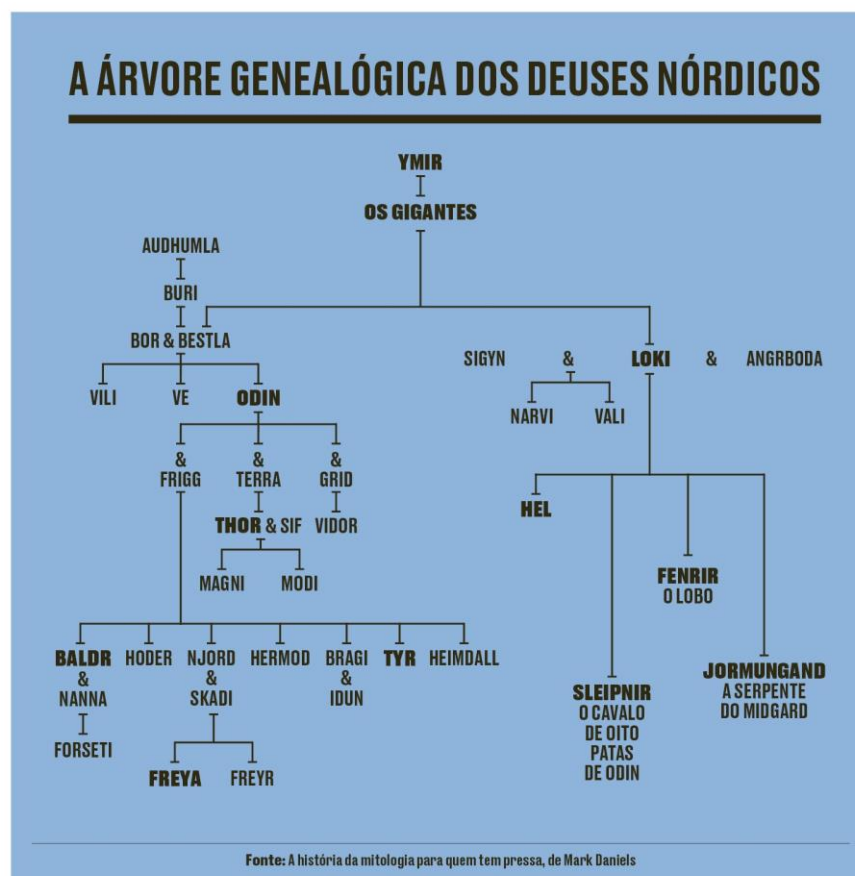
Prolog é uma linguagem de programação lógica. O nome Prolog é um acrônimo de PROgramação LOGica. Ela foi criada por Alain Colmerauer et Phillippe Roussel ao cerca de 1972. O propósito era de criar uma linguagem de programação em que seriam definidas as regras lógicas que se esperam de uma solução e deixar o compilador a transformar em uma sequência de instruções. Prolog é usada na área da IA (Inteligência Artificial) e no tratamento linguístico por computador, principalmente nas linguagens naturais.

## I- Descrição do domínio familiar e árvore genealógica

### A- Descrição do domínio familiar

Nesse trabalho, nós nos inspiramos de uma família muito conhecida no mundo. Essa é a família dos deuses da mitologia nórdica. Essa família tem 5 gerações. O primeiro elemento dessa família é o gigante Ymir. Na mitologia nórdica, temos alguns casos de descendentes que não são o fruto da união de uma pessoa de sexo masculino e uma outra de sexo feminino. Inclusive, temos também uma outra pessoa (Buri) que é descendente de uma vaca (Audhumla). Os demais elementos da família são na maioria dos casos, descendentes da união de uma pessoa de sexo masculino e uma outra de sexo feminino.

### B- Árvore genealógica



## **II- Consultas a serem realizadas**

Nós vamos poder fazer dois tipos de consultas nesse programa. As consultas unárias e as consultas binárias.

### **A- Consultas unárias**

As consultas unárias são aquelas que vamos poder executar no botão “Consulta”. Para executá-las, basta digitar o nome de um membro da família e depois apertar um botão corresponde à informação (regras) que queremos obter dele. O resultado é imprimido em um novo dialog.

### **B- Consultas booleanas**

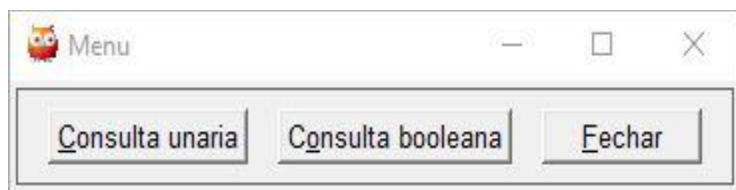
Essas consultas são do tipo booleano. Elas são executadas digitando nos dois campos de texto os nomes dos membros da família. Depois, basta apertar no botão da regra que queremos testar. Ela vai retornar verdadeiro ou falso.

### **C- Exemplos de consultas**

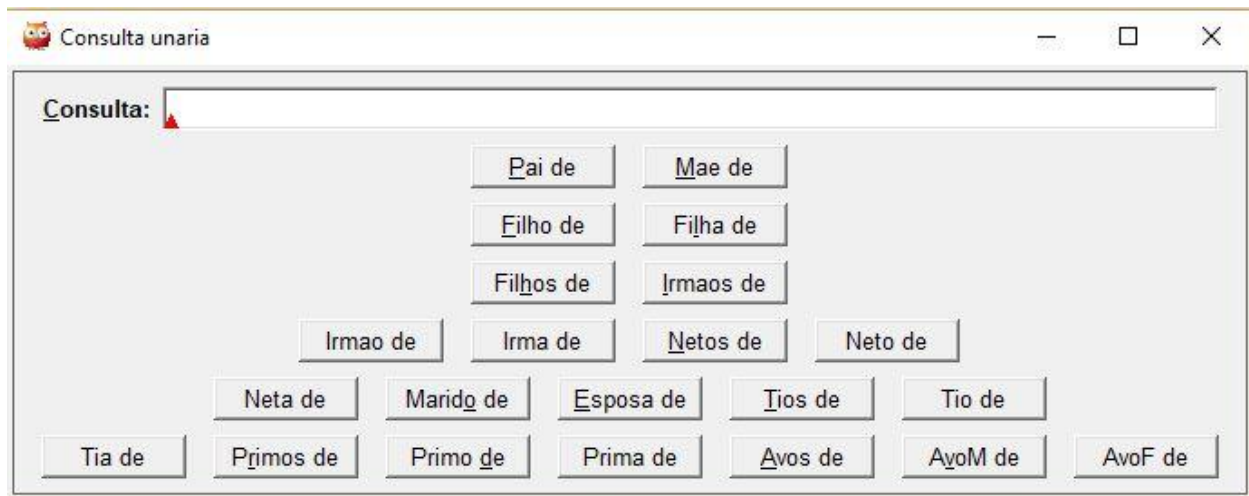
- \* Quem é filho de Odin: deve retornar Thor, Vidar, Baldr, Heimdall, Bragi, Tyr, Hermod, Njord, Hoder;
- \* Quem são os pais de Odin: deve retornar Bor e Bestla;
- \* Quem são os netos de Odin: deve retornar Magni, Modi, Forseti, Freya, Freyr;
- \* Odin e Thor são irmãos: deve retornar falso.

## **III- Prints de tela da interface gráfica**

### **A- Menu principal**




## B- Consulta unária



A screenshot of a software window titled "Consulta unária" with a standard Windows title bar (minimize, maximize, close buttons). The window contains a text input field labeled "Consulta:" with a red triangle icon to its right. Below the input field is a grid of buttons representing various family relationships. The buttons are arranged in a roughly triangular shape, with the most common relationships at the top and more specific ones at the bottom.

Pai de		Mãe de				
Filho de		Filha de				
Filhos de		Irmãos de				
Irmão de	Irmã de	Netos de	Neto de			
Neta de	Marido de	Esposa de	Tios de	Tio de		
Tia de	Primos de	Primo de	Prima de	Avos de	AvoM de	AvoF de

## C- Consulta binária



A screenshot of a software window titled "Verdadeiro ou Falso" with a standard Windows title bar. The window contains two text input fields, "Objeto x:" and "Objeto y:", each with a red triangle icon to its right. Below the input fields is a grid of buttons representing various family relationships between two objects (x and y). The buttons are arranged in a grid, with the most common relationships at the top and more specific ones at the bottom.

Filho x y	Filha x y	Irmão x y	Irmã x y
Pai x y	Mãe x y	Marido x y	Esposa x y
Tio x y	Tia x y	Primo x y	Prima x y
AvoM x y	AvoF x y	Neto x y	Neta x y
Filhos x y	Primos x y	Tios x y	Netos x y
Avos x y		Irmãos x y	

## Conclusão

Além de aplicações puramente Prolog, a possibilidade de combinar Prolog a linguagens tradicionais permitiu no início dos anos 85 de implementar em várias aplicações, módulos inteligentes. Ainda tem várias coisas a descobrir no domínio da inteligência Artificial, e com o tempo, tecnologia para facilitar a vida dos humanos serão descobertas. Esse trabalho nos permitiu de poder explorar um pouco o mundo da lógica e seus derivados, aprender uma nova linguagem de programação e desenvolver um pouco mais nosso espírito lógica.

Referências:

[https://fr.wikipedia.org/wiki/Prolog#Notes\\_et\\_références](https://fr.wikipedia.org/wiki/Prolog#Notes_et_références)

[http://www.swi-prolog.org/pldoc/doc\\_for?object=manual](http://www.swi-prolog.org/pldoc/doc_for?object=manual)