Protótipo de Software de Apoio ao Aprendizado da Linguagem de Programação Lógica Prolog

ORIENTANDO : WAGNER M. STAHNKE

ORIENTADOR : ROBERTO HEINZLE

FURB - Universidade Regional de Blumenau
Bacharelado em Ciências da Computação

Roteiro da Apresentação

- Introdução
- Trabalhos Correlatos
- Aspectos Matemáticos
- Programação Lógica
- Ambiente Prolog
- Implementação do Protótipo
- Conclusão
- Extensões

Introdução

- História
- Área
- Motivação
- Objetivos

Introdução - História

- Prolog, "PROgramming in LOGic". Edimburgo - Escócia no início da década de 70
- Sua popularidade mundial aconteceu devido ao projeto Japonês de Computadores de Quinta Geração

Introdução - Área

- A principal área de Aplicação para o PROLOG é a Inteligência Artificial (IA)
- O presente trabalho se enquadra na Área da Educação

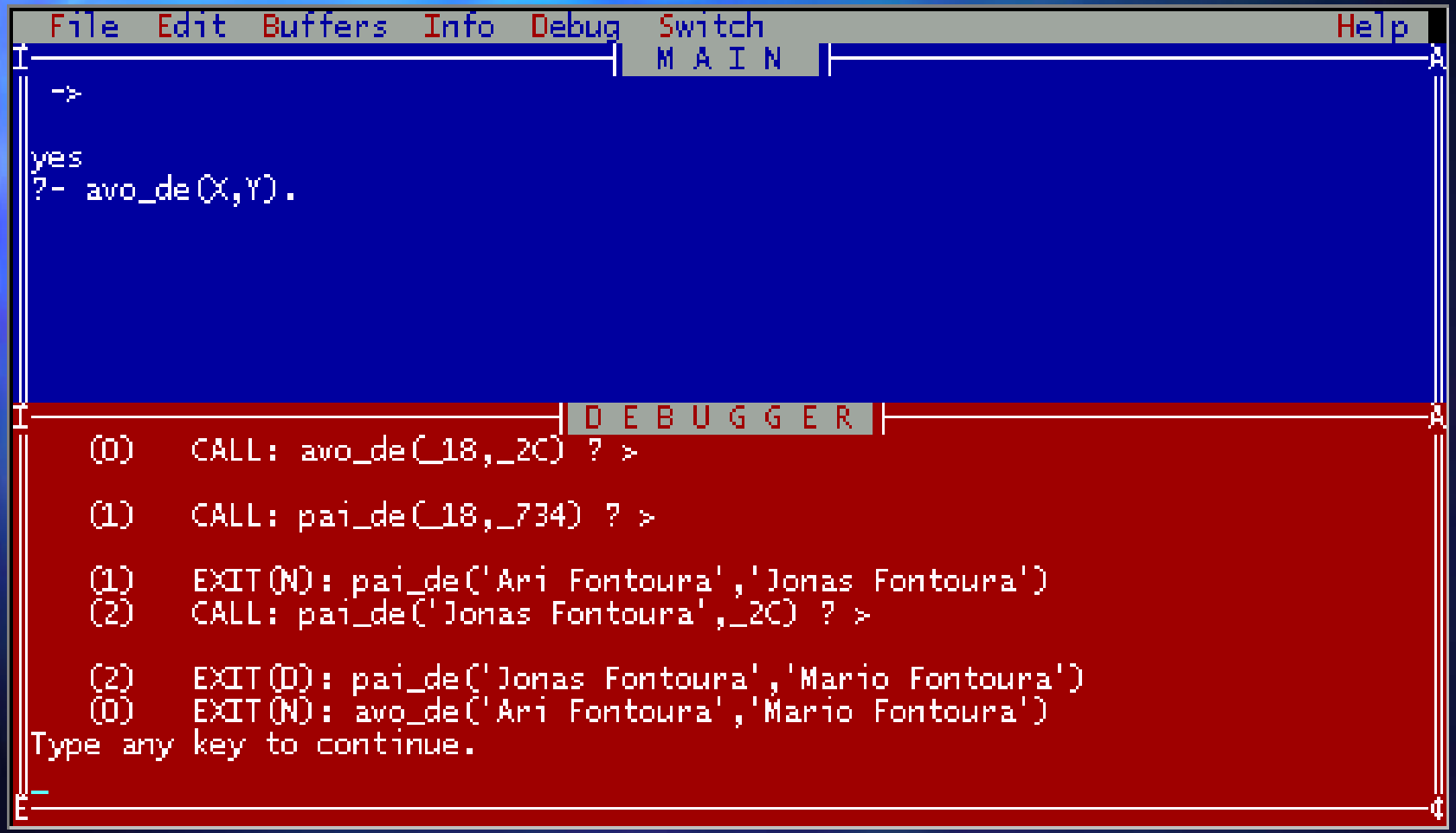
Introdução - Motivação

- Existem diversas razões para se fazer um estudo sobre programação lógica
- A programação lógica oferece uma maneira diferente de pensar sobre a resolução de problemas
- Sua Linguagem se dá de forma mais Natural do que outras linguagens

Introdução - Objetivos

- Desenvolvimento de um protótipo de software capaz de visualizar o processo de execução de programas escritos na linguagem de programação Prolog

Trabalhos Correlatos



Programação Lógica (PL)

- Relações
- Lógica
- Programação em Lógica
  - Fatos e Regras
- Áreas de Aplicação

Aspectos Matemáticos: Lógica das Proposições

- Fórmulas Atômicas:
  - A, B, PedroTemCarro
- Símbolos:
  - Conectivos: $\wedge$ (e), $\vee$ (ou), $\Rightarrow$ (s somente se), $\neg$ (negação).
- Regras de Sintaxe:
  - Uma fórmula atômica é uma fbf **(fórmula bem formada)**
  - Se α é uma fbf então $(\neg \alpha)$ é uma fbf
  - Se α e β são fbfs, então $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \Rightarrow \beta)$ e $(\alpha \equiv \beta)$ são fbfs

Exemplos de fbfs:

$(A \wedge B) \Rightarrow (\neg C)$

$A \Rightarrow (\neg B)$

$(A \vee B) \Rightarrow C$

$(A \Rightarrow B) \Rightarrow (\neg C \Rightarrow \neg D)$

$\neg \neg A$

Aspectos Matemáticos: Lógica dos Predicados

- Fórmulas Atômi
  - pai(X, Y), home
- Símbolos: (além d
  - Quantificadores: ∀ (para todo), ∃ (existe)
- Regras de Sintaxe: (além das já citadas)
  - se $t_1,\ldots,t_n$ são termos e p é um predicado que tem n argumentos, então $p(t_1,\ldots,t_n)$ é uma fbf
  - se V é uma variável e α é uma fbf, então $(\forall V\,\alpha)$ e $(\exists V\,\alpha)$ são fbfs

Exemplos de fbfs:

pai(joão, vitor)

$\forall X\ \forall Y\ (\text{pai}(Z, X) \wedge \text{pai}(Z, Y) \Rightarrow \text{irmao}(X, Y))$

$\forall X\ (\text{humano}(X) \Rightarrow \text{mortal}(X))$

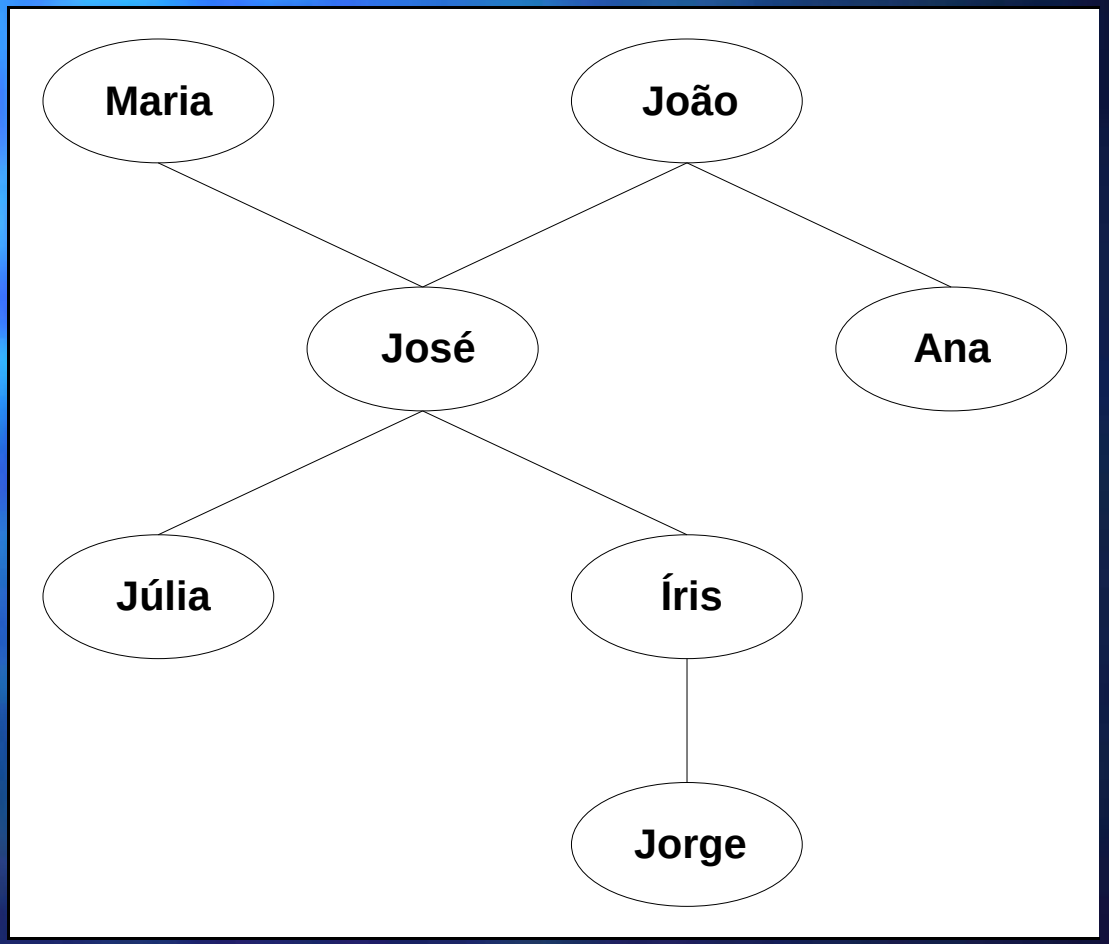
$\exists X\ \exists Y\ (\text{humano}(X) \wedge \text{humano}(Y) \wedge \text{ama}(X, Y))$

Ambiente Prolog

- Exemplo
- Fatos
- Regras
- Construções Recursivas

Prolog: Exemplo

- Família



Prolog: Fatos

```
progenitor(maria, josé).
progenitor(joão, josé).
progenitor(joão, ana).
progenitor(josé, júlia).
progenitor(josé, íris).
progenitor(íris, jorge).
```

```
?- progenitor(X, íris).
X=josé
?- progenitor(josé, X).
X=júlia;
X=íris;
não
```

```
?- progenitor(josé, íris).
sim
?- progenitor(ana, jorge).
não
?- progenitor(luís, íris).
não
```

```
?- progenitor(X, Y).
X=maria  Y=josé;
X=joão   Y=josé;
X=joão   Y=ana;
X=josé   Y=júlia.
```

```
?- progenitor(X, Y), progenitor(Y, jorge).
X=josé  Y=íris.
```

Prolog: Regras

```
filho(Y, X) :-
      progenitor(X, Y).
avo(X, Z) :-
      progenitor(X, Y),
      progenitor(Y, Z).
irmao(X, Y) :-
      progenitor(Z, X),
      progenitor(Z, Y).
```

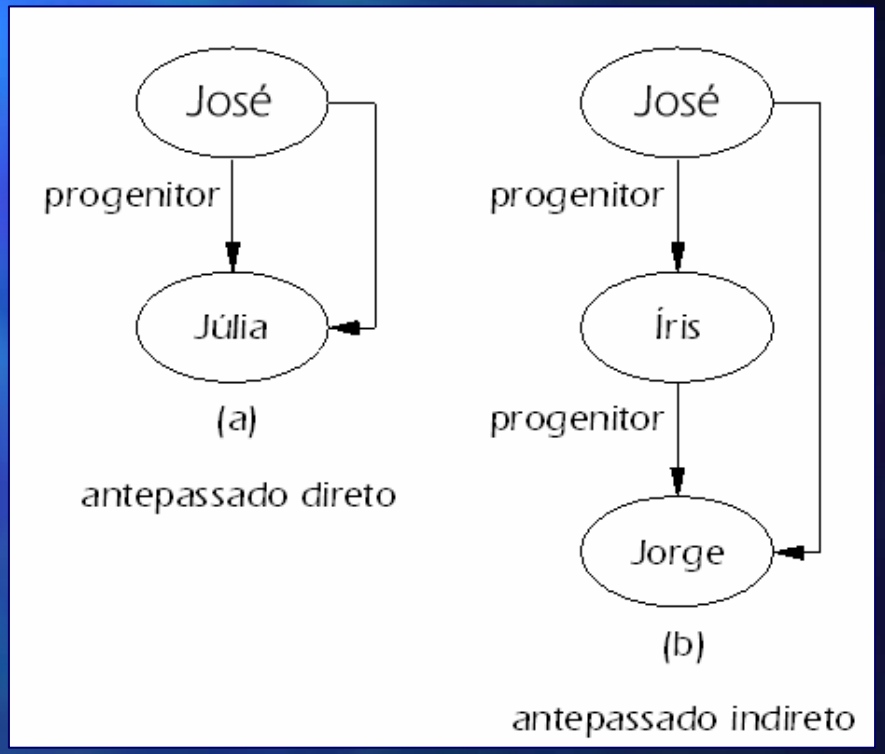
```
?- filho(X, josé).
X=júlia;
X=íris.
?- avo(X, jorge).
X=josé;
?- irmao(X, íris).
X=júlia;
X=íris.
```

```
?- filho(jorge, X), irmao(X, Y), avo(joão, Y).
X=íris  Y=júlia.
```

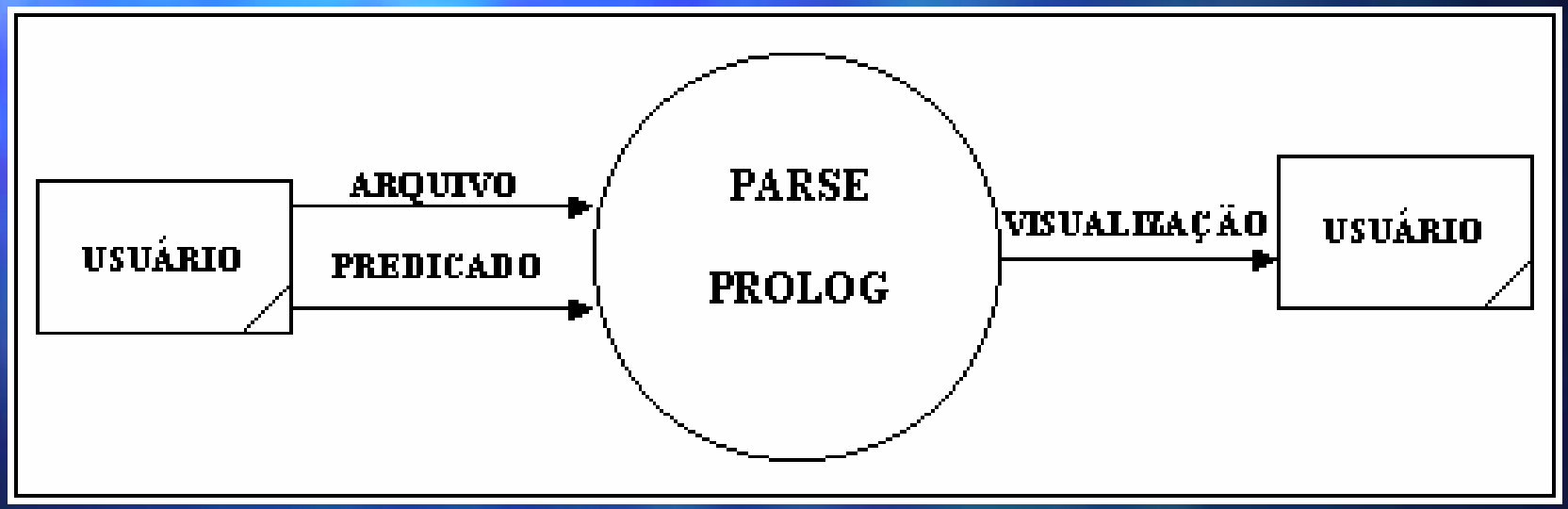
Prolog: Construções Recursivas

```
antepassado(X, Z) :-
      progenitor(X, Z).
antepassado(X, Z) :-
      progenitor(X, Y),
      antepassado(Y, Z).
```

```
?- antepassado(maria,julia).
sim
?- antepassado(X, jorge).
X=íris;
X=joão;
X=maria.
```



Implementacão do Protótipo: Especificação



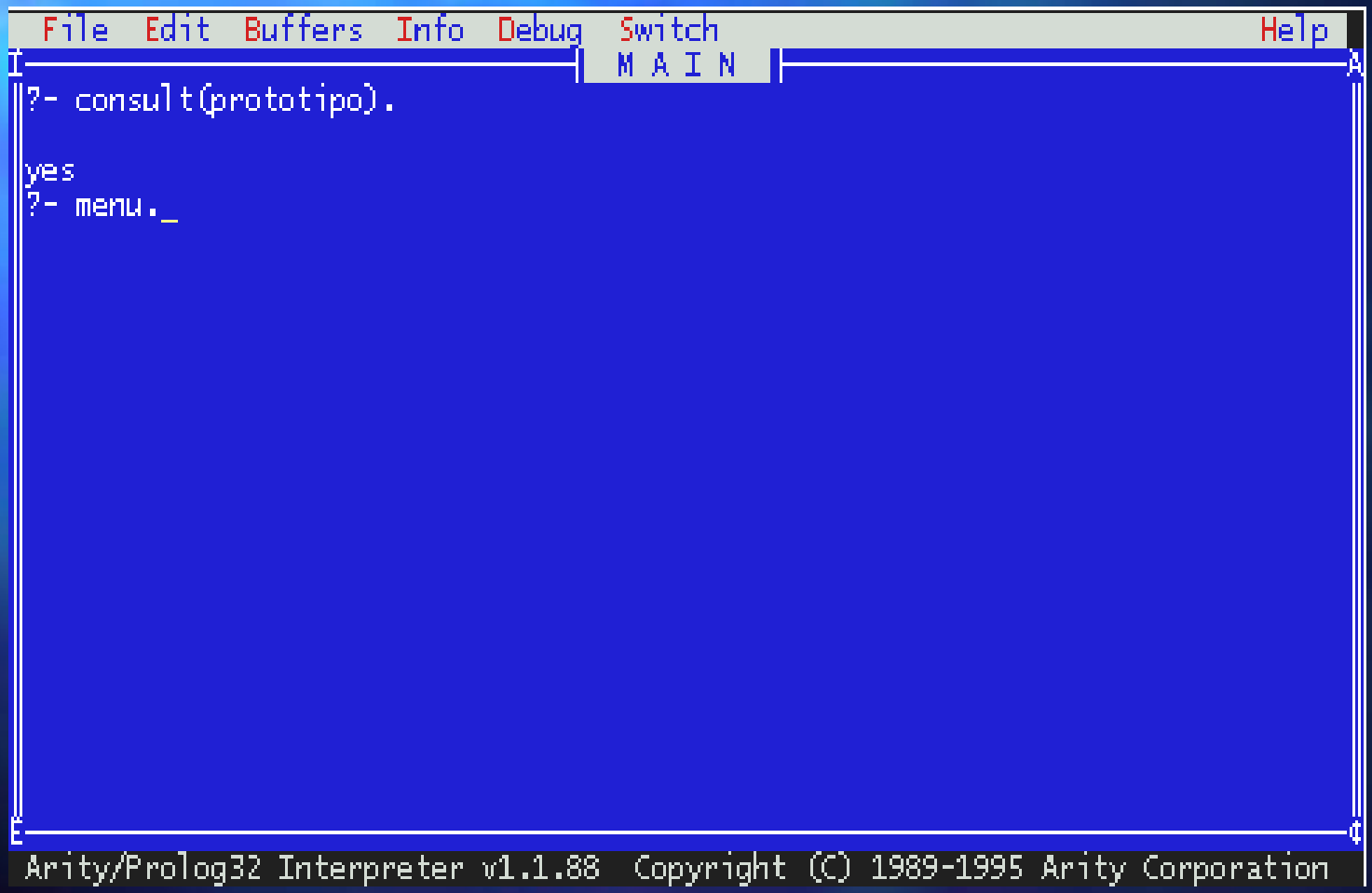
Implementacão do Protótipo: Especificação

- <regra>::= <condição>
- <condição>::= <cláusula1> | <cláusula2> | <cláusula3> | <cláusula4>
- <cláusula1>::= <predicado> e <predicado> e <valor>
- <cláusula2>::= <sistema> e <valor>
- <cláusula3>::= <predicado> e <valor>
- <cláusula4>::= <predicado> e <valor>
- <predicado>::= = = | !=
- <valor>::= 1 | 2 | 3 | 4 | 5 | .....|10
- <sistema>::= <predicado_sistema>
- <predicado_sistema>::= write | read | keyb | nl | .....| open | .....

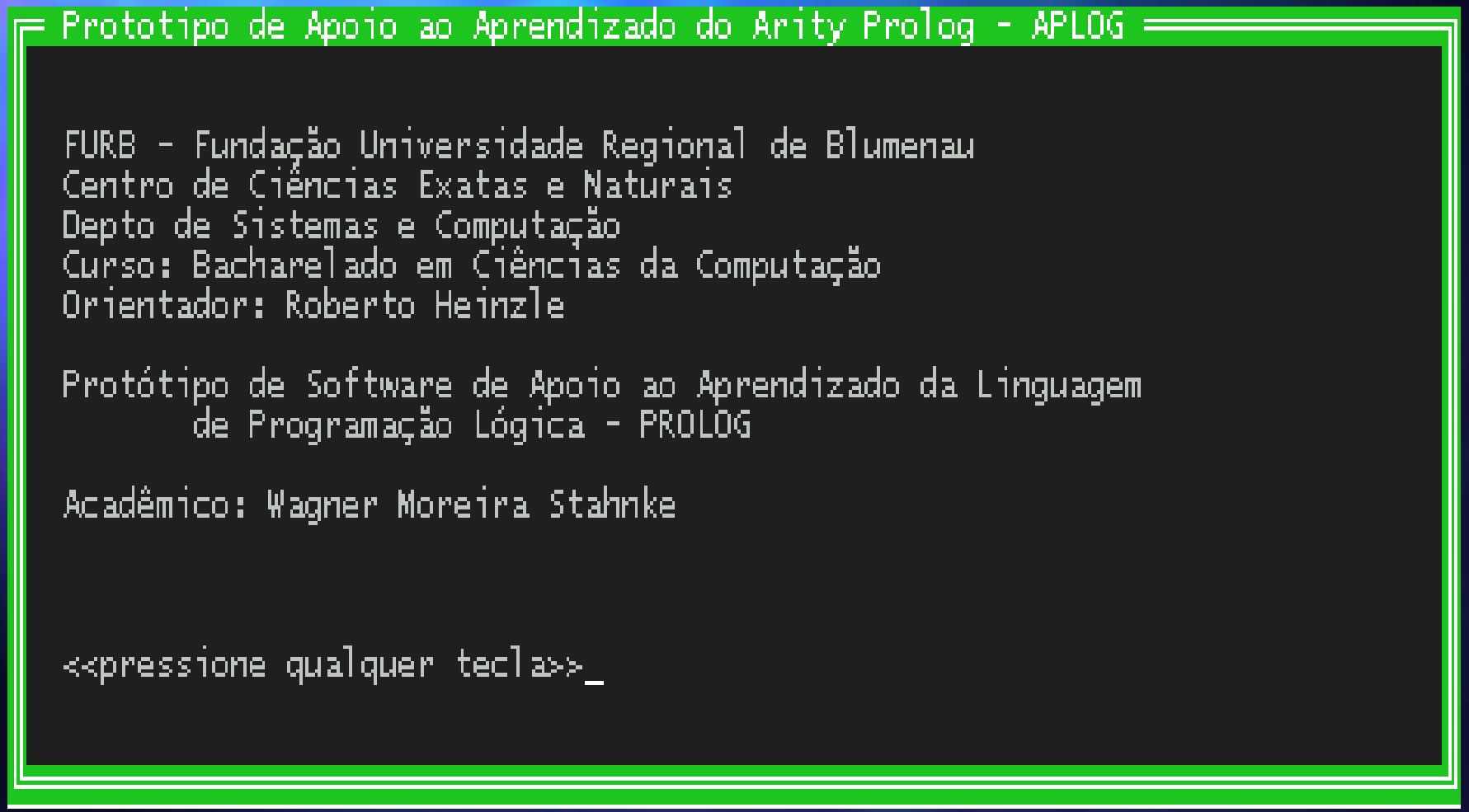
Implementação do Protótipo

- Metodologia Utilizada:
  - Prototipação
- Ferramenta Utilizada:
  - Arity Prolog32 V1.1.88

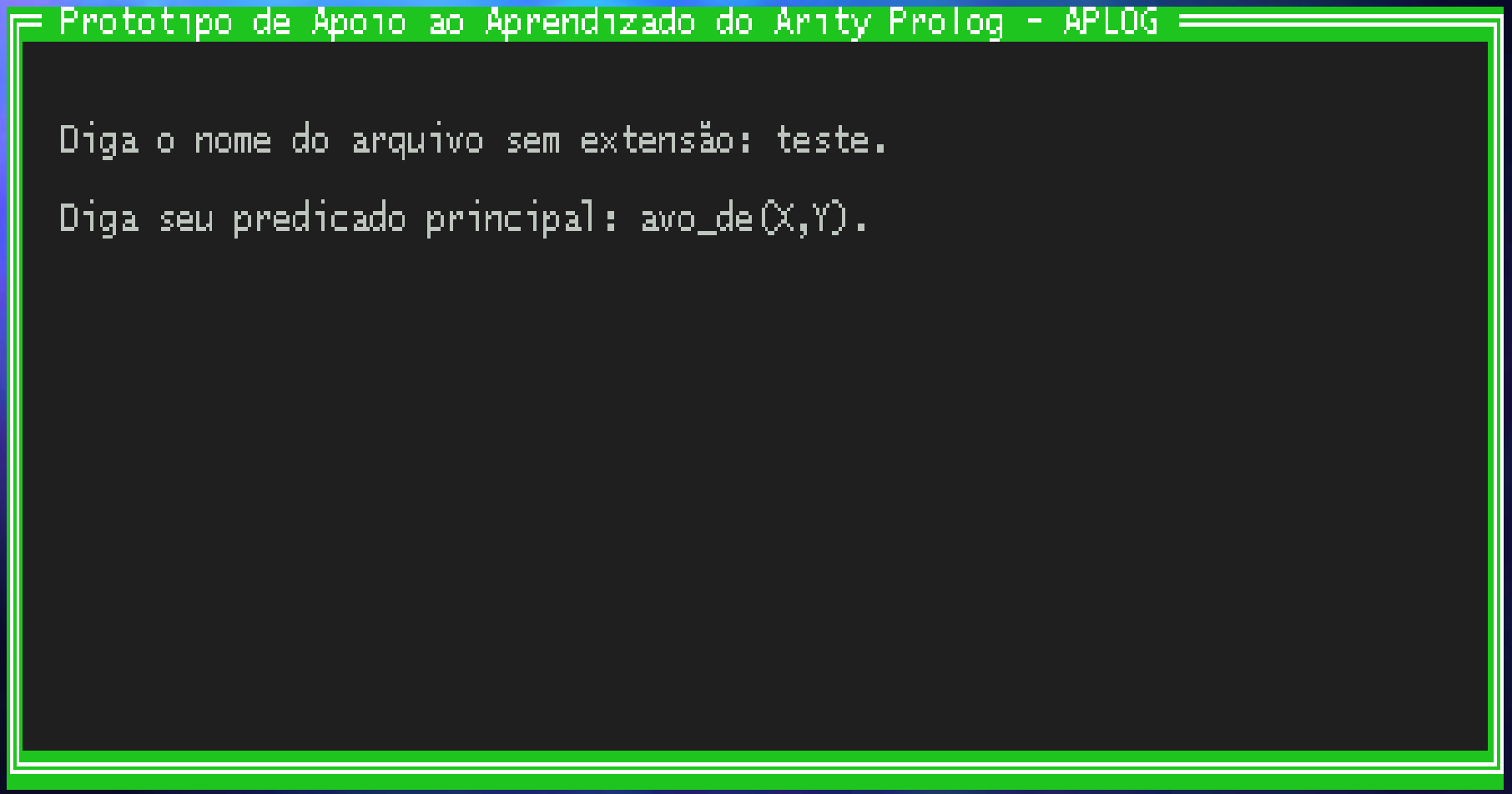
Protótipo: Operacionalidade



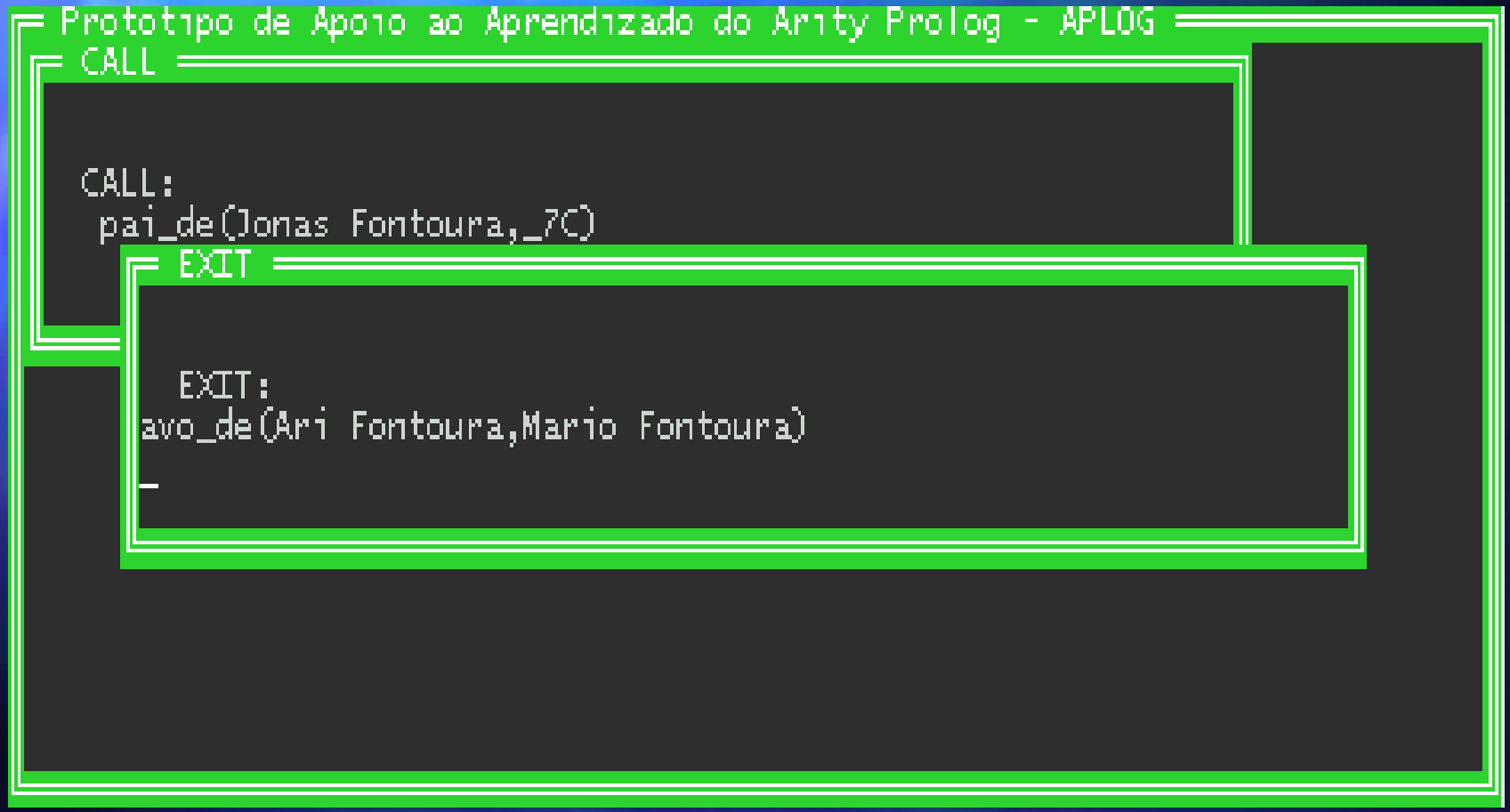
Protótipo: Operacionalidade



Protótipo: Operacionalidade



Protótipo: Operacionalidade



Conclusões

- O presente trabalho permitiu um estudo mais aprofundado da linguagem de programação Prolog, bem como alguns de seus componentes e ferramentas para construção deste protótipo aqui demonstrado
- O protótipo atendeu aos requisitos propostos inicialmente

Extensões

- Protótipo
  - Implementar: Análise Sintática, de modo a verificar se um arquivo está com o formato correto das instruções
  - Utilizar de recursos de outra linguagem de programação, de preferência visual
  - Uma melhor geração de telas