



UNIVERSIDADE ESTADUAL PAULISTA
“JÚLIO DE MESQUITA FILHO”

Relatório sobre o trabalho: Base de conhecimento dinâmica.

Docentes: *Matheus Gonçalves Ribeiro*

Discentes: *Flávia Narumi Nitto,
Miguel Donizeti da Silva e Silva,
Vitor Pelicer de Mesquita França.*

São José do Rio Preto
2021

Resumo

Foi solicitado para que fosse produzido uma base de dados dinâmica em linguagem *Prolog*, funcionando como um programa de inteligência artificial, que aprende fatos e cláusulas conforme é executado.

O mesmo possui um arquivo para “disciplina.bd”, que contém dados das disciplinas a serem/ou já cadastradas no banco de dados, como: Nome da Disciplina, Nome do docente da disciplina e departamento da disciplina; outro arquivo é usado para armazenar os dados dos docentes, como: Nome da disciplina do docente, sexo do docente, se ele é substituto ou não, seu departamento e seu nome, sendo denominado como “docentes.bd”.

Sua forma de execução se define na Anamnese (Anamnese é uma entrevista curta geralmente utilizada para conhecer alguém, principalmente na área médica), onde são executadas 4 perguntas para definir o docente, são elas: “Qual o nome da disciplina que o docente leciona?”, “Qual o sexo do docente? [F/M]”, “O docente é substituto? [s/n]” e “Qual o departamento do docente?”, onde no decorrer do programa ocorrem casamentos de fatos e cláusulas para que possa ser identificado ou não o mesmo.

```
%regras para professores (docentes)
pergunta1P(D) :- format('~n-Docente--~n'),
    format('1) Qual o nome da disciplina que o docente leciona?~nR: '),
    gets(D).

pergunta2P(S) :- format('~n2) Qual o sexo do docente? [F/M]~nR: '),
    gets(S).

pergunta3P(Sub) :- format('~n3) O docente é substituto? [s/n]~nR: '),
    get_char(Sub), get_char('\n').

pergunta4P(Dep) :- format('~n4) Qual o departamento do docente?~nR: '),
    gets(Dep).

respondeP(D,S,Sub,Dep) :- docente(D,S,Sub,Dep,N), !, format('~n~nSISTEMA: Docente
encontrado!~n-Docente: ~w.~n-Disciplina: ~w.~n', [N,D]).

respondeP(D,S,Sub,Dep) :- format('~n~nSISTEMA: Docente não encontrado!~nDeseja gra
var a resposta na base de dados do sistema? [s/n]~nR: '),
    get_char(R), get_char('\n'),
    if(R=s, nomeP(D,S,Sub,Dep), write('Dados apagados\n\n')).
```

Figura 1: Print das funções declaradas para as perguntas de docentes.

Ainda correspondente ao quesito de Anamnese, são reproduzidas 3 perguntas para o reconhecimento de disciplinas: “Qual o nome da disciplina?”, “Qual o nome do docente?”, “Qual o nome do departamento da disciplina?”. Dessa forma, se a disciplina não estiver no banco de dados, é possível que a mesma seja adicionada.

```
%regras para disciplinas
pergunta1D(D) :- format('~n-Disciplina~n'),
    format('1) Qual o nome da disciplina?~nR: '),
    gets(D).

pergunta2D(N) :- format('~n2) Qual o nome do docente?~nR: '),
    gets(N).

pergunta3D(Dep) :- format('~n3) Qual o departamento da disciplina?~nR: '),
    gets(Dep).

respondeD(D,N,Dep) :- disciplina(D,N,Dep), !, format('~n~nSISTEMA: Disciplina encontrada!~n-Docente: ~w.~n-Disciplina: ~w.~n', [N,D]).

respondeD(D,N,Dep) :- format('~n~nSISTEMA: Disciplina não encontrada!~nDeseja gravar a resposta na base de dados do sistema? [s/n]~nR: '),
    get_char(R), get_char('\n'), write('-Dados salvos!~n'),
    if(R=s, asserta(disciplina(D,N,Dep)), write('Dados apagados\n\n')).
```

Figura 2: Print das regras declaradas para identificação da disciplina.

Também criamos regras para uso geral, como *ifs*, onde estabelece comparações e dita operações conforme a condição; “continua”, que pergunta para o usuário se ele deseja continuar a usar o sistema ou não, tomando rumos diferentes conforme resposta negativa ou positiva; “salva”, usada para fazer o salvamento dos arquivos onde se encontram as bases de dados; “carrega”, onde faz o carregamento da base de dados referidas em arquivos existentes; e “gets” usada para obter informações através do teclado do usuário.

```
%regras genéricas
if(Condition,Then,_) :- Condition, !, Then.
if(_,_,Else) :- Else.

continua(R) :- format('~nDeseja continuar? [s/n]~nR: '),
    get_char(R), get_char('\n').
salva(P,A) :- tell(A), listing(P), told.
carrega(A) :- exists_file(A), consult(A); true.
gets(S) :- read_line_to_codes(user_input, D), name(S,D).
```

Figura 3: Print das regras de uso geral declaradas.