

[Painel do utilizador](#)

As minhas unidades curriculares

[Programação em Lógica](#)[Evaluation](#)[MT1 \(2020-11-28\)](#)**Início** sábado, 28 de novembro de 2020 às 08:34**Estado** Prova submetida**Data de  
submissão:** sábado, 28 de novembro de 2020 às 10:03**Tempo gasto** 1 hora 29 minutos**Nota** **19,30** de um máximo de 20,00 (**97%**)

## Informação

Considere a seguinte base de conhecimento, que representa um conjunto de jogos de futebol de um campeonato.

```
jogo(1,sporting,porto,1-2).
jogo(1,maritimo,benfica,2-0).
jogo(2,sporting,benfica,0-2).
jogo(2,porto,maritimo,1-0).
jogo(3,maritimo,sporting,1-1).
jogo(3,benfica,porto,0-2).

treinadores(porto,[[1-3]-sergio_conceicao]).
treinadores(sporting,[[1-2]-silas,[3-3]-ruben_amorim]).
treinadores(benfica,[[1-3]-bruno_lage]).
treinadores(maritimo,[[1-3]-jose_gomes]).
```

O predicado **jogo/4** contém informação relativa a jogos, incluindo o número da jornada, o nome da equipa que joga em casa, o nome da equipa visitante e o resultado:

**jogo(Jornada, EquipaCasa, EquipaVisitante, Resultado)**

O predicado **treinadores/2** contém a lista de treinadores que cada equipa teve durante o campeonato. Para cada treinador, é indicado o intervalo de jornadas onde comandou a equipa:

**treinadores(Equipas, [ [JornadaInicial-JornadaFinal]-NomeTreinador | Lista])**

Assumem-se os seguintes factos:

- Duas equipas apenas se defrontam uma única vez no campeonato.
- Um treinador treinou apenas uma equipa, e fê-lo num conjunto de jornadas consecutivas (ou seja, se sair de uma equipa, não pode voltar a treinar nenhuma equipa do campeonato).

Responda às perguntas 1 a 11 **SEM** utilizar predicados de obtenção de soluções múltiplas (*findall*, *setof* e *bagof*), e **SEM** usar qualquer biblioteca do SICStus.

## Pergunta 1

Respondida

Pontuou 1,500 de 1,500

Implemente o predicado ***n\_treinadores(?Equipa, ?Numero)*** que sucede se a equipa ***Equipa*** teve um número de treinadores igual a ***Numero***.

```
n_treinadores(Equipa, Numero) :-
    treinadores(Equipa, Treinadores),
    length(Treinadores, Numero).
```

Comentário: ['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']

## Pergunta 2

Respondida

Pontuou 1,500 de 1,500

Implemente o predicado ***n\_jornadas\_treinador(?Treinador, ?NumeroJornadas)*** que sucede se o número de jornadas que o treinador ***Treinador*** esteve a comandar uma equipa for igual a ***NumeroJornadas***.

```
treinadorInLista(Treinador, [Jornadas-Treinador | _], Jornadas).
treinadorInLista(Treinador, [_ | Rest], Jornadas) :- treinadorInLista(Treinador, Rest, Jornadas).

n_jornadas_treinador(Treinador, NumeroJornadas) :-
    treinadores(_, Treinadores),
    treinadorInLista(Treinador, Treinadores, [PrimeiraJornada-UltimaJornada]),
    NumeroJornadas is UltimaJornada - PrimeiraJornada + 1.
```

Comentário: ['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']

## Pergunta 3

Respondida

Pontuou 1,500 de 1,500

Implemente o predicado ***ganhou(?Jornada, ?EquipaVencedora, ?EquipaDerrotada)*** que sucede se na jornada ***Jornada*** a equipa ***EquipaVencedora*** venceu a equipa ***EquipaDerrotada***.

```
ganhou(Jornada, EquipaVencedora, EquipaDerrotada) :-
    jogo(Jornada, EquipaVencedora, EquipaDerrotada, GolosCasa-GolosFora),
    GolosCasa > GolosFora.

ganhou(Jornada, EquipaVencedora, EquipaDerrotada) :-
    jogo(Jornada, EquipaDerrotada, EquipaVencedora, GolosCasa-GolosFora),
    GolosCasa < GolosFora.
```

Comentário: ['O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O']



## Informação

Pretende-se definir operadores que permitam escrever factos no seguinte formato:

```
| ?- o benfica venceu o sporting.  
yes  
|?- o porto venceu o X.  
X = maritimo ? ;  
X = sporting ? ;  
X = benfica ? ;  
no
```



## Pergunta 4

Correta Pontuou 0,500 de 0,500

Qual a forma mais correta de definir o operador "o" em termos sintáticos e semânticos?

Selecione uma opção de resposta:

- ☐ a. :- op(150, xfy, o).
- ☐ b. :- op(140, yf, o).
- ☒ c. :- op(180, fx, o). 
- ☐ d. [Não Responder]
- ☐ e. :- op(160, xfx, o).
- ☐ f. :- op(170, fy, o).

A sua resposta está correta.



## Pergunta 5

Correta Pontuou 0,500 de 0,500

Qual a forma mais correta de definir o operador "**venceu**" em termos sintáticos e semânticos?

Selecione uma opção de resposta:

☐ a. [Não Responder]

☐ b. :- op(200, fx, venceu).

☐ c. :- op(100, xfy, venceu).

☐ d. :- op(1200, xfy, venceu).

☒ e. :- op(200, xfx, venceu).

☐ f. :- op(100, xfx, venceu).

A sua resposta está correta.

## Pergunta 6

Respondida Pontuou 0,500 de 0,500

Implemente uma regra que permita em Prolog escrever frases em linguagem natural como as apresentadas acima.

o X :- treinadores(X, \_).  
o X venceu o Y :- ganhou(\_, X, Y).

Comentário:

['O', 'O', 'O', 'O', 'O', 'O']

Correto, não precisas da regra para o "o".

## Pergunta 7

Respondida

Pontuou 1,800 de 2,000

Considere o predicado ***predX(?N, +A, +B)***, definido da seguinte forma:

```
predX(N,N,_).
predX(N,A,B):-
    !,
    A \= B,
    A1 is A + sign(B - A),
    predX(N,A1,B).
```

a) Descreva sucintamente o que faz o predicado ***predX(?N, +A, +B)***.

b) O cut presente no corpo da regra é verde ou vermelho? Justifique brevemente.

- a) Se N for enviado como argumento, o predicado permite saber se N se encontra dentro do intervalo [A, B]. Se N for o valor de retorno, N tem o valor de A inicialmente, e permite obter todos os valores até B, aproximando-se de B a cada Redo, sendo a última resposta N = B.
- b) É um cut verde pois a sua presença em nada altera as soluções obtidas pelo predicado. Apenas serve para evitar backtracking desnecessário.

Comentário:

- + Explica corretamente para o caso onde N está instanciado.
- + Explica corretamente para o caso onde N não está instanciado.
- + Classifica corretamente o cut
- + Justifica corretamente que o cut é verde.
- Fala que este cut aumenta a eficiência / poda a pesquisa.

## Pergunta 8

Respondida

Pontuou 2,000 de 2,000

Implemente o predicado ***treinador\_bom(?Treinador)*** que sucede se o treinador nunca perdeu um jogo ao comando da sua equipa.

```
treinadorInLista(Treinador, [Jornadas-Treinador | _], Jornadas).
treinadorInLista(Treinador, [_ | Rest], Jornadas) :- treinadorInLista(Treinador, Rest, Jornadas).

treinador_bom(Treinador) :-
    treinadores(Equipa, Treinadores),
    treinadorInLista(Treinador, Treinadores, [PrimeiraJornada-UltimaJornada]), !,
    \+ (
        predX(Jornada, PrimeiraJornada, UltimaJornada),
        \+ ganhou(Jornada, Equipa, _)
    ).
```

Comentário: ['X', 'X', 'O', 'X']

## Informação

Considere os predicados *imprime\_totobola/2* e *imprime\_texto/2*, que relacionam um inteiro que codifica o resultado de um jogo com um átomo.

```
imprime_totobola(1, '1').
imprime_totobola(0, 'X').
imprime_totobola(-1, '2').

imprime_texto(X, 'vitoria da casa'):-
    X = 1.
imprime_texto(X, 'empate'):-
    X = 0.
imprime_texto(X, 'derrota da casa'):-
    X = -1.
```

O primeiro argumento pode ter um de três valores:

- 1 se a equipa da casa venceu
- 0 se houve um empate
- -1 se a equipa da casa perdeu (ou seja, a equipa visitante ganhou)

## Pergunta 9

Respondida

Pontuou 2,000 de 2,000

Implemente o predicado ***imprime\_jogos(+F)*** que recebe como argumento o nome de um predicado ***F*** de aridade 2 e cujos argumentos têm a mesma semântica que ***imprime\_totobola/2*** e ***imprime\_texto/2***.

O predicado ***imprime*** os resultados dos jogos no ecrã, em linhas da forma **<Jornada J: EquipaCasa x EquipaVisitante - Output>**, onde **J** é o número da jornada, **EquipaCasa** é a equipa da casa, **EquipaVisitante** é a equipa visitante e **Output** é o átomo devolvido no segundo argumento do predicado ***F*** em função do resultado do jogo.

Os jogos devem ser impressos na mesma ordem pela qual aparecem na base de conhecimento (onde já estão ordenados pelo número da jornada). O predicado deve sempre suceder.

```
| ?- imprime_jogos(imprime_totobola).
```

```
Jornada 1: sporting x porto - 2
```

```
Jornada 1: maritimo x benfica - 1
```

```
Jornada 2: sporting x benfica - 2
```

```
Jornada 2: porto x maritimo - 1
```

```
Jornada 3: maritimo x sporting - X
```

```
Jornada 3: benfica x porto - 2
```

```
yes
```

```
| ?- imprime_jogos(imprime_texto).
```

```
Jornada 1: sporting x porto - derrota da casa
```

```
Jornada 1: maritimo x benfica - vitoria da casa
```

```
Jornada 2: sporting x benfica - derrota da casa
```

```
Jornada 2: porto x maritimo - vitoria da casa
```

```
Jornada 3: maritimo x sporting - empate
```

```
Jornada 3: benfica x porto - derrota da casa
```

```
yes
```

```
obterResultado(Jornada, EquipaCasa, EquipaFora, 1) :-
```

```
    ganhou(Jornada, EquipaCasa, EquipaFora), !.
```

```
obterResultado(Jornada, EquipaCasa, EquipaFora, -1) :-
```

```
    ganhou(Jornada, EquipaFora, EquipaCasa), !.
```

```
obterResultado(_, _, 0) :- !.
```

```
imprime_jogos(F) :-
```

```
    jogo(Jornada, EquipaCasa, EquipaFora, _),
```

```
    obterResultado(Jornada, EquipaCasa, EquipaFora, Resultado),
```

```
    format('Jornada ~d: ~s x ~s - ', [Jornada, EquipaCasa, EquipaFora]),
```

```
    X =.. [F, Resultado, S], X, write(S), nl,
```

```
    fail.
```

```
imprime_jogos(_).
```

Comentário:

## Pergunta 10

Incorreta

Pontuou 0,000 de 0,500

Qual das seguintes afirmações está correta?

Selecione uma opção de resposta:

- ☐ a. Os predicados ***imprime\_totobola/2*** e ***imprime\_texto/2*** são igualmente eficientes, pois efetuam a mesma quantidade de unificações quando executados.
- ☒ b. O predicado ***imprime\_totobola/2*** é mais eficiente que ***imprime\_texto/2***, pois tira proveito do mecanismo de indexação de predicados do SICStus, que é feito pelo functor e todos os argumentos. ✗
- ☐ c. [Não Responder]
- ☐ d. O predicado ***imprime\_texto/2*** é mais eficiente que ***imprime\_totobola/2***, pois tira proveito do mecanismo de indexação de predicados do SICStus, que é feito pelo functor e pelo conjunto de goals da forma "***X = Y***" presentes no corpo da regra.
- ☐ e. O predicado ***imprime\_totobola/2*** é mais eficiente que ***imprime\_texto/2***, pois tira proveito do mecanismo de indexação de predicados do SICStus, que é feito apenas pelo functor e primeiro argumento.
- ☐ f. Os predicados ***imprime\_totobola/2*** e ***imprime\_texto/2*** são igualmente eficientes, pois ambos tiram proveito de igual forma do mecanismo de indexação de predicados do SICStus, que é feito apenas pelo functor.

A sua resposta está incorreta.

## Pergunta 11

Correta

Pontuou 0,500 de 0,500

Considere um predicado ***shutdown/0*** que, quando é invocado, faz com que a máquina que está a correr o SICStus Prolog se desligue. Considere também os predicados ***=../2*** (operador "univ") e ***imprime\_jogos/1***, apresentado anteriormente.

Entre os predicados ***shutdown/0***, ***=../2*** e ***imprime\_jogos/1***, qual/quais são predicados extra-lógicos?

Selecione uma opção de resposta:

- ☐ a. [Não Responder]
- ☐ b. Os três predicados são extra-lógicos.
- ☐ c. Nenhum dos três predicados é extra-lógico.
- ☐ d. Apenas ***imprime\_jogos/1*** é extra-lógico.
- ☒ e. Apenas ***shutdown/0*** e ***imprime\_jogos/1*** são extra-lógicos. ✓
- ☐ f. Apenas ***=../2*** e ***imprime\_jogos/1*** são extra-lógicos.

A sua resposta está correta.

## Informação

Nas perguntas seguintes, pode fazer uso de predicados de obtenção de múltiplas soluções (*findall*, *setof* e *bagof*) e das bibliotecas do SICStus (com a exceção de *clpfd* e *clpb*).



## Pergunta 12

Respondida

Pontuou 2,000 de 2,000

Implemente o predicado **lista\_treinadores(?L)** que unifica a lista **L** com o conjunto de treinadores que comandaram alguma equipa durante alguma fase do campeonato.

```

treinadorInLista(Treinador, [Jornadas-Treinador | _], Jornadas).
treinadorInLista(Treinador, [_ | Rest], Jornadas) :- treinadorInLista(Treinador, Rest, Jornadas).

lista_treinadores(L) :-
    findall(
        Treinador,
        (
            treinadores(_, Treinadores),
            treinadorInLista(Treinador, Treinadores, _)
        ),
        L
    ), !.

```

Comentário: ['O']

## Pergunta 13

Respondida

Pontuou 2,000 de 2,000

Implemente o predicado **duracao\_treinadores(?L)** que devolve em **L** a lista de treinadores e respetivo número de jornadas em comando de uma equipa, ordenada por ordem decrescente segundo este número. Cada elemento é da forma **NumeroDeJornadas-Treinador**.

Caso dois treinadores tenham trabalhado durante o mesmo número de jornadas, poderá devolvê-los em qualquer ordem na lista.

```

| ?- duracao_treinadores(L).
L = [3-sergio_conceicao,3-jose_gomes,3-bruno_lage,2-silas,1-ruben_amorim] ? ;
no

```

```

:- use_module(library(lists)).

duracao_treinadores(L) :-
    findall(
        N-Treinador,
        (
            n_jornadas_treinador(Treinador, N)
        ),
        Treinadores
    ),
    sort(Treinadores, Sorted),
    reverse(Sorted, L), !.

```

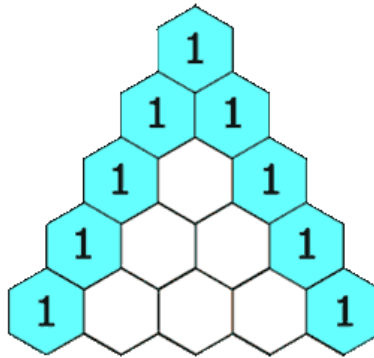
Comentário: ['O', 'O']

## Pergunta 14

Respondida Pontuou 3,000 de 3,000

Implemente o predicado **pascal(+N, -L)** que devolve na lista **L** a **N**-ésima linha do triângulo de Pascal.

Cada número do triângulo de Pascal é igual à soma dos números imediatamente acima como exemplificado na seguinte figura.



Será valorizada uma implementação eficiente, isto é, que obtenha a solução para dimensões grandes (p. ex.,  $N = 1000$ ) em menos de 1 segundo.

```
| ?- pascal(1, L).
L = [1] ? ;
no
| ?- pascal(2, L).
L = [1,1] ? ;
no
| ?- pascal(3, L).
L = [1,2,1] ? ;
no
| ?- pascal(4, L).
L = [1,3,3,1] ? ;
no
| ?- pascal(5, L).
L = [1,4,6,4,1] ? ;
no
```

```
pascal(1, [1]) :- !.
pascal(N, L) :-
    N1 is N - 1,
    pascal(N1, L1),
    append([0], L1, L2),
    append(L2, [0], Aux),
    aux_pascal([], L, Aux), !.

aux_pascal(L, L, _ | []).
aux_pascal(Acc, L, [Elem1, Elem2 | Rest]) :-
    NewElem is Elem1 + Elem2,
    append(Acc, [NewElem], NewAcc),
    aux_pascal(NewAcc, L, [Elem2 | Rest]).
```

Comentário: ['O', 'O', 'O', 'O']

◀ TP2 - Entrega Final

Ir para...

MT1 - Notas ▶



