

Número:	Nome:	Turno:
---------	-------	--------

Para a resolução desta ficha apenas está permitida a seguinte utilização do módulo `lists`.

```
:- use_module(library(lists),[append/3, member/2]).
```

Se necessitar de outros predicados definidos neste módulo terá que os implementar.

Questão 1: Considere a representação de conjuntos em listas sem repetições. Defina o predicado `diferenca(+C1,+C2,-C3)` que sucede quando o conjunto `C3` é a diferença do conjunto `C1` com o conjunto `C2`. (Ou seja, $C3 = C1 \setminus C2$)

Questão 2: Defina o predicado `prods(+L,-P1,-P2)` que sucede se `P1` é o produtório dos elementos positivos da lista `L` e `P2` é o produtório dos seus elementos negativos. A implementação deste predicado só deve fazer uma travessia da lista. (Os produtórios da lista vazia deverão ser 1.)

Questão 3: Defina o predicado `drop(+N, ?L1, ?L2)` que sucede se a lista `L2` é o resultado de retirar da lista `L1` os seus `N` primeiros elementos. Se o comprimento da lista `L1` for menor do que `N`, `L2` deverá ser igual a lista vazia. Se `N` for um número negativo o predicado deve falhar.

Questão 4: Assuma que a informação referente aos horários das salas de aula está guardada na base de conhecimento, em factos com o seguinte formato:

`sala(número, hora de início, hora de fim, diciplina, docente) .`

a) Defina o predicado `ocupada(+NSala, +Hora)` que sucede se a sala número `NSala` está ocupada à hora `Hora`.

b) Defina o predicado `total_doc(+NSala, +Doc, -NHoras)` que sucede se `NHoras` é o número de horas de aulas que o docente `Doc` tem na sala `NSala`.