

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 `uniao(+C1,+C2,-C3)` que sucede quando o conjunto `C3` é a união do conjunto `C1` com o conjunto `C2`.

**Questão 2:** Defina o predicado `maior_menor(+L,-M1,-M2)` que sucede se `M1` é o maior elemento da lista `L` e `M2` é o menor elemento da lista `L`. A implementação deste predicado só deve fazer uma travessia da lista.

**Questão 3:** Defina o predicado `zip(?L1,?L2,?L3)` que sucede se a lista `L3` resulta do emparelhamento das listas `L1` e `L2`. Ou seja, `L3` é uma lista de pares, tal que na posição  $i$  está o par com os elementos que estão também na posição  $i$  das listas `L1` e `L2`.

**Questão 4:** Assuma que a informação referente aos horários dos cursos está guardada na base de conhecimento, em factos com o seguinte formato:

`horario( curso, dia da semana, disciplina, tipo, hora de início, hora de fim, docente ).`

a) Defina o predicado `lecciona(?Doc,?Curso)` que sucede se o docente `Doc` dá aulas ao curso `Curso`.

b) Defina o predicado `num_discip(+Doc,+Curso,-NDiscip)` que sucede se `NDiscip` é o número de disciplinas distintas que o docente `Doc` lecciona ao curso `Curso`. Se o docente leccionar aulas T,TP e/ou P da mesma disciplina, isso só deve contar como uma disciplina. Se o `Doc` não der aulas ao `Curso`, `NDiscip` deverá ser igual a 0.