

```

<html><head></head><body><pre style="word-wrap: break-word; white-space: pre-wrap;">
import Sintaxis
-----
-- Cláusulas -----
-----
-- Definimos el tipo de datos Cláusula como una par (pos, neg), donde
-- pos es la lista de los átomos correspondientes a los literales
-- positivos de la cláusula y neg la lista de los átomos
-- correspondientes a los literales negativos de la cláusula.
-----

type Clausula = ([Prop],[Prop])

-----
-- Ejercicio 1: (3 puntos) Definir la función
-- esModeloClausula :: Interpretacion -> Clausula -> Bool
-- tal que (esModeloClausula i c) se verifica si i es modelo de c. Por
-- ejemplo,
-- esModeloClausula [p,r] ([p, q],[ ]) ==> True
-- esModeloClausula [r] ([p], [q]) ==> True
-- esModeloClausula [q,r] ([p], [q]) ==> False
-- esModeloClausula [q,r] ([ ],[ ]) ==> False
-- esModeloClausula [q,r] ([ ],[q]) ==> False
-- esModeloClausula [r] ([ ],[q]) ==> True
-----

esModeloAtom :: Interpretacion -> Prop -> Bool
esModeloAtom i (Atom s) = elem (Atom s) i

esModeloClausula :: Interpretacion -> Clausula -> Bool
esModeloClausula i (pos,neg)
  | null pos && null neg = False
  | otherwise           = or [esModeloAtom i a | a <- pos] ||
                          and [not (esModeloAtom i a) | a <- neg]

</pre></body></html>

```



CLASES PARTICULARES, TUTORÍAS TÉCNICAS ONLINE
 LLAMA O ENVÍA WHATSAPP: 689 45 44 70

ONLINE PRIVATE LESSONS FOR SCIENCE STUDENTS
 CALL OR WHATSAPP:689 45 44 70