

```
<html><head></head><body><pre style="word-wrap: break-word; white-space: pre-wrap;">-- LMF
(3º del Grado en Matemáticas)
-- 3º examen de evaluación continua (6 de junio de 2014)
-- -----

-- Nombre:

-- Apellidos:

import Sintaxis

-- -----
-- Ejercicio 1: Definir la función
--   esModeloClausula :: Interpretación -> Cláusula -> Bool
-- tal que (esModeloClausula i c) se verifica si i es modelo de c . Por
-- ejemplo,
--   esModeloClausula [(p,True),(q,False),(r,True)] [p, q]      ==> True
--   esModeloClausula [(p,False),(q,False),(r,True)] [p, no q] ==> True
--   esModeloClausula [(p,False),(q,True),(r,True)] [p, no q]  ==> False
--   esModeloClausula [(p,False),(q,True),(r,True)] []          ==> False
--   esModeloClausula [(p,False),(q,True),(r,True)] [no q]      ==> False
--   esModeloClausula [(p,False),(q,False),(r,True)] [no q]     ==> True
-- -----

esModeloLiteral :: Interpretación -> Literal -> Bool
esModeloLiteral i (Atom s)      = busca (Atom s) i
esModeloLiteral i (Neg (Atom s)) = not (esModeloLiteral i (Atom s))

busca :: Eq t1 => t1 -> [(t1, t)] -> t
busca p i = head [v | (a,v) <- i, a == p]

esModeloClausula :: Interpretación -> Clausula -> Bool
esModeloClausula i c = or [esModeloLiteral i l | l <- c]

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