<html><head></head><body><pre style="word-wrap: break-word; white-space: pre-wrap;">
```
import Sintaxis
-- ------------------------------------------------------------------
-- ClÃ¡usulas                                                      --
-- ------------------------------------------------------------------

-- ------------------------------------------------------------------
-- Definimos el tipo de datos ClÃ¡usula como una par (pos, neg), donde
-- pos es la lista de los Ã¡tomos correspondientes a los literales
-- positivos de la clÃ¡usula y neg la lista de los atomos
-- correspondientes a los literales negativos de la clÃ¡usula.
-- ------------------------------------------------------------------

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

-- ------------------------------------------------------------------
-- Ejercicio 1: (3 puntos) Definir la funciÃ³n
--    esModeloClausula :: Interpretacion -&gt; Clausula -&gt; Bool
-- tal que (esModeloClausula i c) se verifica si i es modelo de c. Por
-- ejemplo,
--    esModeloClausula [p,r] ([p, q],[]) ==&gt;  True
--    esModeloClausula [r] ([p], [q])    ==&gt;  True
--    esModeloClausula [q,r] ([p], [q])  ==&gt;  False
--    esModeloClausula [q,r] ([],[])     ==&gt;  False
--    esModeloClausula [q,r] ([],[q])    ==&gt;  False
--    esModeloClausula [r] ([],[q])      ==&gt;  True
-- ------------------------------------------------------------------

esModeloAtom :: Interpretacion -&gt; Prop -&gt; Bool
esModeloAtom i (Atom s) = elem (Atom s) i

esModeloClausula :: Interpretacion -&gt; Clausula -&gt; Bool
esModeloClausula i (pos,neg)
    | null pos &amp;&amp; null neg = False
    | otherwise            = or [esModeloAtom i a | a &lt;- pos] ||
                             and [not (esModeloAtom i a) | a &lt;- neg]
```

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