

## Sobre el proyecto Principia

Principia = pasos intermedios. Esa es la esencia.

El proyecto Principia pretende crear un programa capaz de resolver problemas de lógica proposicional y de predicados, extendiéndose más adelante hacia otros problemas matemáticos. Pero el objetivo no es que el ordenador resuelva problemas (para eso hay otros programas), sino mostrar un camino para llegar a esa solución: es un programa educativo. Podría definirse como la versión interactiva de los libros de problemas resueltos.

En particular, un programa así es muy necesario en la educación a distancia, donde el contacto con el profesor no es diario y se espera un aprendizaje bastante autónomo.

- Escrito en [Haskell](#)
- Liberado bajo [licencia BSD](#) (de 3 clausulas)
- [Enlace a la forja](#) (alojada en RedIris)



Proyecto participante en el tercer Concurso Universitario de Software Libre

### Banner de proyecto participante

[Bajar en SVG](#)

**Fuente:** [Expressway Free](#) de [Typodermic](#).  
[Jose Fernandez](#) de [Dafont](#).

Banner bajo licencia "[Reconocimiento-Compartir bajo la misma licencia 2.5 España](#)"  
(click para leer)

## Requisitos

- Tener instalado [Hugs](#) o [GHC](#) (intérpretes<sup>1</sup> del lenguaje Haskell).

## Uso

1. Leer el archivo que contenga el tipo de ejercicio que se quiere resolver.
  - Utilizando el interprete de comandos, ir al directorio donde se encuentre Principia e invocar a Hugs o a GHCi mediante los comandos "*hugs*" o "*ghci*" respectivamente.
  - Leer el archivo mediante el comando "*:load*". Por ejemplo, para cargar Proposicional.hs, "*:load Proposicional*".
2. Seguir las instrucciones dadas en el apartado del tipo de ejercicio correspondiente.

---

1 GHC es un compilador, pero incorpora el intérprete GHCi.

## Lógica proposicional

Los ejercicios de lógica proposicional que Principia incluye se encuentran en el archivo "Proposicional.hs".

Las expresiones se escriben con el formato,

```
<Expresion > ::= <Variable> | <Operacion>
<Variable> ::= V <nombre>
<Operacion> ::= O <Operador> “[<Expresion>{“,<Expresion>}”]”
<Operador> ::= NOT | AND | OR | SI | SII
```

donde <nombre> es el nombre de variable entre comillas.

Operadores:

- NOT           negación                   (monario)
- AND           conjunción                   (binario)
- OR            disyunción inclusiva       (binario)
- SI            condicional                   (binario)
- SII           bicondicional                  (binario)

Por el momento, el programa no verifica que el numero de operandos introducido sea correcto. En el futuro se facilitará la escritura de expresiones lógicas.

Ejemplos de expresiones correctas:

```
O Si [V "p", V "q"]                    si p entonces q
O Sii [V "p", O NOT [V "q"]]         p si y solo si no q
O NOT[ O OR[ V "p", V "q" ] ]       no (p o q)
```

## Tipos de ejercicio

### **Forma clausulada**

Transforma una expresión cualquiera en su forma clausulada.

Para solucionar este tipo de ejercicio, debe llamarse a la función sClausulada.

Ejemplo:

```
Proposicional> sClausulada( O NOT[O OR[ V "p", O NOT[V "q"] ] ] )
NOT OR p NOT q (dato)
NOT OR p NOT q, {(NOT OR p q) -> (AND NOT p NOT q)} => AND NOT p NOT NOT q
NOT NOT q, {(NOT NOT p) -> (p)} => q
AND NOT p q
```

## Ejercicios no incluidos en Principia

Para definir un nuevo tipo de ejercicio, es necesario crear un módulo con las transformaciones que sobre una expresión (dato) hay que hacer.

### **Definir tipo de ejercicio de lógica proposicional**

Utilizar el archivo Plantilla.hs como plantilla. Se trata de un módulo que utiliza los

## Principia alfa1 - Documentación para el usuario final

operadores definidos en el módulo Proposicional.

Principia está diseñado de manera que **cualquier tipo de ejercicio de lógica proposicional** que el usuario cree para ésta versión, **seguirá funcionando con posteriores versiones.**

Si crea un ejercicio nuevo, considere la posibilidad de compartirlo en el foro “contribuciones” del proyecto principia:

[https://forja.rediris.es/forum/forum.php?forum\\_id=1594](https://forja.rediris.es/forum/forum.php?forum_id=1594) El foro no dispone de la posibilidad de subir ficheros, simplemente copie y pegue el código en el mensaje que publique; incluso si la indentación no se viera correctamente al hacerlo así.

### ***Definir otro tipo de ejercicio***

El usuario con conocimientos de Haskell que desee definir un tipo de ejercicio que no utilice los operadores definidos en Proposicional, debe crear un módulo declarando otros operadores y utilizando el tipo Expression y las funciones facilitadas en el módulo Expression. Puede consultarse la documentación para desarrolladores para tener más información sobre éste módulo.

No obstante, no puede garantizarse que los tipos de ejercicio que se creen de esta manera sigan funcionando en versiones posteriores de Principia.