



Universidad  
Rey Juan Carlos

ESCUELA TÉCNICA SUPERIOR  
DE INGENIERÍA INFORMÁTICA

**Manual**  
**DeduccionNatural.pl**

LÓGICA  
CURSO 2022-2023

**Autores: Iván Ramírez**  
**Joaquín Arias**



Copyright (c) 2022 Iván Ramírez, Joaquín Arias. Esta obra está bajo la licencia CC BY-SA 4.0, [Creative Commons Atribución-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).

# 1. Introducción

Este manual se enmarca en el contexto de la asignatura de Lógica en la Escuela Técnica Superior de Ingeniería Informática de la Universidad Rey Juan Carlos.

En concreto se centra en el aprendizaje de la Deducción Natural, introducida por Gentzen, 1935. La Deducción Natural busca capturar la manera en que las personas razonan naturalmente al construir demostraciones matemáticas. En vez de contar con unos pocos axiomas a los que se aplican unas pocas reglas de inferencia, la deducción natural propone vaciar la lista de axiomas y ampliar la de reglas de inferencia, introduciendo dos reglas para cada constante lógica ( $\wedge$ ,  $\vee$ ,  $\neg$ ,  $\rightarrow$ ,  $\leftrightarrow$ ):

- Una para introducirla.
- Otra para eliminarla.

Por lo tanto una demostración se construye partiendo de supuestos y aplicando las reglas para llegar a la conclusión deseada. Sirve para demostrar la validez de un argumento.

Para facilitar el aprendizaje hemos desarrollado una herramienta usando programación lógica (disponible en <https://github.com/Xuaco/DeduccionNatural>) que permite evaluar si dada unas premisas y un consecuente, la demostración planteada por el alumno es correcta. La fig. 1 muestra una captura de la ejecución, en el Playground de Ciao<sup>1</sup>, del ejemplo 1.

**Ejemplo 1** *Ejemplo extraído de los apuntes de Arias, 2022.*

*Desarrollo con notación lógica:*

$T[s \wedge (p \vee q), p \rightarrow \neg r, q \rightarrow \neg r] \vdash s \wedge \neg r$

- |    |                        |                     |
|----|------------------------|---------------------|
| 1. | $s \wedge (p \vee q)$  | premisa             |
| 2. | $p \vee q$             | $E_{\wedge}(1)$     |
| 3. | $p \rightarrow \neg r$ | premisa             |
| 4. | $q \rightarrow \neg r$ | premisa             |
| 5. | $\neg r$               | $E_{\vee}(2, 3, 4)$ |
| 6. | $s$                    | $E_{\wedge}(1)$     |
| 7. | $s \wedge \neg r$      | $I_{\wedge}(5, 6)$  |

*Traducción en `DeduccionNatural.pl`:* 

```
ejemplo1 :-  
    main([ s and p or q,  
          p -> ! r,  
          q -> ! r      ],  
          s and ! r,  
          [ 'Premisa'(1),  
            'E' and b(1),  
            'Premisa'(2),  
            'Premisa'(3),  
            'E' or (2,3,4),  
            'E' and a(1),  
            'I' and (6,5)  ]).
```

A continuación, la Sección 2 describe como se han traducido los operadores y las reglas de inferencia. La Sección 3 explica como ejecutar el programa para comprobar si una demostración es correcta dado unas premisas y un consecuente. Finalmente, la Sección 4 muestra como definir reglas derivadas (o de usuario) para factorizar las demostraciones.

<sup>1</sup>Ciao Prolog es un interprete de Prolog, disponible en <https://ciao-lang.org/>, que cuenta con un Playground online en <https://ciao-lang.org/playground/index.html>

```
1 %~~~~~ Copyright (C)2022 Joaquin Arias (URJC) ~~~~~
2 %
3 % Name: deduccion.pl
4 % Author: Joaquin Arias
5 % Date: 15 August 2022
6 % Purpose: Execute Natural Deduction Proofs
7 %~~~~~
8
9 :- module(_,_).
10
11 :- op(200, fy, []).
12 :- op(400, xfy, [and, or, not, !]).
13 :- op(600, xfy, [=>, <=>]).
14
15 :- data counter/1, formula/2, tabular/1, cerrado/1, hay_susuesto/1, probar/1.
16
17 %% Ejemplos
18 ejemplo1 :-
19     main([s and p or q, p => ! r, q => ! r], s and ! r, ['Premisa'(1), 'E' and b(
20
21 ejemplo2 :-
22     main([p => q and !q], p, ['Premisa'(1), 'I' ! (1), 'E' ! (2)]).
23
24 ejemplo3 :-
25     main([p => !r, !r=>q, p], q, ['Premisa'(1), 'Premisa'(3), 'E' => (1,2), 'Premisa'(2
26
27 ejemplo4 :-
28     main([p => q, q=>r], p=>r, ['Premisa'(1), 'Premisa'(2), 'Supuesto'(p), 'E' => (1,
29
30 ejemploMT :-
31     main([r => (q and s), !(q and s)], !r, ['Premisa'(1), 'Premisa'(2), 'MT'(1,2)]
32
33 ejemploSupuesto :- %% Falla porque no esta cerrado el supuesto :-
34     main([s and p or q, p => ! r, q => ! r], s and ! r, ['Premisa'(1), 'E' and b(
35
36
37 main(Premisas, Deduccion, ReglasInferencia) :-
38     retractall(counter(_)), retractall(formula(_,_)), retractall(tabular(_)), retract
39     assert(counter(0)), assert(tabular(0)),
40     format('~p ~~~~\n', [Premisas, Deduccion]),
41     eval(Premisas, Deduccion, ReglasInferencia),
42     probar_pendientes.
43
44 probar_pendientes :-
45     setof(Nombre, probar(Nombre, Pendientes), L,
46     probar_pendientes_(Pendientes).
47 probar_pendientes.
```

```
?- use_module('/draft.pl').
yes
?- ejemplo1.
T[s and p or q, p=>!r, q=>!r] |- s and !r
1 s and p or q Premisa(1)
2 p or q E and b(1)
3 p=>!r Premisa(2)
4 q=>!r Premisa(3)
5 !r E or(2,3,4)
6 s E and b(1)
7 s and !r I and(6,5)
ok
yes
?-
```

Figura 1: Captura de pantalla con la ejecución de `DeduccionNatural.pl`

2. Constantes lógicas y reglas de inferencia
3. Ejecución del programa
4. Definir reglas derivadas

## Referencias

Arias, Joaquín (2022). **Lógica: desde Aristóteles hasta Prolog**. Madrid: Servicio de Publicaciones de la Universidad Rey Juan Carlos. ISBN: 978-84-09-38265-1.

Gentzen, Gerhard (1935). **Untersuchungen über das logische schließen. I**. En: *Mathematische zeitschrift* 35.