

Enseñanza del sistema de Deducción Natural en Lógica Proposicional aplicando programación lógica

BANCO DE BUENAS PRÁCTICAS DOCENTES

JOAQUÍN ARIAS

IVÁN RAMÍREZ



Universidad
Rey Juan Carlos



CENTRO DE INNOVACIÓN DOCENTE Y EDUCACIÓN DIGITAL
Universidad Rey Juan Carlos

I. La práctica

- **Título:** Enseñanza del sistema de Deducción Natural en Lógica Proposicional aplicando programación lógica.
- **Curso Académico:** 2023/2024 y 2023/2024
- **Asignatura:** Lógica
- **Área/Titulación:** Ingenierías y Arquitectura
- **Grupo de Estudiantes:** Grado en Inteligencia Artificial (Móstoles) y Grado en Ingeniería de la Ciberseguridad (Móstoles)
- **Palabras clave:** Lógica; Metodología Docente; Prolog; Programación; Deducción Natural

La lógica matemática y computacional son materias básicas en la mayoría de las titulaciones universitarias relacionadas con la Ingeniería Informática. Son fundamentales en el estudio de bases de datos, complejidad computacional, lenguajes de programación, inteligencia artificial, diseño y verificación de sistemas hard y soft, entre otros. Sin embargo, para estudiantes de un primer curso no es inmediato reconocer las conexiones entre la lógica y lo que necesitarán aprender en estudios futuros. Para mitigar esta sensación de desconexión entre contenidos básicos y más avanzados hemos adaptado la metodología de la asignatura de Lógica incorporando el uso de `DeduccionNatural.pl`, una herramienta implementada en Prolog (lenguaje de programación lógica), para la resolución y corrección de las demostraciones de deducción natural en lógica proposicional. Con `DeduccionNatural.pl` las demostraciones son programas, las reglas de inferencia son funciones de una librería predefinida, y las reglas derivadas son subrutinas que permiten refactorizar las demostraciones. Para evaluar el impacto de esta nueva metodología hemos trabajado con dos grupos de alumnos de titulaciones distintas en el curso 2022/2023 y en el curso 2023/2024. Los resultados de la encuesta sobre el grado de satisfacción y la utilidad de `DeduccionNatural.pl` realizados en el curso 2022/2023 no son concluyentes, pero nos animan a seguir mejorando la herramienta (se adjunta link anonimizado a la versión modificada en el curso 2023/2024, en concreto el 24 octubre 2023) y diseñando actividades en lógica de primer orden.

2. Justificación

La lógica es la ciencia que estudia la validez formal de los razonamientos. Por medio de la formalización del lenguaje y de sus reglas básicas, proporciona las herramientas necesarias para poder tratar y resolver rigurosamente problemas que tienen sus orígenes y aplicaciones en todas las áreas de las ciencias [3, 5, 8].

Es, por lo tanto, natural que la lógica matemática y computacional sean materias básicas en la mayoría de las titulaciones universitarias relacionadas con la Ingeniería Informática [1, 4]. Su estudio tiene a menudo la difícil tarea de compensar las posibles carencias de la formación previa de los alumnos. Además, sirve para mejorar las tres competencias básicas de lectura, matemáticas y ciencias (evaluadas en el contexto español a través del informe PISA 2018 [14]) y la capacidad de razonar de forma analítica y crítica. Uno de los principales objetivos didácticos de las asignaturas de lógica es “la capacidad para adquirir, obtener, formalizar y representar el conocimiento humano en una forma computable para la resolución de problemas mediante un sistema informático en cualquier ámbito de aplicación, particularmente los relacionados con aspectos de computación, percepción y actuación en ambientes o entornos inteligentes”.

Desde nuestro punto de vista, un objetivo principal de la asignatura Lógica es el de introducir algunos sistemas de demostración formal. Para la mayoría de los alumnos no es inmediato justificar correctamente la validez de un razonamiento o la resolución de un problema concreto y necesitan de una amplia introducción a los lenguajes formales para aprender esta destreza.

Los métodos deductivos de la lógica matemática están en la base de la demostración automática de teoremas. Se trata de buscar los sistemas de demostración más eficientes para su implementación en un ordenador. Los alumnos tendrán que aplicar métodos de demostración en todos los cursos de Matemáticas y en cursos más avanzados de Informática, p.ej., en Programación Declarativa [15], que tradicionalmente incluye programación lógica y programación funcional.

Sin embargo, en los últimos años hemos observado que no es fácil conseguir que alumnos de un primer curso universitario entiendan la importancia de la lógica y de sus aplicaciones a la programación.

Por su formulación más intuitiva, en nuestras asignaturas decidimos presentar el método de Deducción Natural de Gentzen [2, 5, 9]. En el sistema de Deducción Natural no hay axiomas y las reglas de inferencia se pueden fácilmente interpretar como funciones predefinidas con una API (Interfaz de Programación de Aplicaciones) concreta. Las reglas derivadas son subrutinas que el usuario puede definir para reutilizarlas en

diversas demostraciones sin tener que repetir un patrón de demostración concreto, sirva como ejemplo la regla derivada “ModusTollens”.

Para motivar a los alumnos se decidió desarrollar e incorporar a la metodología docente un programa escrito en programación lógica, DeduccionNatural.pl, cuya función principal es la comprobación automática de la corrección de demostraciones en el sistema de deducción natural, en el contexto de la lógica proposicional.

Al fin de evaluar nuestra experiencia docente, hemos implementado el uso de esta herramienta durante el desarrollo de actividades prácticas en dos grupos de alumnos pertenecientes a dos grados diferentes. Para uno de ellos, el uso de la herramienta era obligatorio, mientras que para el otro su uso era voluntario. Las contribuciones principales de este trabajo son:

Explicar y demostrar que es posible incorporar el uso de DeduccionNatural.pl para comprobar que las demostraciones hechas por el alumno son correctas, y que al tener una curva de aprendizaje muy baja (no requiere instalación previa), esta incorporación se puede realizar sin necesidad de alterar el temario diseñado en las guías docentes.

DeduccionNatural.pl es de código abierto y se podría adaptar para que se ajuste al planteamiento de otros profesores, p. ej., en el formato de presentar las demostraciones y/o en el conjunto de reglas de inferencia o derivadas que se pueden aplicar.

Aunque el programa podría implementarse en cualquier lenguaje de programación, entendemos que el uso de un lenguaje de programación lógica permite visualizar una de las más inmediatas aplicaciones de la lógica.

Los resultados presentados en esta memoria han sido mencionados en trabajos anteriores y presentados en congresos de innovación docente tanto nacionales como internacionales [se omiten referencias], sin embargo, esta es la primera vez que se publican los resultados de manera exhaustiva.

3. Desarrollo

Objetivos

El objetivo principal es el de concienciar a los alumnos de la importancia de la Lógica en las carreras de la rama de Informática y que comprendan la similitud entre las demostraciones matemáticas y los programas informáticos. Para obtener este objetivo hemos definido tres objetivos secundarios.

El primer objetivo secundario está encaminado a disponer de material didáctico (en abierto) adecuado:

- Desarrollar una herramienta para la comprobación de demostraciones en Deducción Natural usando Ciao Prolog, porque es un lenguaje programación lógico, de código abierto, y que ofrece un Playground online para poder ejecutar los programas directamente en cualquier navegador web [6].
- Que fuese un programa sencillo: (i) con pocas líneas de código para que el alumno pueda revisar el programa y (ii) que las demostraciones se pueden escribir en una línea.
- Que se pueda distribuir bajo licencia de código abierto para facilitar su utilización, mejor y/o adaptación.

El segundo se refiere a las acciones necesarias para diseñar una actividad que no requiera adaptar la guía docente de la asignatura de modo que permita su replicación por otros docentes (en otras universidades):

- Hemos diseñado una práctica en la que los alumnos tengan que desarrollar capacidades transversales como el trabajo autónomo, trabajo en equipo, comunicación con los profesores (incluyendo el uso de los foros del aula virtual) entre otros.
- Para poder evaluar la idoneidad de la actividad propuesta hemos ajustado la práctica para poder tener un grupo de evaluación y otro de control.

Finalmente, para evaluar el impacto de la actividad propuesta:

- Hemos diseñado y realizado una encuesta de satisfacción.
- Hemos comparado los resultados obtenidos por los alumnos comparando: por un lado (i) resultados entre el grupo de evaluación y el de control, (ii) resultados al usar la herramienta vs. cuando no se usaba; y por otro los resultados en el conjunto de la práctica vs. el resultado al realizar el ejercicio concreto de Deducción Natural.

Diseño de la práctica

En el estudio de la lógica computacional existen varios trabajos mostrando las semejanzas entre el proceso de demostración en deducción natural y la programación [12]. Además, en el contexto de la deducción natural, se han desarrollado herramientas como Taut-Logic (<https://www.taut-logic.com>), Fitch (<http://logic.stanford.edu/logica>), el Asistente para la Deducción Natural (ADN) desarrollado en la Universidad de Alicante [11], la Lógica Simbólica Deductiva (LSD) desarrollado en la Universidad de Girona [7], y otras, como las descritas en [13].

Sin embargo, estas herramientas no resaltan las similitudes entre deducción natural y la programación, y, pese a que la mayoría son herramientas de código abierto no es fácil acceder a su código fuente (en el caso de estar operativas, que no todas lo están) para poder adaptarlas. Además, dicha adaptación no sería fácil porque las implementaciones incluyen interfaces de usuario ad hoc y modificaciones en la lógica de los sistemas implicaría adaptar la interfaz.

Basado en el análisis de estas herramientas, hemos implementado DeduccionNatural.pl con los siguientes objetivos: (i) crear un programa sencillo (tiene menos de 300 líneas de código y las demostraciones se pueden escribir con pocas líneas de código), (ii) distribuirla bajo licencia de código abierto, (iii) implementarla en Ciao Prolog, un intérprete (de código abierto) de Prolog, y (iv) que se pudiese ejecutar directamente en cualquier navegador web usando el Playground de Ciao Prolog. El siguiente enlace [blinddeduccionnatural23b](#), da acceso directo a la versión de 2023 en el navegador, sin necesidad de instalación.

Por último, hemos consultado algunas de las propuestas existentes a la hora de formalizar las demostraciones en deducción natural y ejemplos de juegos lógicos utilizados en la enseñanza de Lógica:

- Pepa Hernández (<http://www.dia.fi.upm.es/es/phernan> UPM), propone las reglas de inferencia y el formato basado en sangría para las demostraciones utilizadas en este trabajo.
- José A. Alonso Jiménez (<https://www.cs.us.es/~jalonso> US), tiene multitud de trabajos sobre Lógica, desde apuntes hasta ejercicios introductorios a Prolog (E I formato para representar los supuestos difiere del formato usado en este trabajo (aunque son equivalentes). La introducción de constantes y variables en lógica de primer orden también es diferente, pero no es relevante para este trabajo).
- José Morales (<https://jfm.c.github.io> UPM), usa demostradores de teoremas, por ejemplo, comprueban el problema lógico del lobo, la oveja y la lechuga con Z3.

Como resultado, hemos decidido usar la sangría frente a otras propuestas (como las basadas en la notación de Huth/Ryan o la de Fitch, con cajas o líneas) por ser más cercana a la estrategia utilizada en distintos lenguajes de programación (por ejemplo, C y Python). Obsérvese que mientras en C la sangría se usa para facilitar la lectura, en Python se usa para definir bloques de código (ver https://www.w3schools.com/python/python_syntax.asp).

Metodologías docentes aplicadas

Los conocimientos previos y los hábitos de estudio de nuestros alumnos de primer curso suelen ser bastante heterogéneos. Lo mismo pasa con su nivel de motivación. Por ello, hemos optado por aplicar una metodología docente híbrida [3, 10] y hemos adaptado la misma a las características específicas de cada grupo de alumnos y/o titulación.

Se realizan clases invertidas cuando la dificultad de los contenidos lo permite. Como regla general, los alumnos tienen a su disposición el material docente con una cierta antelación. El Aula Virtual sirve de referencia principal para el desarrollo de la asignatura y la introducción de material y herramientas adicionales (ejercicios y exámenes resueltos, vídeos, herramientas de autoevaluación, enlaces interesantes). Por otro lado, el uso del Aula Virtual facilita la interacción entre alumnos y profesores.

Adicionalmente, desde que se planteó la docencia a distancia se han desarrollado herramientas y métodos docentes novedosos y aptos a facilitar el trabajo independiente del alumno y su autoevaluación. La mayoría de estas herramientas están diseñadas también para facilitar el trabajo docente y de evaluación de los profesores. Asimismo, la crisis generada por el COVID sirvió para incrementar el empleo de herramientas diseñadas para la docencia en remoto con el fin de adaptar asignaturas presenciales a la nueva situación.

Para incorporar el uso de DeduccionNatural.pl y poder evaluar el impacto y la aceptación por parte de los alumnos, hemos decidido implementar su uso en dos clases similares, pero con requerimientos diferentes:

- Para el grupo del grado en Inteligencia Artificial (40 alumnos) en la práctica del bloque 1 el uso de DeduccionNatural.pl era obligatorio.
- En el grupo del grado de Ingeniería de la Ciberseguridad (65 alumnos) la práctica del bloque 1 era similar, pero el uso de DeduccionNatural.pl era optativo (no afectaba a la nota).

Recursos materiales, humanos, espaciales o tecnológicos empleados

En esta sección explicamos el funcionamiento de la herramienta. La Sección 4.2.1 describe cómo se han traducido los operadores y las reglas de inferencia. La Sección 4.2.2 explica cómo ejecutar el programa para comprobar si una demostración es correcta dadas unas premisas y un consecuente y, finalmente, la Sección 4.2.3 muestra cómo definir reglas derivadas, y/o reglas auxiliares para factorizar las demostraciones.

Sintaxis, constantes y reglas

Como hemos visto en los apuntes de la asignatura de Lógica, redactados por [2], en el contexto de la lógica proposicional, contamos con un alfabeto compuesto por símbolos de proposición y conectivas. Para facilitar la representación de dichos símbolos en Prolog hemos decidido realizar la siguiente representación:

- Los símbolos de proposición (que representan hechos), se representan como átomos en Prolog, palabras que empiezan por minúscula (con o sin números), p.ej., p,q,r,...,p1,p2,....
- Las conectivas lógicas se traducen en los siguientes operadores (la preferencia de los operadores está definida en las líneas 11-13):

Conectiva	\neg	\vee	\wedge	\rightarrow	\leftrightarrow
Operador	!	or	and	-->	<-->

Figura 1. Equivalencia entre conectivas lógicas y operadores en programación.

- De igual manera, hemos traducido las 10 reglas de inferencia básicas (2 por cada conector lógico). La sintaxis de dichas reglas según la implementación de DeduccionNatural.pl es la siguiente:

Conectiva	Introducción	Eliminación
and	'I' and (_,_)	'E' and a(_) 'E' and b(_)
or	'I' or a(_,F) 'I' or b(F,_)	'E' or (_,_,_)
!	'I' ! (_)	'E' ! (_)
-->	'Supuesto' (F) 'I' --> (_,_)	'E' --> (_,_)
<-->	'I' <--> (_,_)	'E' <--> a(_) 'E' <--> b(_)

Figura 2. Reglas de introducción y eliminación para conectivas lógicas.

Nótese que las reglas de inferencia están en mayúsculas y, por lo tanto, para que Prolog las trate como átomos (y no como variables) hay que usar comillas, i.e., 'I'. Dependiendo del número de fórmulas que intervienen en cada regla de inferencia, tenemos diferente número de argumentos en la implementación, p. ej. para la negación tanto la introducción, 'I' ! (_), como la eliminación, 'E' ! (_), solo precisa de una fórmula, y para la eliminación de la disyunción 'E' or (_ , _ , _), hay tres argumentos, i.e., $\phi \vee \psi$, $\phi \rightarrow \chi$, y $\psi \rightarrow \chi$.

Adicionalmente, las reglas de inferencia, 'I' or a (_ , F), 'I' or b (F , _), y 'Supuesto' (F), permiten al usuario introducir nuevas fórmulas. En este caso, hemos usado la sintaxis de Prolog para las variables, p.ej., F, y en cada caso se instanciarán con la fórmula pertinente.



```

1  %~~~~~
2  % Copyright (C)2023 xxx (URJC)
3  % Name: DeduccionNatural.pl
4  % Author: xxx
5  % Date: 24 October 2023
6  % Purpose: Execute Natural Deduction Proofs
7  % LICENSE: Apache License 2.0
8  %~~~~~
9
10
11
12 % Operator precedence
13 :- op(200, fy, !).
14 :- op(400, xfy, [and, or]).
15 :- op(600, xfy, [==>, <=>]).
16
17 % Auxiliary precedence for !
18 % Used to define the inference rules
19 :- op(400, xfy, !).
20
21 %% Bug found by students of the cybersecurity degree of the URJC in
22 %% the academic year 23/24. Solved in 24 October 2023
23 bug :-
24     main([s ==> c],
25         [
26             'Premisa'(1),
27             'Supuesto'(s),
28             'E' ==> (1,2),
29             'I' ==> (2,3),
30             'I' or a(3,sa),
31             'E' ==> (5,2)
32         ]
33     ).
34
24  ?- use_module('/draft.pl').
25  yes
26  ?-

```

Figura 3. Captura de pantalla con la ejecución de *DeduccionNatural.pl* actualizada durante el curso 2023/2024.

Implementar Demostraciones

Para ejecutar *DeduccionNatural.pl* recomendamos utilizar el Playground de Ciao siguiendo el siguiente enlace [blinddeduccionnatural23b](#). Como hemos comentado anteriormente, ejecutando la consulta (en el panel de la derecha del Playground) `?- ejemplo1`, obtenemos el resultado esperado (ver Figura 3).

Programar nuevas Reglas Derivadas

Como hemos mencionado anteriormente, es posible definir las reglas derivadas clásicas y/o definir reglas auxiliares para factorizar demostraciones. A modo de ejemplo en las líneas 208-216 se facilita la implementación de la regla derivada Modus Tollens que explicamos a continuación:

La regla de Modus Tollens, se puede resumir como “Si P implica Q, y Q no es cierto, entonces P no es cierto”, i.e., indica que puede negarse el antecedente de una implicación si se niega su consecuencia. Como ya hemos comentado anteriormente, esta regla se puede, y debe, demostrar usando las reglas básicas de inferencia de la siguiente manera:

```

1  regla( 'MT',                % Nombre
2      [FA -> FB, ! FB],      % Premisas
3      ! FA,                  % Consecuente
4      [ 'Premisa' (1),       % Reglas
5        'Premisa' (2),       % ...
6        'Supuesto' (FA),
7        'E' -> (1, 3),
8        'I' and (4, 2),
9        'I' -> (3, 5),
10       'I' ! (6) ]).
```

Figura 4. Demostración de la regla de Modus Tollens.

El predicado que utilizaremos para definir reglas derivadas en regla/4, que recibe cuatro argumentos: (i) nombre de las reglas, (ii) lista de premisas, (iii) consecuente, (iv) lista de reglas de inferencia básicas que demuestran la validez de la regla. Obsérvese que en las reglas derivadas nos referimos a las fórmulas en mayúsculas para indicar que son variables y que las instanciaremos con las fórmulas que correspondan según el ejemplo o ejercicio correspondiente.

Desarrollo de la práctica

En la guía docente de ambos grupos (Inteligencia Artificial y Ciberseguridad) se proponen dos prácticas a lo largo del curso. La práctica 1 propuesta se compone de los siguientes 3 ejercicios:

1. Formalización en lenguaje proposicional.
2. Clasificación semántica de fórmulas haciendo uso de las tablas de verdad.
3. Comprobación de la validez de un argumento, i.e., comprobar si el consecuente es consecuencia lógica de las premisas. Este ejercicio consta de los siguientes apartados:

- a) Verificación mediante un método semántico.
- b) Demostración mediante deducción natural. Es el apartado objeto de este trabajo: Al grupo de Inteligencia Artificial se les exige el código usado en DeduccionNatural.pl, al de Ciberseguridad se les da como optativo.
- c) Traducir el argumento (escrito en lógica proposicional) a lenguaje natural.

Es importante hacer notar que, en el ejercicio 3 de deducción lógica, se propone verificar la veracidad de la fórmula por dos métodos distintos para forzar que, de cometer algún error e incurrir en una contradicción en los apartados 3a) y 3b), puedan corregirla. Sin embargo, si ambos apartados se resuelven incorrectamente, es menos probable, para un alumno/a, sospechar del error. Este último caso no es posible si se hace uso de la herramienta propuesta, DeduccionNatural.pl, porque ella solo nos validará el resultado cuando se siga un procedimiento válido de deducción.

Por último, y en la misma línea, el apartado 3c) en el que se pide escribir en lenguaje natural un conjunto de premisas y un consecuente que corresponda con el dado en el ejercicio, tiene por objetivo corroborar que los apartados 3a) y 3b) son correctos, encontrando un caso particular que sirva de ejemplo ilustrativo.

La segunda práctica se centra en lógica de primer orden y, por tanto, no es posible el uso de la herramienta DeduccionNatural.pl.

Sistemas de evaluación del alumnado empleados

Tabla 1. Test U Mann-Whitney: IA vs. Ciber

	Globales		Específicas
	Práctica 1	Práctica 2	DN
<i>p</i> -valor	0,113 > α	0,108 > α	0,025 < α

Tabla 2. Test U Mann-Whitney: Con APP vs. Sin APP

	Globales		Específicas
	Práctica 1	Práctica 2	DN
<i>p</i> -valor	0,048 < α	0,021 < α	0,016 < α

Llevamos a cabo un estudio estadístico para comparar los resultados de las calificaciones obtenidas en dos escenarios: 1) El grupo A es el grado de Inteligencia Artificial (IA) y el grupo B es el grado de Ciberseguridad (Ciber), y 2) El grupo A son los grupos que han utilizado DeduccionNatural.pl y el grupo B los grupos que no lo han utilizado.

En ambos casos consideramos tres comparaciones: Comparación Global: por un lado, compara las calificaciones obtenidas en la Práctica 1, por el otro, compara las calificaciones en la Práctica 2; Comparación

Específica: se comparan las calificaciones obtenidas en el ejercicio de deducción natural (DN) de la Práctica

1. En los tres casos se utilizan las siguientes hipótesis: Hipótesis alternativa H1: la media de calificaciones del grupo A, es mayor que la del grupo B. Hipótesis nula H0: ambas medias no son significativamente diferentes.

Debido al número reducido de muestras, realizamos una prueba no paramétrica de U Mann-Whitney, tomando como nivel de significación $\alpha = 0,05$. En la Tabla 1 se muestran los resultados de las tres comparaciones descritas anteriormente, y los histogramas de las calificaciones de cada grupo en el Apéndice.

1. Inteligencia Artificial vs Ciberseguridad

La Tabla 1 muestra que los resultados de la prueba estadística en las comparaciones globales impiden rechazar la hipótesis nula H0 ($p > 0,05$), por lo que, de forma global, no se aprecia una diferencia significativa entre los dos grupos.

Por otra parte, cuando comparamos las calificaciones en el ejercicio específico de deducción natural, a la luz de los resultados, podemos rechazar la hipótesis nula con el valor de significación considerado ($p < 0,05$). Nótese que, mientras en el grupo de Inteligencia Artificial, todos han hecho uso de la aplicación de DeduccionNatural.pl (era obligatorio), en el grupo de Ciberseguridad, solo el 17,4 % hizo uso de esta (era voluntario). Esto podría justificar la diferencia observada en el ejercicio específico de deducción natural.

2. Con vs. Sin Programa (APP)

La Tabla 2 muestra los resultados comparando los grupos que han usado DeduccionNatural.pl con los que no, y en este caso las diferencias estadísticas son más pronunciadas. En las comparaciones globales, tanto en la Práctica 1 como en la Práctica 2, podemos rechazar la hipótesis nula. Pero, en el ejercicio específico de Deducción Natural (DN), obtenemos aún más diferencias que en la comparación anterior, lo cual se alinea con la hipótesis inicial de que el uso de DeduccionNatural.pl pueda ser significativa en el aprendizaje de Lógica.

Sin embargo, los valores p obtenidos, aunque menores que el valor de significación considerado ($\alpha = 0,05$), son aún cercanos a este, y los resultados deben ser tomados con cautela. Estos resultados, aunque limitados, sugieren y motivan una mayor experimentación e investigación en el uso y desarrollo de DeduccionNatural.pl.

4. Resultados

Metodología de análisis

De las consultas que nos realizaron los alumnos durante las sesiones de práctica hemos identificado dos tipos de errores, Ejemplos 1 y 2, que en la corrección de un examen no se calificarían como errores, pero al usar DeduccionNatural.pl el alumno obtiene un error.

Ejemplo 1 (Orden de los argumentos)

A la hora de aplicar la introducción de la negación la API de DeduccionNatural.pl requiere que la regla sea de la forma: $B \rightarrow A \wedge \neg A$.

Por lo tanto, a la hora de “construir” la contradicción $A \wedge \neg A$ hay que aplicar la regla de introducción de la conjunción pasando A como primer argumento y $\neg A$ como segundo argumento.

Ejemplo 2 (No hay atajos)

La regla derivada Modus Tollens (MT) permite reducir el número de instrucciones a la hora de realizar la demostración y por lo tanto los alumnos han querido usarla a la menor oportunidad. Recordamos que la regla MT es [2]: $\{A \rightarrow B, \neg B\} \vdash \neg A$.

Sin embargo, cuando durante la resolución de la práctica se encontraron con unas reglas de la forma: $\{p \rightarrow \neg q, q\}$.

El programa daba error porque lo que había que pasar como segundo argumento es $\neg B$, i.e., $\neg \neg q$. En lógica clásica sabemos que $\neg \neg A \equiv A$. De hecho, entre las reglas de inferencia tenemos la regla que nos permite inferir A partir de $\neg \neg A$ (regla de eliminación de la negación), sin embargo, la regla $\{A\} \vdash \neg \neg A$ hay que demostrarla (ver Tema 2, Ej. 12, pág. 50 en [2]).

Tabla 3. Comparativa resultados ejercicio de deducción natural.

		Calificaciones		Orden incorrecto	Atajos (errores / soluciones)		
		Nota	# Grupos	# Errores	# Errores	Sol. I^*	Sol. MT^*
Con APP (18)	IA (14)	9-10	9	0	0	0	9
		7-9	4	0	0	0	4
		0-7	1	0	0	0	1
	Ciber (23)	9-10	2	0	0	1	1
		7-9	1	0	0	0	1
		0-7	1	0	0	0	1
Sin APP (19)		9-10	6	0	1	0	5
		7-9	8	1	1	2	5
		0-7	5	1	3	0	2

La Tabla 3 muestra una comparativa entre grados y entre grupos que han usado (o no) DeduccionNatural.pl, para analizar: las calificaciones obtenidas, la presencia de los errores más relevantes detectados durante la ejecución y/o corrección de las prácticas.

En la primera columna indicamos el número de grupos que han sacado sobresaliente, notable, o menos. En la segunda columna, cuantos grupos han cometido “errores” del Ejemplo 1. Para los errores del Ejemplo 2 tenemos las siguientes columnas: (i) cuántos grupos han cometido este tipo de error, (ii) cuántos grupos han usado/replicado la regla $I \rightarrow^*$, i.e., $\{A\} \vdash \neg \neg A$, para luego aplicar MT y (iii) cuántos grupos han usado el esquema deductivo de MT o han definido una nueva regla MT^* tal que $\{A \rightarrow \neg B, B\} \vdash \neg A$.

Encuesta de satisfacción

Todo desarrollo software es fruto de un proceso iterativo de diseño, implementación y evaluación. En esta sección hacemos uso de una encuesta como herramienta de evaluación para detectar limitaciones, posibles errores y mejorar de DeduccionNatural.pl con vistas a futuro. La Fig. 5 muestra los resultados de la encuesta realizada por los alumnos del grado en Inteligencia Artificial (que tenían la obligación de hacer uso de la aplicación en la práctica grupal). La Fig. 5a muestra los resultados de las preguntas que buscan conocer el alcance de uso por parte de los estudiantes. Para ellos hemos realizado las siguientes preguntas: (A) ¿Has utilizado DeduccionNatural.pl?, (B) ¿Lo has utilizado para comprobar demostraciones?, (C) ¿Lo has utilizado como herramienta de estudio?

Las Fig. 2b y 2c muestran los resultados de las valoraciones (de aquellos alumnos que contestaron afirmativamente a las tres preguntas) con una escala numérica entre 1 y 4 (1-muy mal, 4-muy bien) de, respectivamente, la facilidad de uso de la aplicación y de su utilidad para el aprendizaje de deducción natural.

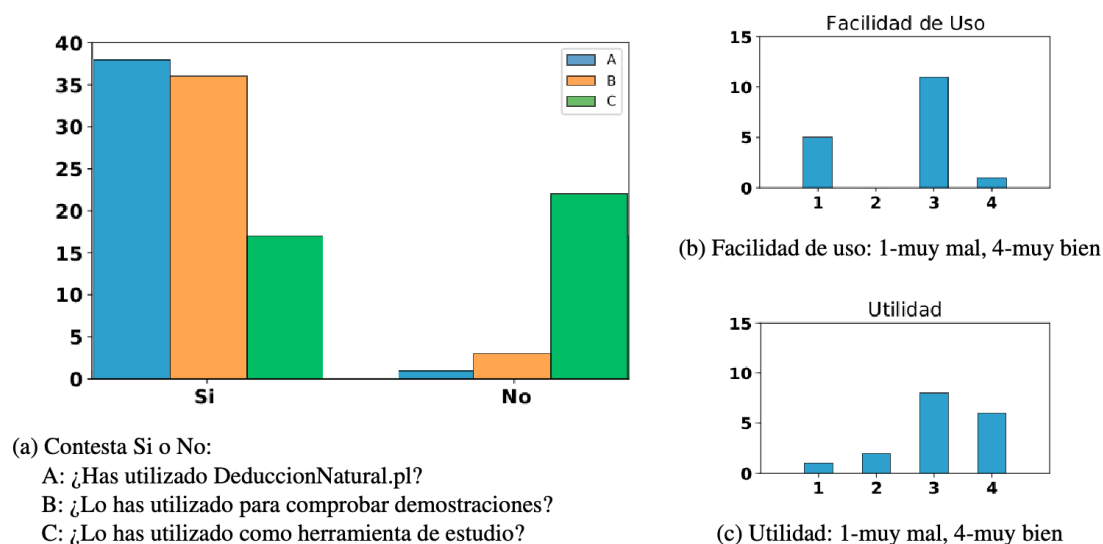


Figura 5. Resultados de la encuesta de satisfacción.

La Fig. 5a muestra una disminución del uso de la aplicación como herramienta de estudio, a pesar de que sí ha sido utilizada como herramienta de comprobación. Es importante resaltar que los grupos eran de tres alumnos, por lo tanto, no todos tenían porque haber utilizado DeduccionNatural.pl al realizar la práctica – estos resultados coinciden con la percepción que tuvimos durante el desarrollo de las prácticas, pues los todos los alumnos asistieron a clase y trabajaron de manera “solidaria” en todos los ejercicios.

Por otro lado, en la Fig. 5b, se aprecia una cierta neutralidad en las valoraciones de facilidad de uso. De nuestra experiencia en clase, interpretamos que la herramienta ha resultado fácil de usar a los alumnos con experiencia en programación y no tan fácil a aquellos sin experiencia.

Finalmente, hemos apreciado un sesgo positivo, en los resultados mostrados en la Fig. 5c, que indica que la herramienta les ha parecido útil. A la vista de estos resultados, concluimos que una posible mejora de DeduccionNatural.pl tiene como elemento prioritario un diseño de la interfaz más amigable e intuitivo, y que no requiera conocimientos de programación.

Conclusiones

En este artículo hemos presentado evidencias de las ventajas que ofrece la adaptación de la metodología docente en la asignatura de Lógica para incorporar el uso de DeduccionNatural.pl. Hemos observado igualmente que los alumnos han encontrado dificultades al tener que lidiar con los errores de sintaxis al usar la herramienta. Esta dificultad, aunque recurrente durante el desarrollo de software, no era una tarea prevista al plantear esta actividad y, por lo tanto, estamos trabajando en el desarrollo de una interfaz gráfica que facilite el uso de DeduccionNatural.pl. Por ejemplo, un alumno de la URJC está desarrollando una interfaz web (basada la librería Blockly de Google) como parte de su TFG.

Por otro lado, una vez los alumnos se familiarizan con la sintaxis, el uso de DeduccionNatural.pl ha permitido comprobar que los alumnos tienden a aceptar como válidas aplicaciones de reglas de inferencia sin atender al orden de los argumentos y/o realizando inferencias (en el aire) de modo que se convencen de que es el programa el que falla. Como ya indicamos, estos errores no serían considerados fallos al evaluar un examen, sin embargo, son errores típicos a la hora de programar y que, aunque suelen ser detectados por los compiladores, si no es así (especialmente en el caso de lenguajes de programación no tipados), estos errores implican la aparición de bugs.

Adicionalmente, hemos comprobado que nuestra propuesta ha despertado el interés de alumnos y profesores de otros grados y universidades: (i) otros alumnos de nuestra universidad, del grado de Ingeniería Informática, preguntaron a su profesora por el programa que estaban usando en el grado de Ciberseguridad y que resolvía los ejercicios de Deducción Natural (pese al entusiasmo de estos alumnos, por el momento, DeduccionNatural.pl solo permite comprobar si la demostración es correcta), y (ii) los profesores Pepa Hernández y José Morales, que imparten Lógica en la ETSIInf de la UPM, han mostrado interés por DeduccionNatural.pl para utilizarlo el próximo curso.

Como trabajo futuro tenemos dos líneas de actuación previstas: (i) seguir mejorando (y diseminando) la metodología docente que hemos implementado gracias a DeduccionNatural.pl y (ii) continuar con las mejoras de DeduccionNatural.pl con nuevos (y más intuitivos) interfaces, con soporte para Lógica de primer orden (incluyendo cuantificadores), con la generación automática de ejemplos, la corrección automática de prácticas y/o exámenes, entre otros.

5. Equipo docente



Joaquín Arias

Desde septiembre de 2020 soy profesor en la Universidad Rey Juan Carlos. En febrero de 2020 obtuve el doctorado en Software, Sistemas y Computación con mención cum-laude y distinción internacional, culminando un camino que empezó en 2011 cuando me matriculé en el grado en Matemáticas e Informática en la ETSIInf de la Universidad Politécnica de Madrid. Desde 2013

he trabajado como investigador pre-doctoral en IMDEA Software Institute donde he participado en numerosos proyectos internacionales y nacionales realizando dos estancias de más de tres meses: en Dallas y Helsinki (co- financiado por el programa de innovación y emprendimiento de EIT-Digital). Con anterioridad (en 2002) obtuve en título de arquitecto por la Universidad Politécnica de Madrid desde 2002 y durante más de 10 años trabajé como arquitecto y gestor de proyectos residenciales en Madrid, Marbella y Rumanía.



Iván Ramírez

Soy profesor en la Escuela Técnica Superior de Ingeniería Informática de la URJC, y colaborador en el Grupo de Investigación CAPO. Líneas de investigación e intereses generales: Mis intereses abarcan desde la teoría del aprendizaje automático a métodos variacionales aplicados a la Visión Artificial, y de forma más general, a la Inteligencia Artificial: Image Processing;

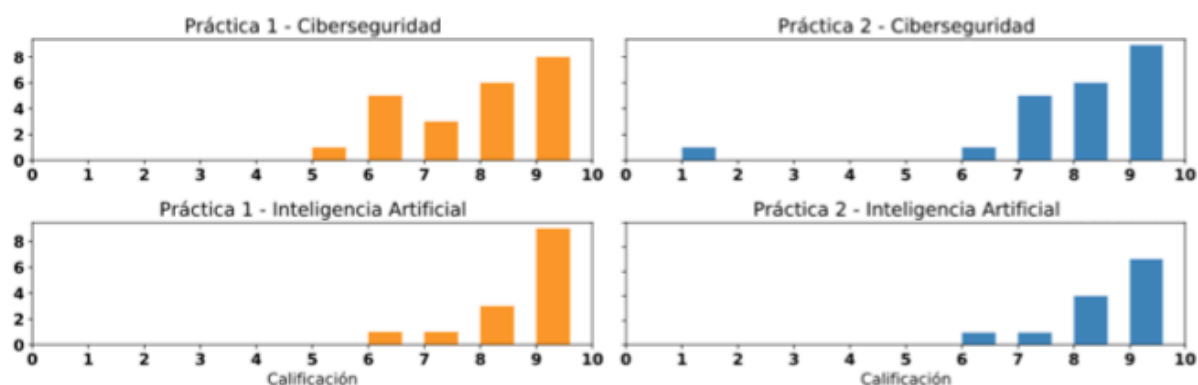
Representation Learning; Robustness in Machine Learning; Bayesian Neural Networks; Generative Models; Fairness in Machine Learning.

6. Referencias bibliográficas

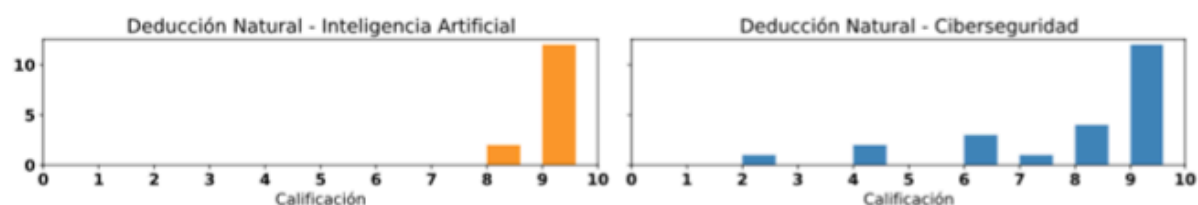
- [1] ANECA. Libro Blanco del Título de Grado en Ingeniería Informática. Proyecto EICE. <https://www.aneca.es>, 2005.
- [2] Joaquín Arias. Lógica: desde Aristóteles hasta Prolog. Servicio de Publicaciones URJC, 2022. <http://hdl.handle.net/10115/20014>.
- [3] María Jesús Castel de Haro, Carlos-José Villagrà Arnedo, y S Cortés Vaíllo. Materiales docentes para el aprendizaje de la asignatura lógica computacional. En Propuestas de diseño, desarrollo e innovaciones curriculares y metodología en el EEES, pp. 289–302. Instituto de Ciencias de la Educación, 2009.
- [4] Association for Computing Machinery (ACM) e IEEE Computer Society (IEEE-CS). Computing Curricula 2020 (CC2020): Paradigms for Global Computing Education. <https://www.acm.org>, 2020.
- [5] Alessandra Gallinari. Apuntes y problemas de la lógica matemática. Dykinson, Madrid, 2009.
- [6] Guillermo García-Pradales, José F. Morales, Manuel V. Hermenegildo, Joaquín Arias, y Manuel Carro. An s(CASP) In-Browser Playground based on Ciao Prolog. En ICLP Workshops, 3193. CEUR-WS, 2022.
- [7] Josep Humet. LSD, una herramienta didáctica para el aprendizaje de la lógica. JENUI 2001, pp. 482–485, 2001.
- [8] Jose Emilio Labra Gayo. ¿Hay Lógica en la situación actual de las titulaciones informáticas? En JENUI 2004, pp. 227–234, 2004.
- [9] Gerhard Gentzen. Untersuchungen über das logische schließen. I. Mathematische Zeitschrift, 35:176–210, 1935.
- [10] Joe Miró Julià. ¿Tiene sentido la innovación docente? En JENUI 2019, pp. 47–53, 2019.
- [11] Faraón Llorens Largo y Sergio Mira Cabrera. Herramienta para la enseñanza de la Deducción Natural. En JENUI 2000, pp. 496–502, 2000.
- [12] Faraón Llorens Largo, Rosana Satorre Cuerda, Francisco Escolano, y Pilar Arques Corrales. Deducción natural versus computación. JENUI 1999, pp. 259–265, 1999.
- [13] Petr Manas. CLPractice 2.0. Tools for Learning: Computation and Logic. University of Edinburgh, 2021. https://project-archive.inf.ed.ac.uk/ug4/20201668/ug4_proj.pdf
- [14] OECD. Resultados de PISA 2018 para España. <https://www.oecd.org/pisa>, 2019.
- [15] Ana Pradera y Juan M. Serrano. Programación Declarativa: presentaciones. BURJCDigital, 2022. <https://hdl.handle.net/10115/20609>

7. Apéndice

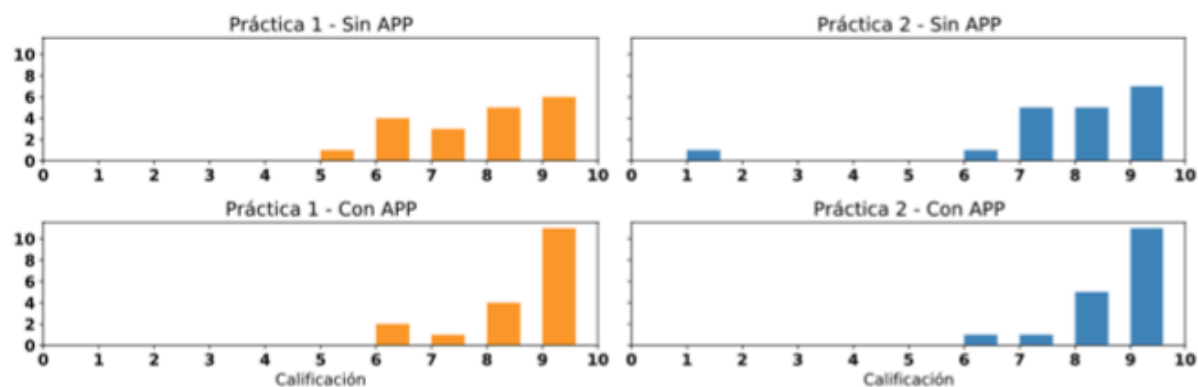
A.1. Prácticas 1 y 2: Ciber vs. IA



A.2. Ejercicio DN: Ciber vs. IA



A.3. Prácticas 1 y 2: Con vs. Sin APP



A.4. Ejercicio DN: Con vs. Sin APP

