Revista Brasileira de Informática na Educação – RBIE
Brazilian Journal of Computers in Education
(ISSN online: 2317-6121; print: 1414-5685)
http://br-ie.org/pub/index.php/rbie

Submission: 15/Mar/2023; Camera ready: 20/Feb/2024; 1st round notif.: 07/Jul/2023; Edition review: 02/Aug/2024; New version: 12/Sep/2023; Available online: 27/Aug/2024; 2nd round notif.: 27/Nov/2023; Published: 27/Aug/2024;

NADIA - Natural DeductIon proof Assistant: Um Assistente de Provas de Dedução Natural para Lógica Proposicional e Lógica de Predicados

Title: NADIA - Natural DeductIon proof Assistant: a Natural Deduction Assistant for Propositional and Predicate Logic

Título: NADIA - Natural DeductIon proof Assistant: Un Asistente de Pruebas de Deducción Natural para Lógica Proposicional y Lógica de Predicados

Davi Romero de Vasconcelos
Universidade Federal do Ceará – Campus Quixadá
ORCID: 0000-0002-4857-4156
daviromero@ufc.br

Maria Viviane Menezes
Universidade Federal do Ceará – Campus Quixadá
ORCID: 0000-0002-7096-4732
vivianemenezes@ufc.br

Resumo
A disciplina de Lógica para Computação faz parte da maioria dos cursos de Tecnologia da Informação e Comunicação. O sistema de Dedução Natural é amplamente utilizado para o ensino de demonstrações e este conteúdo consta em muitos dos livros-texto de Lógica. Este trabalho apresenta um assistente de provas, NADIA Natural Deduction Proof Assistant, para o sistema de Dedução Natural em Lógica Proposicional e Lógica de Predicados, no estilo de Fitch (caixas), com a finalidade de auxiliar no ensino-aprendizagem de estudantes de graduação e pós-graduação. NADIA permite que os estudantes escrevam suas demonstrações de forma mais próxima possível das provas que realizam no papel. NADIA verifica automaticamente se a demonstração está correta e, caso contrário, exibe os erros encontrados. Para avaliar a experiência dos estudantes no uso do NADIA realizamos uma avaliação da ferramenta com alunos de turmas semestrais da disciplina de Lógica para Computação que foram ofertadas nos anos de 2021 e 2022.
Palavras-chave: *Lógica para Computação; Dedução Natural; Assistente de Provas; Ferramenta de Ensino e Aprendizagem;*

Abstract
Logic in Computer Science course is part of most Information and Communication Technology courses. The Natural Deduction system is widely used for teaching proofs and appears in many Logic textbooks. This work presents a proof assistant, NADIA Natural Deduction Proof Assistant, for Natural Deduction system, in Fitch (boxes) style, in order to assist in the teaching-learning of undergraduate and graduate students. NADIA allows students to write their proofs as closely as possible to the proofs they take on paper. NADIA automatically checks whether the proof is correct and, if not, displays any errors found. To assess students' experience in using NADIA, we performed an evaluation of the tool in courses that were offered in 2021 and 2022.
Keywords: *Logic in Computer Science; Natural Deduction; Proof Assistant; Teaching Tools;*

Resumen
La disciplina de Lógica para Computación forma parte de la mayoría de los cursos de Tecnología de la Información y Comunicación. El sistema de Deducción Natural es ampliamente utilizado para la enseñanza de demostraciones

Cite as: Vasconcelos, D. R. & Menezes, M. V. (2024). NADIA - Natural DeductIon proof Assistant: Um Assistente de provas de Dedução Natural para Lógica Proposicional e Lógica de Predicados. Revista Brasileira de Informática na Educação – RBIE, 32, 842-870. DOI: 10.5753/rbie.2024.3249.

y este contenido aparece en muchos de los libros de texto de Lógica. Este trabajo presenta un asistente de pruebas, NADIA (Natural Deduction Proof Assistant), para el sistema de Deducción Natural en Lógica Proposicional y Lógica de Predicados, en el estilo de Fitch (con cajas), con el propósito de ayudar en la enseñanza-aprendizaje de estudiantes de grado y posgrado. NADIA permite que los estudiantes escriban sus demostraciones de la manera más cercana posible a las pruebas que realizan en papel. NADIA verifica automáticamente si la demostración es correcta y, en caso contrario, muestra los errores encontrados. Para evaluar la experiencia de los estudiantes en el uso de NADIA, realizamos una evaluación de la herramienta con alumnos de clases semestrales de la disciplina de Lógica para Computación ofrecidas en los años 2021 y 2022.
Palavras-chave: *Lógica para Computación; Deducción Natural; Asistente de Pruebas; Herramienta de Enseñanza y Aprendizaje;*

1 Introdução

No início do século passado, correntes filosóficas surgiram objetivando a fundamentação da matemática em bases sólidas utilizando a lógica para tal da Costa, 2008. A escola formalista de Hilbert, por exemplo, buscava a prova da auto-consistência da matemática. Em particular, ela buscava solucionar este problema de uma forma automatizada através de um procedimento (mecânico) que fosse capaz de verificar a consistência das proposições matemáticas. Contudo, apenas com o desenvolvimento dos computadores e com a automatização do raciocínio em Inteligência Artificial na década de 60 é que métodos automatizáveis de prova mais eficientes foram desenvolvidos Chang e Lee, 1976.

Como a escola formalista buscava a prova da auto-consistência da matemática, provas em matemática passaram a ser objetos de estudo da própria matemática dando início a uma área conhecida como *Teoria da Prova*. Um dos conceitos centrais desta teoria é o de *sistema dedutivo* que pode ser entendido como um mecanismo que permite a construção de argumentos formais, estabelecendo conclusões a partir de premissas.

O sistema dedutivo de Dedução Natural surgiu neste contexto Gentzen, 1969, buscando uma melhor análise das estrutura das provas lógicas, e é uma ferramenta essencial para a Teoria da Prova, pois permite um maior entendimento da noção de prova e do papel dedutivo das constantes lógicas, individualizadas pelas regras de introdução e eliminação. Nos sistemas de Dedução Natural, as inferências são quebradas em passos atômicos de tal forma que cada passo envolve uma constante lógica diferente. Tais passos são de dois tipos: aqueles que permitem a introdução dos conectivos lógicos; e, os que eliminam tais conectivos Gentzen, 1969. As provas iniciam a partir de premissas que são assumidas ou de suposições que são necessariamente descartadas no decorrer da prova. As derivações são, em geral, apresentadas na forma de árvores, também conhecido como estilo de Gentzen, ou por meio de caixas, denominado de estilo de Fitch.

O sistema de Dedução Natural é amplamente utilizado para o ensino de demonstrações e consta em muitos dos livros-texto de Lógica da Silva et al., 2006; de Souza, 2008; Huth e Ryan, 2004; van Dalen, 2013. O artigo Pelletier, 1999 apresenta uma breve história de Dedução Natural, traçando o desenvolvimento da Dedução Natural do ponto de vista que esses fundadores a abraçaram à ampla aceitação do método na década de 1960. O referido artigo apresenta as diferentes escolhas feitas por escritores de livros-texto com o objetivo de determinar quais são as "características essenciais" da natureza da dedução.

Além de ser a base do entendimento de provas matemáticas, a lógica tem um papel de relevo na fundamentação da computação e possui uma vasto espectro de aplicações desde a especificação e verificação de sistemas a aplicações em áreas como banco de dados, inteligência artificial, engenharia de software, dentre outras. No campus da Universidade Federal do Ceará em Quixadá, a disciplina Lógica para Computação faz parte dos currículos dos cursos de Sistemas de Informação, Engenharia de Software, Ciência da Computação e Engenharia de Computação como componentes obrigatórios. A disciplina possui um alto índice de reprovação. Para uma melhor assimilação dos conteúdos, é fundamental que os estudantes exercitem e que tenham *feedback* de suas demonstrações.

Nesse contexto, para auxiliar o aprendizado dos alunos em demonstração em Dedução Natural, propomos um assistente de provas, denominado *Natural Deduction Proof Assistant* (NADIA), que é uma ferramenta computacional que nos permite verificar se uma demonstração está correta ou não. Caso a demonstração não esteja correta, o assistente apresenta os erros encontrados na demonstração. O NADIA foi desenvolvido em Python, tem uma versão *Web* e pode ser integrada à plataforma Moodle.

Este artigo tem como objetivo apresentar a ferramenta desenvolvida; o contexto em que ela foi utilizada como ferramenta auxiliar no processo de ensino-aprendizagem de Dedução Natural em Lógica Proposicional e em Lógica de Predicados; e, os resultados obtidos com a avaliação do sistema realizada com alunos matriculados em turmas da disciplina de Lógica para Computação durante os anos letivos de 2021 e 2022. O trabalho é uma versão estendida de um artigo previamente publicado no WEI - *Workshop sobre Educação em Computação* D. Vasconcelos et al., 2022.

O restante desse trabalho está organizado como segue. A Seção 2 apresenta o Sistema de Dedução Natural. Na Seção 3, mostramos o Assistente de Provas. A Seção 4 traz os trabalhos relacionados. A Seção 5 relata a experiência de integração da ferramenta ao Ambiente Virtual de Aprendizagem Moodle. A Seção 6 aponta os resultados da avaliação da experiência do uso do NADIA. E, na Seção 7, apresentamos as conclusões e trabalhos futuros.

2 Fundamentação Teórica

O objetivo do estudo da lógica nos cursos de Tecnologia da Informação e Comunicação é desenvolver linguagens para modelar as situações do mundo e dos sistemas, de modo que possamos analisá-las formalmente, construindo argumentos sobre elas para serem apresentados e justificados rigorosamente. Nesta seção, apresentamos os conceitos da linguagem e do sistema de Dedução Natural da Lógica Proposicional e da Lógica de Predicados.

2.1 A Lógica Proposicional

A Lógica Proposicional se baseia em *proposições* ou *frases declarativas* que se pode argumentar sobre sua veracidade ou falsidade. As frases declarativas são consideradas como *atômicas*. Por exemplo, "*comprei um bilhete*" e "*ganhei na loteria*". Desta forma, podemos atribuir símbolos distintos $A, B, C, \cdots$ a cada uma destas frases:

- A : "*comprei um bilhete*",
- B : "*ganhei na loteria*".

Para codificar frases mais complexas usamos os conectivos lógicos: $\neg$ (negação), $\wedge$ (conjunção), $\vee$ (disjunção) e $\rightarrow$ (implicação) Huth e Ryan, 2004. Por exemplo:

- $A \wedge B$ codifica a frase "comprei um bilhete *e* ganhei na loteria";
- $A \vee B$ codifica a frase "comprei um bilhete *ou* ganhei na loteria.";
- $B \rightarrow A$ representa "*se* ganhei na loteria *então* comprei um bilhete." e;
- $\neg B$ representa "não ganhei na loteria".

2.1.1 Linguagem

O conjunto dos átomos proposicionais, juntamente com os conectivos e os símbolos '(', ')' formam o alfabeto da linguagem da Lógica Proposicional. A linguagem da Lógica Proposicional, que descreve quais as combinações de símbolos do alfabeto são consideradas fórmulas, pode ser definida por uma gramática na forma de *Backus Naur* (BNF - *Backus Naur Form*) como a seguir:

$$\varphi ::= \perp \mid A \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi)$$

em que $\perp$ e A representam, respectivamente, a contradição e qualquer proposição atômica e cada ocorrência φ a direta de '::=' representa qualquer fórmula já construída Huth e Ryan, 2004.

2.1.2 Dedução Natural

O sistema de Dedução Natural é um mecanismo que permite a construção de uma prova formal, estabelecendo uma conclusão φ a partir de um conjunto de premissas $\Gamma = \{\varphi_1, \varphi_2, \cdots, \varphi_n\}$, denotado por $\Gamma \vdash \varphi$, aplicando-se sucessivamente *regras de demonstração*. O artigo Pelletier, 2000 compara o estilo de demonstrações de Dedução Natural de 33 livros-texto, sendo o estilo *Fitch* (caixas) o adotado na maioria deles.

Neste estilo, as demonstrações são apresentadas de forma linear e sequencial, na qual cada uma das linhas da prova é numerada, tem uma afirmação e uma justificativa. As justificativas são definidas por serem *premissas* da prova ou pela aplicação de uma das *regras do sistema dedutivo*. As subprovas dentro de uma prova maior têm *caixas* ao redor e servem para delimitar o escopo de hipóteses temporárias. Provas podem ter caixas dentro de caixas, ou pode-se abrir outras caixas depois de fechar outras, obedecendo as regras de demonstração. Uma fórmula só pode ser utilizada em uma prova em um determinado ponto se essa fórmula aconteceu anteriormente e se nenhuma caixa que contenha essa ocorrência da fórmula tenha sido fechada.

Uma demonstração em Dedução Natural é construída a partir da enumeração das premissas e da aplicação das *regras de Dedução Natural*. Em geral, cada conectivo possui uma regra para adicionar e outra regra para eliminar este conectivo. A construção de uma demonstração é um exercício criativo. Não é óbvio que regras aplicar e em qual ordem para, a partir das premissas, obter a conclusão desejada. A seguir, serão apresentadas as regras de Dedução Natural.

Premissas O primeiro passo em uma demonstração em Dedução Natural no estilo *Fitch* é enumerar as premissas da prova. A Figura 1 apresenta a estrutura geral da regra das premissas, na qual $\varphi_1, \varphi_2, \ldots, \varphi_n$ são representadas em uma linha cada, seguindo uma numeração sequencial e como justificativa "premissa".

$$
\begin{array}{lll}
1. & \varphi_1 & \text{premissa} \\
2. & \varphi_2 & \text{premissa} \\
\vdots & \vdots & \vdots \\
\text{n.} & \varphi_n & \text{premissa} \\
\vdots & \vdots & \vdots
\end{array}
$$

Figura 1: Enumeração das premissas.

Conjunção Introdução ($\wedge i$) A regra da introdução da conjunção ($\wedge i$) é apresentada na Figura 2a (ou Figura 2b), na qual a fórmula $\varphi \wedge \psi$ pode ser concluída em uma linha p se φ e ψ foram demonstradas nas linhas m (ou n) e n (ou m), respectivamente, anteriores a linha p e que não foram descartadas. A Figura 2c exibe a aplicação $\wedge i$ da fórmula $A \wedge B$ na linha 3 a partir das fórmulas A e B, definidas nas linhas 1 e 2, respectivamente, que são anteriores a linha 3.

$$
\begin{array}{lll}
\vdots & \vdots & \vdots \\
m. & \varphi & \\
\vdots & \vdots & \vdots \\
n. & \psi & \\
\vdots & \vdots & \vdots \\
p. & \varphi \wedge \psi & \wedge\text{i } m,n
\end{array}
$$

(a) Regra $\wedge i$

$$
\begin{array}{lll}
\vdots & \vdots & \vdots \\
m. & \psi & \\
\vdots & \vdots & \vdots \\
n. & \varphi & \\
\vdots & \vdots & \vdots \\
p. & \varphi \wedge \psi & \wedge\text{i } m,n
\end{array}
$$

(b) Regra $\wedge i$

$$
\begin{array}{lll}
1. & A & \text{premissa} \\
2. & B & \text{premissa} \\
3. & A \wedge B & \wedge i \text{ 1,2}
\end{array}
$$

(c) $A, B \vdash A \wedge B$

Figura 2: Conjunção Introdução ($\wedge i$)

Conjunção Eliminação ($\wedge e$) A regra da eliminação da conjunção ($\wedge e$) é apresentada na Figura 3a (ou Figura 3b), na qual a fórmula φ (ou ψ) pode ser concluída na linha p a partir da eliminação à esquerda (ou à direita) da conjunção da fórmula $\varphi \wedge \psi$ da linha m (anterior a p e não foi descartada). A Figura 3c exibe uma aplicação da regra na qual A é obtida na linha 3 pela eliminação da conjunção à esquerda da fórmula $A \wedge B$ da linha 1.

Implicação Eliminação ($\rightarrow e$) A regra da eliminação da implicação ($\rightarrow e$), também conhecida como *Modus Ponens*, é apresentada na Figura 4a (ou Figura 4b), na qual a fórmula ψ pode ser concluída na linha p a partir da eliminação da implicação da fórmula $\varphi \rightarrow \psi$ da linha m (ou n) e φ da linha n (ou m), anteriores a p e não descartadas. A Figura 5b exibe uma aplicação da regra na qual a fórmula C é obtida na linha 4 pela eliminação da implicação das fórmulas $B \rightarrow C$ e B das linha 2 e 3, respectivamente.

$$\begin{array}{lll} \vdots & \vdots & \vdots \\ m. & \varphi \wedge \psi & \\ \vdots & \vdots & \vdots \\ p. & \varphi & \wedge\text{e } m \end{array}$$

(a) Regra $\wedge e$ à esquerda

$$\begin{array}{lll} \vdots & \vdots & \vdots \\ m. & \varphi \wedge \psi & \\ \vdots & \vdots & \vdots \\ p. & \psi & \wedge\text{e } m \end{array}$$

(b) Regra $\wedge e$ à direita

$$\begin{array}{lll} 1. & A \wedge B & \text{premissa} \\ 2. & C & \text{premissa} \\ 3. & A & \wedge e\ 1 \\ 4. & A \wedge C & \wedge i\ 3{,}2 \end{array}$$

(c) $A \wedge B, C \vdash A \wedge C$

Figura 3: Conjunção Eliminação ($\wedge e$)

$$\begin{array}{lll} \vdots & \vdots & \vdots \\ m. & \varphi \to \psi & \\ \vdots & \vdots & \vdots \\ n. & \varphi & \\ \vdots & \vdots & \vdots \\ p. & \psi & \to\text{e } m,n \end{array}$$

(a) Regra $\to e$

$$\begin{array}{lll} \vdots & \vdots & \vdots \\ m. & \varphi & \\ \vdots & \vdots & \vdots \\ n. & \varphi \to \psi & \\ \vdots & \vdots & \vdots \\ p. & \psi & \to\text{e } m,n \end{array}$$

(b) Regra $\to e$

$$\begin{array}{lll} 1. & A \wedge B & \text{premissa} \\ 2. & B \to C & \text{premissa} \\ 3. & B & \wedge e\ 1 \\ 4. & C & \to e\ 2{,}3 \end{array}$$

(c) $A \wedge B, B \to C \vdash C$

Figura 4: Implicação Eliminação ($\to e$)

Implicação Introdução ($\to i$) A regra da introdução da implicação ($\to i$) constrói condicionais que não ocorrem como premissas. Para construção de um condicional é necessário realizar *raciocínio hipotético*, isto é, supor *temporariamente* que uma dada fórmula é verdadeira. Chamamos esta fórmula de *hipótese*. Assim, utilizamos *caixas de demonstração*, que servem para delimitar o *escopo da hipótese temporária*. Observe na Figura 5a que para provar o condicional $\varphi \to \psi$ na linha $n+1$, colocamos φ como *hipótese* no topo de uma caixa (linha m), aplicamos um número finito de regras de forma a obter ψ na linha n. Todo o raciocínio para obter ψ depende da veracidade de φ e, por isso, as fórmulas resultante deste raciocínio ficam delimitadas na caixa. Na linha seguinte $(n+1)$ podemos aplicar a regra $\to i$ para obter $\varphi \to \psi$, sendo que este condicional não mais depende da hipótese φ. Na justificativa da linha $n+1$ utilizamos o nome da regra seguido das linhas inicial e final da caixa ($\to i$ m-n). A Figura 5b exibe uma aplicação da regra na qual a fórmula $A \to C$ é obtida na linha 6 a partir da caixa das linhas 3 a 5 em que A é a hipótese e C é a conclusão da caixa.

Observe que para a obtenção de ψ é possível utilizar quaisquer outras fórmulas como premissas e conclusões provisórias feitas até então. Demonstrações podem ter caixas dentro de caixas, ou pode-se abrir novas caixas depois de fechar outras. No entanto, existem regras sobre quais fórmulas podem ser utilizadas em que ponto na demonstração. Em geral, só podemos usar uma fórmula em um determinado ponto se esta fórmula ocorre *antes* desse ponto e se nenhuma caixa que contenha a ocorrência desta fórmula tenha sido fechada Huth e Ryan, 2004.

Disjunção Introdução A regra da introdução da disjunção ($\vee i$) é a apresentada na Figura 6a (ou Figura 6b), na qual dizemos que temos $\varphi \vee \psi$ em uma linha p se φ (ou ψ) ocorre em uma linha m

$\vdots$	$\vdots$	$\vdots$
$m.$	φ	hipótese
$\vdots$	$\vdots$	$\vdots$
$n.$	ψ	
$n+1.$	$\varphi \to \psi$	$\to$i m-n

(a) Regra $\to i$

1.	$A \to B$	premissa
2.	$B \to C$	premissa
3.	A	hipótese
4.	B	$\to e$ 3,1
5.	C	$\to e$ 4,2
6.	$A \to C$	$\to i$ 3-5

(b) $A \to B, B \to C \vdash A \to C$

Figura 5: Implicação Introdução ($\to i$)

anterior a p e que não foi descartada. A Figura 6c ilustra a aplicação da introdução da disjunção na linha 3 com a introdução de $A \vee B$ a partir da fórmula A definida na linha 2.

$\vdots$	$\vdots$	$\vdots$
m.	φ	
$\vdots$	$\vdots$	$\vdots$
p.	$\varphi \vee \psi$	$\vee$i m

(a) Regra $\vee i$ à esquerda

$\vdots$	$\vdots$	$\vdots$
m.	ψ	
$\vdots$	$\vdots$	$\vdots$
p.	$\varphi \vee \psi$	$\vee$i m

(b) Regra $\vee i$ à direita

1.	$(A \vee B) \to C$	premissa
2.	A	hipótese
3.	$A \vee B$	$\vee$ i 2
4.	C	$\to$ e 3,1
5.	$A \to C$	$\to$ i 2-4

(c) $(A \vee B) \to C \vdash A \to C$

Figura 6: Disjunção Introdução ($\vee i$)

Disjunção Eliminação ($\vee e$) a regra da disjunção eliminação ($\vee e$) é a apresentada na Figura 7a, na qual dizemos que podemos concluir uma fórmula α na linha $p+1$ se eliminarmos a disjunção da fórmula $\varphi \vee \psi$ na linha m e se fizermos a suposição de φ em uma caixa, na linha m, e a suposição de ψ em outra caixa, na linha $n+1$, tal que ambas as caixas tenham como conclusão α, nas linhas n e p, respectivamente, por meio de uma sequência finita de passos (regras). Note que essa regra se assemelha a introdução da implicação no sentido de que fazemos raciocínio hipotético, nas caixas de $(m+1)-n$ e $(n+1)-p$. A Figura 7b ilustra a aplicação da eliminação da disjunção na linha 8, na qual concluímos C, a partir da disjunção de $A \vee B$ na linha 3 e das caixas $4-5$ e $6-7$ onde supomos A na linha 4 e concluímos C na linha 5 e supomos B na linha 6 e concluímos C na linha 7.

Negação Eliminação ($\neg$e) A regra da negação eliminação ($\neg$e) envolve a noção de *contradição*. Contradições são expressões da forma $\varphi \wedge \neg\varphi$ ou $\neg\varphi \wedge \varphi$ onde φ é qualquer fórmula da Lógica Proposicional. A fórmula $\perp$ é utilizada para denotar uma contradição e este fato é expresso na regra $\neg e$. Na Figura 8a (ou Figura 8b) temos a fórmula φ na linha m (ou n) e a sua negação $\neg\varphi$ na linha n (ou m) sendo combinadas para aparecimento da contradição $\perp$ na linha p com a aplicação

⋮	⋮	⋮
m.	$\varphi \vee \psi$	
m+1.	φ	hipótese
⋮	⋮	⋮
n.	α	
n+1.	ψ	hipótese
⋮	⋮	⋮
p.	α	
p+1.	α	$\vee$e m, (m+1)-n, (n+1)-p

(a) Regra $\vee e$

1.	$A \rightarrow C$	premissa
2.	$B \rightarrow C$	premissa
3.	$A \vee B$	premissa
4.	A	hipótese
5.	C	$\rightarrow$ e 4,1
6.	B	hipótese
7.	C	$\rightarrow$ e 6,2
8.	C	$\vee$ e 3, 4-5, 6-7

(b) $A \rightarrow C, B \rightarrow C, A \vee B \vdash C$

Figura 7: Disjunção Eliminação ($\vee e$)

da regra $\neg e$. A Figura 8c exibe uma aplicação da regra na qual a contradição $\perp$ é obtida na linha 3 devido às fórmulas A na linha 1 e $\neg A$ na linha 2.

⋮	⋮	⋮
$m.$	φ	
⋮	⋮	⋮
$n.$	$\neg\varphi$	
⋮	⋮	⋮
$p.$	$\perp$	$\neg$e m,n

(a) Regra $\neg e$

⋮	⋮	⋮
$m.$	$\neg\varphi$	
⋮	⋮	⋮
$n.$	φ	
⋮	⋮	⋮
$p.$	$\perp$	$\neg$e m,n

(b) Regra $\neg e$

1.	A	premissa
2.	$\neg A$	premissa
3.	$\perp$	$\neg e$ 1,2

(c) $A, \neg A \vdash \perp$

Figura 8: Negação Eliminação ($\neg e$)

Negação Introdução ($\neg i$) A regra da negação introdução ($\neg i$) é a apresentada na Figura 9a. Esta é uma regra que envolve raciocínio hipotético e contradição. Se tomarmos φ como hipótese (linha m) e, após a aplicação de um número finito de regras, chegarmos a uma contradição $\perp$ na linha n, significa que a hipótese não pode ser verdadeira. Desse modo, finalizamos nosso raciocínio hipotético introduzindo a negação na hipótese e obtendo $\neg\varphi$ na linha $n+1$. A Figura 9b mostra um exemplo da aplicação da regra $\neg i$ em para provamos $\neg A$ na linha 6, assumimos A como hipótese no topo da caixa na linha 3 e chegamos a uma contradição no final da caixa na linha 5.

Contradição Eliminação ($\perp e$): A regra da contradição eliminação ($\perp e$) é exibida na Figura 10a, na qual podemos concluir uma fórmula qualquer ψ na linha n se demonstramos em uma linha m, anterior a n, a contradição. A Figura 10b apresenta a demonstração de B na linha 4 a partir da eliminação da contradição $\perp$ da linha 3.

$\vdots$	$\vdots$	$\vdots$
$m.$	φ	hipótese
$\vdots$	$\vdots$	$\vdots$
$n.$	$\bot$	
$n+1.$	$\neg\varphi$	$\neg$i m-n

(a) Regra $\neg i$

1.	$A \to B$	premissa
2.	$\neg B$	premissa
3.	A	hipótese
4.	B	$\to e$ 1,3
5.	$\bot$	$\neg e$ 2,4
6.	$\neg A$	$\neg i$ 3-5

(b) $A \to B, \neg B \vdash \neg A$

Figura 9: Negação Introdução ($\neg i$)

$\vdots$	$\vdots$	$\vdots$
$m.$	$\bot$	
$\vdots$	$\vdots$	$\vdots$
$n.$	ψ	$\bot$e m

(a) Regra $\bot e$

1.	A	hipótese
2.	$\neg A$	hipótese
3.	$\bot$	$\neg$ e 1,2
4.	B	$\bot$e 3
5.	$\neg A \to B$	$\to$ i 2-4
6.	$A \to (\neg A \to B)$	$\to$ i 1-5

(b) $\vdash A \to (\neg A \to B)$

Figura 10: Contradição Eliminação ($\bot e$)

Redução ao Absurdo (raa): A regra de redução ao absurdo é apresenta na Figura 11a, na qual para provarmos uma fórmula φ em uma linha $n+1$, iremos supor temporariamente a negação da fórmula, $\neg\varphi$, em uma caixa que inicia na linha m e que conclui a contradição, $\bot$, na linha n, após uma sequência de aplicações de regras. A Figura 11b exibe a demonstração de $A \vee \neg A$, também conhecido como terceiro-excluído. Para provarmos $A \vee \neg A$ na linha 8, fazemos a suposição de $\neg(A \vee \neg A)$, na linha 1 (início da caixa) e concluímos a contradição $\bot$, na linha 7 (fim da caixa).

Copie: A regra copie, apresentada na Figura 12a, é necessária, no estilo de Fitch, para permitir concluir uma caixa com uma fórmula que já apareceu anteriormente na demonstração. Por exemplo, para demonstrar $A \to B$, veja Figura 12b, na linha 10, é preciso que a caixa que justifica a introdução da implicação inicie com A, linha 8, e termine com B, linha 9. Ocorre que a justificativa de B já havia sido realizada e, portanto, a justificativa da linha 9 é a cópia da fórmula da linha 7.

2.2 Lógica de Predicados

A Lógica de Predicados surgiu da necessidade de representar aspectos mais ricos da linguagem natural do que apenas as frases declarativas da Lógica Proposicional. Esta lógica foi desenvolvida de forma independente por Gottlob Frege e Charles Sanders Peirce Ferreirós, 2001; Hammer, 1998. As fórmulas da lógica de predicados são compostas por predicados, constantes, variáveis,

$$
\begin{array}{lll}
\vdots & \vdots & \vdots \\
\text{m.} & \neg\varphi & \text{hipótese} \\
\vdots & \vdots & \vdots \\
\text{n.} & \bot & \\
\text{n+1.} & \varphi & \text{raa m-n}
\end{array}
$$

(a) Regra raa

1.	$\neg(A \vee \neg A)$	hipótese
2.	$\neg A$	hipótese
3.	$A \vee \neg A$	$\vee$ i 2
4.	$\bot$	$\neg$ e 3,1
5.	A	raa 2-4
6.	$A \vee \neg A$	$\vee$ i 5
7.	$\bot$	$\neg$ e 6,1
8.	$A \vee \neg A$	raa 1-7

(b) $\vdash A \vee \neg A$

Figura 11: Redução ao Absurdo (raa)

$$
\begin{array}{lll}
\vdots & \vdots & \vdots \\
\text{m.} & \varphi & \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots \\
\text{n.} & \varphi & \text{copie m} \\
\vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots
\end{array}
$$

(a) Regra copie

1.	$\neg A \vee B$	premissa
2.	$\neg A$	hipótese
3.	A	hipótese
4.	$\bot$	$\neg$ e 2,3
5.	B	$\bot$ e 4
6.	$A \to B$	$\to$ i 3-5
7.	B	hipótese
8.	A	hipótese
9.	B	copie 7
10.	$A \to B$	$\to$ i 8-9
11.	$A \to B$	$\vee$ e 1, 2-6, 7-10

(b) $\neg A \vee B \vdash A \to B$

Figura 12: Regra Copie

símbolos funcionais, quantificadores e conectivos.

Um *predicado* é um símbolo que representa uma propriedade ou uma relação entre objetos do mundo. Por exemplo, na fórmula $Estudante(maria)$ o predicado *Estudante* expressa a propriedade de que um símbolo constante *maria* é uma estudante. Já na fórmula *Professor(joão, maria)* o predicado *Professor* relaciona as constantes *joão* e *maria*, representando que *joão* é professor de *maria*. Essa relação também se dá para variáveis de um certo domínio. Além disso, a Lógica de Predicados inclui *quantificadores* com os quais é possível representar a noção de "*existe*" (*existe um x que é estudante*) e "*para todo*" (*todos os x são estudantes*). Também é possível representar *símbolos funcionais* que são associados a objetos do mundo. Por exemplo, *Professor(maria)* é uma função que devolve a constante *joão*.

2.2.1 Linguagem

Devido ao seu maior poder de expressividade, a linguagem da Lógica de Predicados é mais complexa do que a da Lógica Proposicional. Nesta lógica há dois tipos de expressões: os termos e as fórmulas. Termos são expressões que relacionam-se a objetos de um domínio: indivíduos como *joão* e *maria* são exemplos; variáveis como x e; símbolos funcionais que também referem-se a objetos, como exemplo *Professor(maria)* refere-se ao objeto *joão*. Já as fórmulas são expressões que podem ser relacionadas a um valor verdade: verdadeiro ou falso.

O conjunto dos termos da linguagem da Lógica de Predicados pode ser definidos por uma gramática na forma BNF Huth e Ryan, 2004:

$$t ::= x \mid c \mid f(t, \cdots, t),$$

em que x é uma variável, c é uma constante e f é um símbolo funcional com aridade $n > 0$.

Já as fórmulas da Lógica de Predicados podem ser definidas em BNF, como a seguir Huth e Ryan, 2004:

$$\varphi ::= \perp \mid P(t_1, t_2, \cdots, t_n), \mid (\neg\varphi) \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \to \varphi) \mid (\forall x\, \varphi) \mid (\exists x\, \varphi),$$

em que $\perp$ e P representam, respectivamente, a contradição e um símbolo predicado n-ário com $n \geq 0$, $t_1, \cdots t_n$ são termos e x é uma variável.

Nas fórmulas da Lógica de Predicados, as variáveis podem ocorrer *livres* ou *ligadas*. Considere a seguinte fórmula φ:

$$(\exists x(P(x) \wedge Q(x)) \vee R(x,y).$$

Dizemos que a variável x é *livre* em φ se ela não é capturada por um quantificador ($\forall x$ ou $\exists x$) em φ, isto é, se a variável x não está no escopo de um quantificador $\forall x$ ou $\exists x$. Caso contrário, dizemos que a variável é ligada na fórmula φ. Na fórmula anterior, as duas primeiras ocorrências da variável x são *ligadas* (estão no escopo do quantificador $\exists x$) e a terceira ocorrência desta variável é *livre* (não está no escopo do quantificador $\exists x$). Já a única ocorrência da variável y é *livre*.

Definimos φ_t^x como a expressão obtida da fórmula φ pela substituição da variável x, sempre que ela ocorrer livre em φ, pelo termo t. Por exemplo, $(H(x) \to \forall x M(x))_y^x = (H(y) \to \forall x M(x))$. Dizemos que o termo t é substituível para x em φ se não existe uma variável y em t tal que ela é capturada por um ($\forall y$ ou $\exists y$) quantificador de φ_t^x. Por exemplo, o termo z é substituível para y em $\forall x P(x,y)$. Por outro lado, o termo x não é substituível para y em $\forall x P(x,y)$.

2.2.2 Dedução Natural

As demonstrações de Dedução Natural para a Lógica de Predicados são semelhantes às demonstrações para a Lógica Proposicional com a adição de novas regras para tratar dos *quantificadores*. Deste modo, qualquer regra de Dedução Natural para a Lógica Proposicional pode ser aplicada também para as fórmulas da Lógica de Predicados. A seguir, serão descritas as regras para introdução e eliminação dos quantificadores universal e existencial.

Eliminação do Universal ($\forall e$): A regra da eliminação do universal ($\forall e$) é apresentada na Figura 13a, na qual a fórmula φ_t^x pode ser concluída em uma linha p se $\forall x \varphi$ foi demonstrada na linha m, desde que o termo t seja substituível pela variável x em φ. A Figura 13b exibe a aplicação $\forall e$ da fórmula $H(s) \rightarrow M(s)$ na linha 3 a partir da fórmula $\forall x(H(x) \rightarrow M(x))$, definida na linha 1.

$$\boxed{\begin{array}{lll} \vdots & \vdots & \vdots \\ \text{m.} & \forall x\varphi & \\ \vdots & \vdots & \vdots \\ \text{p.} & \varphi_t^x & \forall\text{e m} \end{array}}$$

t é substituível para x em φ

(a) Regra Universal Eliminação

$$\begin{array}{lll} 1. & \forall x(H(x) \rightarrow M(x)) & \text{premissa} \\ 2. & H(s) & \text{premissa} \\ 3. & H(s) \rightarrow M(s) & \forall e\ 1 \\ 4. & M(s) & \rightarrow e\ 2{,}3 \end{array}$$

(b) $\forall x(H(x) \rightarrow M(x)), H(s) \vdash M(s)$

Figura 13: Regra Universal Eliminação ($\forall e$)

Introdução do Universal ($\forall i$): A regra da introdução do universal ($\forall i$) é apresentada na Figura 14a, na qual para provarmos $\forall x \varphi$, na linha $n+1$, iremos supor que para uma variável a qualquer (arbitrária) em uma caixa que inicia na linha m e que conclui φ_a^x, na linha n. Dizemos que a variável a é qualquer se ela é uma variável nova na linha m, ou seja, a não ocorre como variável livre de qualquer fórmula que aconteça anteriormente a linha m que não esteja em uma caixa fechada. Na Figura 14b, ilustramos a introdução do universal na linha 7 para provar $\forall x M(x)$ a partir da escolha de a, no início da caixa, na linha 3, e demonstramos $M(a)$ ao final da caixa, na linha 6. Note que a variável a escolhida não é uma variável livre das fórmulas das linhas 1 e 2.

$$\boxed{\begin{array}{ll} \vdots & \begin{array}{ll} \vdots & \quad\vdots \end{array} \\ \text{m.} & \boxed{\begin{array}{ll} a & \\ \vdots & \vdots \\ \varphi_a^x & \end{array}} \\ \vdots & \\ \text{n.} & \\ \text{n+1.} & \forall x\varphi \quad \forall\text{i m-n} \end{array}}$$

a é uma variável nova

(a) Universal Introdução ($\forall i$)

$$\begin{array}{lll} 1. & \forall x(H(x) \rightarrow M(x)) & \text{premissa} \\ 2. & \forall x H(x) & \text{premissa} \\ 3. & \boxed{\begin{array}{lll} a & & \\ & H(a) \rightarrow M(a) & \forall e\ 1 \\ & H(a) & \forall e\ 2 \\ & M(a) & \rightarrow e\ 5{,}4 \end{array}} & \\ 4.\text{–}6. & & \\ 7. & \forall x M(x) & \forall i\ 3\text{-}6 \end{array}$$

(b) $\forall x(H(x) \rightarrow M(x)), \forall x H(x) \vdash \forall x M(x)$

Figura 14: Regra Universal Introdução ($\forall i$)

Introdução do Existencial ($\exists i$): A regra da introdução do existencial ($\exists i$) é apresentada na Figura 15a, na qual a fórmula $\exists x \varphi$ pode ser concluída em uma linha p se φ_t^x foi demonstrada na linha m, desde que o termo t seja substituível para a variável x em φ. A Figura 15b exibe a aplicação $\exists i$ da fórmula $\exists x P(x)$ na linha 3 a partir da fórmula $P(a)$, definida na linha 1.

$\vdots$	$\vdots$	$\vdots$
m.	φ_t^x	
$\vdots$	$\vdots$	$\vdots$
p.	$\exists x\varphi$	$\exists$i m

t é substituível para x em φ

(a) Existencial ($\exists i$)

1.	$P(a)$	premissa
2.	$\exists xP(x) \rightarrow B$	premissa
3.	$\exists xP(x)$	$\exists$ i 1
4.	B	$\rightarrow$ e 3,2

(b) $P(a), \exists xP(x) \rightarrow B \vdash B$

Figura 15: Regra Existencial Introdução ($\exists i$)

Eliminação do Existencial ($\exists e$): A regra da eliminação do existencial ($\exists e$) é apresentada na Figura 16a, na qual para provarmos uma fórmula α, na linha $p+1$, iremos eliminar o existencial da fórmula $\exists x\varphi$, na linha m, supondo a fórmula φ_a^x para alguma variável a em uma caixa que inicia na linha n e que concluiu com uma fórmula α ao final da caixa, na linha p, desde que a não ocorra na conclusão α. Nesta regra sabemos que a fórmula φ vale para algum elemento. Entretanto, não podemos assumir nenhuma propriedade específica para esta variável. Assim, a variável a deve ser uma variável nova na linha n, ou seja, a não ocorre como variável livre de qualquer fórmula que aconteça anteriormente a linha n que não esteja em uma caixa fechada e nem pode estar na conclusão α da regra. Na Figura 16b, ilustramos a eliminação do existencial para concluir $\perp$ na linha 9, a partir da fórmula $\exists xH(x)$, na linha 3, e pela caixa que inicia na linha 4 com a suposição da fórmula $H(a)$ com a (nova) variável a e que termina a caixa na linha 8 com a conclusão $\perp$.

$\vdots$	$\vdots$	$\vdots$
m.	$\exists x\varphi$	
$\vdots$	$\vdots$	$\vdots$
n.	a φ_a^x	hipótese
$\vdots$	$\vdots$	$\vdots$
p.	α	
p+1.	α	$\exists$e m, n-p

a é uma variável nova

a não ocorre em α

(a) Existencial Eliminação ($\exists e$)

1.		$\forall x(H(x) \rightarrow M(x))$	premissa
2.		$\forall x\neg M(x)$	premissa
3.		$\exists xH(x)$	hipótese
4.	a	$H(a)$	hipótese
5.		$H(a) \rightarrow M(a)$	$\forall e$ 1
6.		$M(a)$	$\rightarrow e$ 4,5
7.		$\neg M(a)$	$\forall e$ 2
8.		$\perp$	$\neg e$ 6,7
9.		$\perp$	$\exists e$ 3, 4-8
10.		$\neg\exists xH(x)$	$\neg i$ 3-9

(b) $\forall x(H(x) \rightarrow M(x)), \forall x\neg M(x) \vdash \neg\exists xH(x)$

Figura 16: Regra Existencial Eliminação ($\exists e$)

3 O Assistente de Provas NADIA

O assistente de provas NADIA, *Natural Deduction Proof Assistant*, é uma ferramenta que permite verificar automaticamente a correção de uma demonstração no sistema de Dedução Natural no estilo Fitch. NADIA[1] foi desenvolvido na linguagem Python, pode ser utilizado em *desktop*, ou em uma plataforma *Web*. A ferramenta encontra-se disponível em https://sistemas.quixada.ufc.br/nadia/ e não é necessário realizar cadastro para uso. Além disso, o assistente de provas NADIA pode ser acoplado ao Moodle, permitindo assim uma melhor integração e prática com o restante do conteúdo da disciplina de Lógica.

A ideia é que a demonstração seja escrita de maneira mais próxima possível da prova em papel, permitindo que estudantes possam escrever sua demonstração em texto simples e verificar a correção de sua demonstração. Caso a prova esteja errada, NADIA apresentará quais são as regras que foram utilizadas de forma errada.

- Os átomos são escritos em letras maiúsculas (e.g. A, B, H(x));
- Variáveis são escritas com a primeira letra em minúsculo, seguido por letras ou números (e.g. x, x0);
- Formulas com $\forall x$ e $\exists x$ são representadas por Ax e Ex ('A' e 'E' seguidos da variável x). Por exemplo, Ax (H(x)−>M(x)) representa $\forall x\ (H(x) \to M(x))$.
- A Figura 17 apresenta a correspondência entre os símbolos da lógica e os que são utilizados no NADIA.
- A ordem de precedência dos quantificadores e conectivos é definida por $\neg, \forall, \exists, \wedge, \vee, \to$ com alinhamento à direita. Por exemplo:
  - A formula ∼A&B−>C representa a formula $(((\neg A) \wedge B) \to C)$;
  - O teorema ∼A|B |− A−>C representa $((\neg A) \vee B) \vdash (A \to C)$.
- Cada regra de inferência é rotulada pelo seu respectivo conectivo/quantificador, o símbolo *i* para introdução ou *e* para eliminação. Por exemplo, &i representa a introdução da conjunção.
- As justificativas para as premissas e para as hipóteses utilizam as palavras reservadas pre e hip, respectivamente.

Símbolo	$\neg$	$\wedge$	$\vee$	$\to$	$\forall x$	$\exists x$	$\perp$	premissa	hipótese	caixa	$\vdash$
LaTeX	\lnot	\land	\lor	\rightarrow	\forall x	\exists x	\bot	premissa	hipótese	caixa	\vdash
NADIA	∼	&	|	−>	Ax	Ex	@	pre	hip	{}	|−

Figura 17: Correspondência entre os símbolos da lógica, NADIA e Latex

[1] O código-fonte do NADIA está disponível em https://github.com/daviromero/nadia

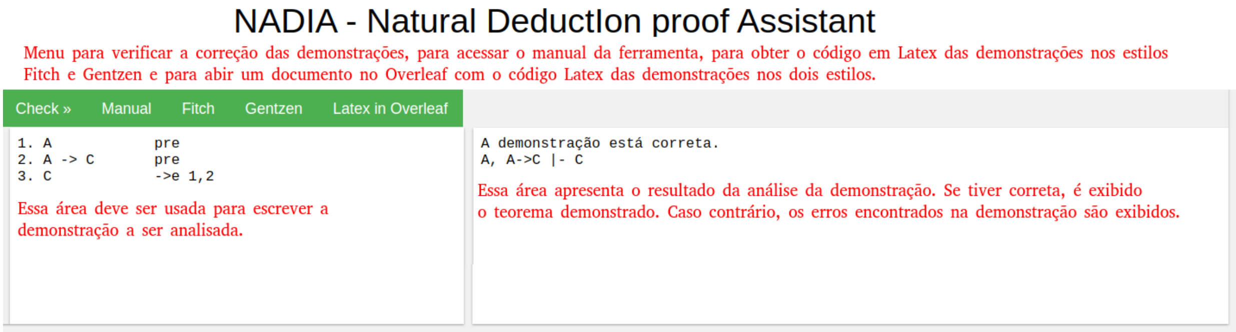


Figura 18: Visão Geral da versão *Web* de NADIA.

A partir de uma demonstração correta, é possível gerar código *Latex* nos estilos de demonstração: Fitch (caixas) utilizando o pacote logicproof e Gentzen (árvores) utilizando o pacote proof.

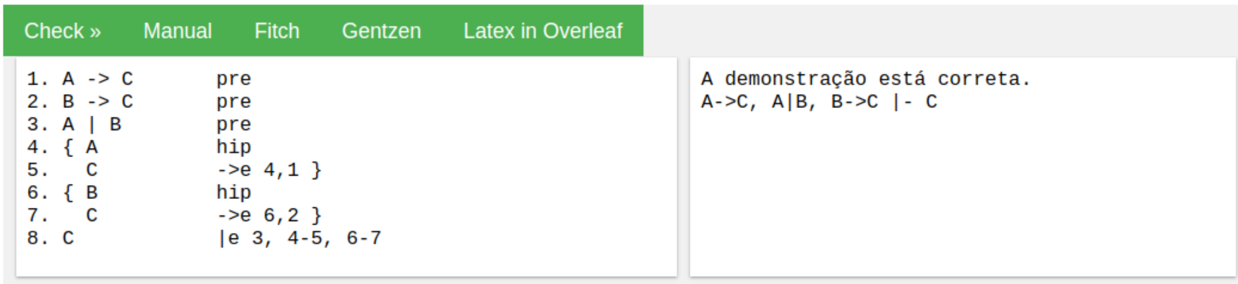
A Figura 18 apresenta a visão geral da versão *Web* do assistente NADIA. A ferramenta tem uma área para edição da demonstração em texto simples, uma seção para o resultado da análise da demonstração e os seguintes *links*: *Check*, para verificar a correção da demonstração; *Manual*, para visualizar um documento com as regras e exemplos de demonstração; *Fitch*, para gerar o código Latex no estilo Fitch; *Gentzen*, para gerar o código *Latex* no estilo de Gentzen e; *Latex em Overleaf*, para abrir um documento no *Overleaf* contendo o código Latex nos estilos *Fitch* e *Gentzen*. Os *links Fitch*, *Gentzen* e *Latex em Overleaf* são exibidos apenas após a verificação da correção de uma demonstração.

A Figura 19a apresenta a demonstração de $A \rightarrow C, B \rightarrow C, A \vee B \vdash C$, em Lógica Proposicional no estilo de Fitch. A partir da verificação de que essa demonstração está correta, é possível gerar o código em Latex desta prova no estilo Fitch (Figura 19b) e no estilo de Gentzen (Figura 19c). A Figura 19d ilustra uma prova incorreta deste teorema. Note que na linha 7 o usuário usou a fórmula A da linha 4. Ocorre que essa fórmula estava no escopo de uma caixa (linhas $4-5$) que já havia sido fechada. Portanto, a fórmula A já havia sido descartada e não poderia ser utilizada na linha 7.

Já a Figura 20 apresenta a demonstração de $\forall x(Q(x) \rightarrow R(x)), \exists x(P(x) \wedge Q(x)) \vdash \exists x(P(x) \wedge R(x))$, em Lógica de Predicados no estilo de Fitch. A partir da verificação de que essa demonstração está correta, é possível gerar o código em Latex desta prova no estilo Fitch (Figura 20b) e no estilo de Gentzen (Figura 20c). A Figura 20d ilustra uma prova incorreta deste teorema. Note que na linha 9 o usuário retirou da caixa a fórmula $P(x0) \wedge R(x0)$ usando a regra do existencial eliminação. No entanto, a regra não permite que na fórmula retirada da caixa ocorra a variável $x0$.

Disponibilizamos uma *playlist* no Youtube[2] com explicações sobre as regras de Dedução Natural e exemplos de resolução de exercícios na ferramenta NADIA em Lógica Proposicional, como pode ser visto em https://youtu.be/Y7yMuRna-5E, e Lógica de Predicados, como pode ser visto em https://youtu.be/HgjjJ930Hwo.

[2]A playlist está disponível em https://youtube.com/playlist?list=PLfOnKvd6pFiogX1TEA5ZTzZLoC9H-hx9R



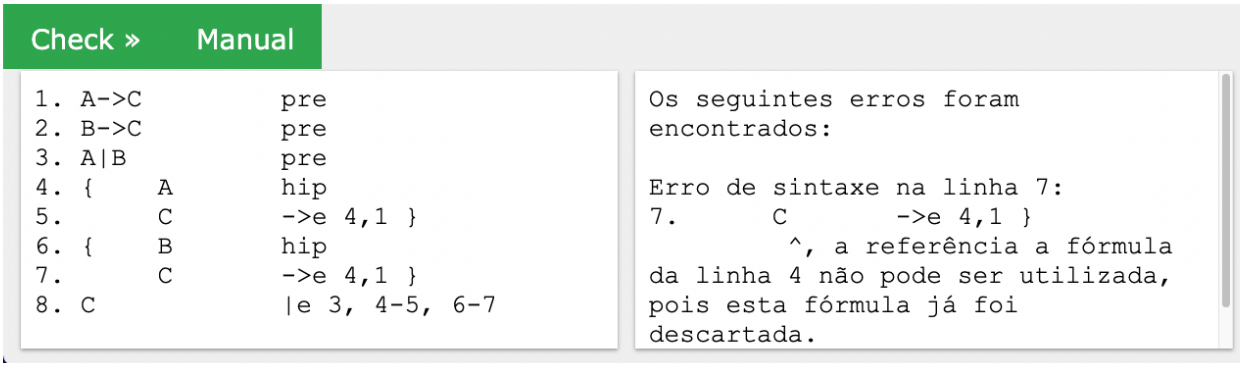
(a) Demonstração correta em Lógica Proposicional.

1.	$A \rightarrow C$	premissa
2.	$B \rightarrow C$	premissa
3.	$A \vee B$	premissa
4.	A	hipótese
5.	C	$\rightarrow e$ 4,1
6.	B	hipótese
7.	C	$\rightarrow e$ 6,2
8.	C	$\vee e$ 3, 4-5, 6-7

(b) Estilo Fitch

$$\dfrac{A \vee B \qquad \dfrac{[A]^1 \quad A \rightarrow C}{C}\rightarrow e \qquad \dfrac{B \rightarrow C \quad [B]^2}{C}\rightarrow e}{C}\vee e^{1,2}$$

(c) Estilo Gentzen



(d) Demonstração incorreta em Lógica Proposicional com mensagem de erro apresentada pelo NADIA.

Figura 19: NADIA - Demonstração de $A \rightarrow C, B \rightarrow C, A \vee B \vdash C$

```
Check »   Manual   Fitch   Gentzen   Latex in Overleaf

1. Ax(Q(x) -> R(x))              pre
2. Ex(P(x) & Q(x))               pre
3. {x0   P(x0) & Q(x0)           hip
4.       Q(x0) -> R(x0)          Ae 1
5.       P(x0)                   &e 3
6.       Q(x0)                   &e 3
7.       R(x0)                   ->e 6,4
8.       P(x0) & R(x0)           &i 5,7
9.       Ex (P(x) & R(x))        Ei 8 }
10. Ex (P(x) & R(x))             Ee 2, 3-9
```

```
A demonstração está correta.
Ax (Q(x)->R(x)), Ex (P(x)&Q(x)) |- Ex (P(x)&R(x))
```

(a) Demonstração correta em Lógica de Predicados.

1.		$\forall x(Q(x) \to R(x))$	premissa
2.		$\exists x(P(x) \land Q(x))$	premissa
3.	$x0$	$P(x0) \land Q(x0)$	hipótese
4.		$Q(x0) \to R(x0)$	$\forall e$ 1
5.		$P(x0)$	$\land e$ 3
6.		$Q(x0)$	$\land e$ 3
7.		$R(x0)$	$\to e$ 6, 4
8.		$P(x0) \land R(x0)$	$\land i$ 5,7
9.		$\exists x(P(x) \land R(x))$	$\exists i$ 8
10.		$\exists x(P(x) \land R(x))$	$\exists e$ 2,3-9

(b) Demonstração em Lógica de Predicados no Estilo Fitch

$$\dfrac{\exists x(P(x)\land Q(x)) \qquad \dfrac{\dfrac{\dfrac{[P(x0)\land Q(x0)]^1}{P(x0)}\land e \qquad \dfrac{\dfrac{[P(x0)\land Q(x0)]^1}{Q(x0)}\land e \qquad \dfrac{\forall x(Q(x)\to R(x))}{Q(x0)\to R(x0)}\forall e}{R(x0)}\to e}{P(x0)\land R(x0)}\land i}{\exists x(P(x)\land R(x))}\exists i}{\exists x(P(x)\land R(x))}\exists e^1$$

(c) Demonstração em Lógica de Predicados no Estilo Gentzen

```
Check »   Manual

1. Ax(Q(x) -> R(x))              pre
2. Ex(P(x) & Q(x))               pre
3. {x0   P(x0) & Q(x0)           hip
4.       Q(x0) -> R(x0)          Ae 1
5.       P(x0)                   &e 3
6.       Q(x0)                   &e 3
7.       R(x0)                   ->e 6,4
8.       P(x0) & R(x0)           &i 5,7 }
9.  P(x0) & R(x0)                Ee 2, 3-8
10. Ex (P(x) & R(x))             Ei 9
```

```
Os seguintes erros foram encontrados:

Erro de sintaxe na linha 9:
9.  P(x0) & R(x0)           Ee 2, 3-8
                   ^, A variável utilizada na caixa que inicia na linha
3 não pode ocorrer como variável livre na conclusão da fórmula e, portanto,
não pode ser utilizada nesta regra.
```

(d) Demonstração incorreta em Lógica de Predicados com mensagem de erro apresentada pelo NADIA.

Figura 20: NADIA - Demonstração de $\forall x(Q(x) \to R(x)), \exists x(P(x) \land Q(x)) \vdash \exists x(P(x) \land R(x))$

4 Trabalhos Correlatos

Na literatura, é possível encontrar algumas ferramentas destinadas a auxiliar o ensino e aprendizagem do conteúdo de Dedução Natural em Lógica Proposicional e Lógica de Predicados. A pesquisa por tais ferramentas foi realizada utilizando a plataforma Google Scholar, com a consulta "*proof assistant natural deduction*". Como resultado dessa busca, identificamos algumas ferramentas relevantes, incluindo NaDea, PANDA e ProofWeb. Além disso, durante a exploração dos artigos resultantes dessa pesquisa, foram também identificadas ferramentas adicionais, como Carnap.io e JAPE.

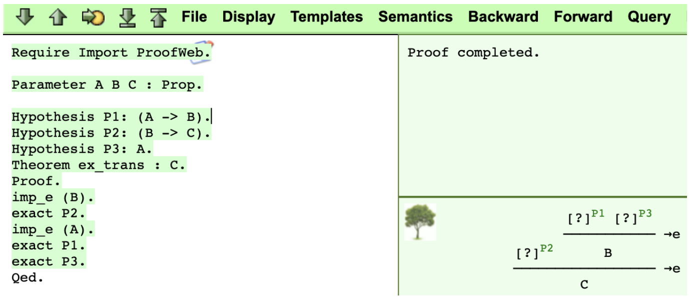
A seguir, resumimos as funcionalidades de alguns assistentes de provas, bem como destacamos as semelhanças e diferenças entre estas ferramentas e a proposta neste trabalho. Também fornecemos exemplos de como realizar a prova $A \rightarrow B, B \rightarrow C, A \vdash C$, utilizando cada uma dessas ferramentas.

- JAPE[3] Bornat e Sufrin, 1996 é um assistente de provas *desktop* desenvolvido em Java para a construção de provas no estilo *Fitch*. As provas são realizadas por meio da inserção das regras de inferência usando a *interface gráfica*. No entanto, a interface não permite o uso errado de uma regra e não é possível escrever em texto e verificar a correção da prova.
- ProofWeb[4] Maxim et al., 2010 é um assistente de provas *web* que pretende ser uma evolução do JAPE. Utiliza o Coq[5], que é um assistente estado-da-arte para escrever provas matemáticas. O usuário deve seguir uma *sintaxe própria* ou usar a *interface gráfica* para adicionar as regras de inferência. O ProofWeb permite exibir as provas nos estilos *Fitch* e *Gentzen*.
- PANDA[6] Gasquet et al., 2011 também é um assistente de provas *Web* desenvolvido em Java que se diferencia dos anteriores por permitir a elaboração de provas no estilo de *Gentzen* a partir de sua *interface gráfica*.
- NaDeA[7] Villadsen et al., 2018 é um assistente de provas para Dedução Natural com uma formalização em Isabelle. O usuário deve escrever suas provas usando a interface gráfica da ferramenta.
- Carnap.io[8] Leach-Krouse, 2018 é um arcabouço código aberto escrito em Haskell para criar e explorar linguagens formais, lógicas e semânticas. Um assistente de provas para Dedução Natural que é capaz de avaliar a corretude de provas em lógica proposicional e lógica de predicados. O usuário pode escrever as provas em texto usando o estilo Fitch.

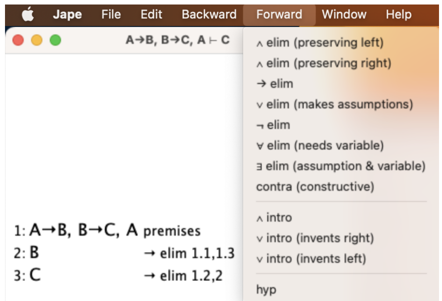
Para fins de ensino e aprendizagem de Dedução Natural em Lógica é muito importante que a ferramenta possibilite que o aluno escreva provas o mais parecido possível com o que é disponibilizado na literatura e com o que usualmente faria no papel. No lado esquerdo da Figura 21a

[3] O código-fonte está disponível em https://github.com/RBornat/jape/
[4] O ProofWeb não foi encontrado na url http://prover.cs.ru.nl/ referenciada no artigo.
[5] Coq está disponível em: https://coq.inria.fr/
[6] PANDA não foi encontrado na url https://www.irit.fr/panda/ referenciada no artigo.
[7] Ferramenta disponível em https://nadea.compute.dtu.dk/
[8] Ferramenta disponível em https://carnap.io

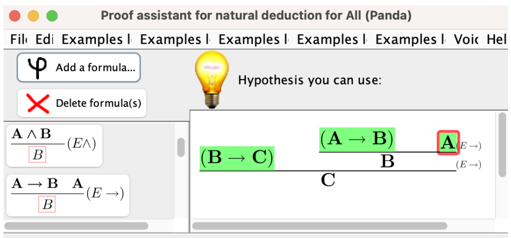
podemos visualizar a escrita da prova de $A \to B, B \to C, A \vdash C$ na ferramenta ProofWeb. Observe que há uma sintaxe própria da ferramenta que o aluno precisa aprender para construção da prova. Essa sintaxe difere muito do que o aluno escreveria no papel. Nesse sentido, o NADIA foi desenvolvido para que o estudante escreva provas da forma mais parecida possível ao que escreveria no papel. Por outro lado, as demonstrações no JAPE (Figura 21b), PANDA (Figura 21c) e NaDeA (Figure 21d) são realizadas por meio da aplicação de regras na interface gráfica. A ferramenta Carnap.io (Figura 21e), assim como o NADIA, também permite que o usuário digite uma demonstração textual bem similar ao que faria no papel.



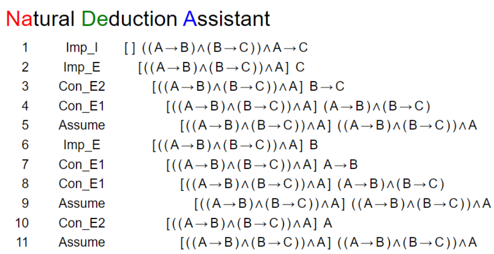
(a) ProofWeb: $A \to B, B \to C, A \vdash C$



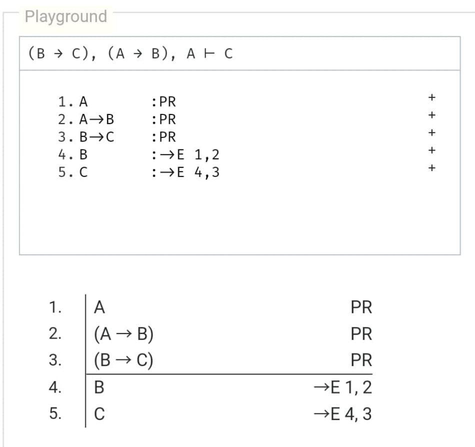
(b) JAPE: $A \to B, B \to C, A \vdash C$



(c) Panda: $A \to B, B \to C, A \vdash C$



(d) NaDeA: $A \to B, B \to C, A \vdash C$



(e) Carnap.io: $A \to B, B \to C, A \vdash C$

Figura 21: Provas utilizando assistentes de Dedução Natural da literatura.

A Tabela 1 exibe um comparativo entre os assistentes de provas para Dedução Natural apresentados nesta seção com relação ao tipo de estilo de demonstração disponível na ferramenta (F para estilo Fitch-style e G for estilo Gentzen) e a forma de entrada para a escrita da prova (GUI para escrita de provas baseada em cliques na interface gráfica e PT para escrita de provas no formato textual).

	ProofWeb	Jape	PANDA	NaDeA	NADIA	Carnap.io
Estilo de Prova	F, G	F	G	F	F, G	F
Forma de Escrita das Provas	GUI, PT	GUI	GUI	GUI	PT	PT

Tabela 1: Comparativo entre os assistentes de provas de Dedução Natural.

5 Integração do NADIA ao Ambiente Virtual de Aprendizagem Moodle

Esta seção tem por objetivo descrever o processo de integração da ferramenta NADIA ao Ambiente Virtual de Aprendizagem (AVA) Moodle[9], já utilizado pelos alunos e professores do Campus da Universidade Federal do Ceará em Quixadá, principalmente, para disponibilização de material didático, propostas de exercícios e comunicação entre professor e turma.

A possibilidade de ter uma ferramenta capaz de corrigir automaticamente exercícios de Dedução Natural fez com que o NADIA tivesse muita utilidade, do ponto de vista do aluno, no período de aulas remotas no ano de 2021. No entanto, faltava - do ponto de vista do professor - um panorama geral para acompanhamento de quais exercícios estavam sendo realizados pelos alunos e quais eram as possíveis dificuldades dos alunos na resolução destes exercícios. Assim, decidiu-se em 2022, já no contexto de aulas presenciais, pela integração do NADIA ao Moodle.

Para a integração do NADIA ao Moodle utilizamos o *plugin Virtual Programming Lab* (VPL) Rodríguez-del-Pino et al., 2012, com o qual é possível incorporar o código fonte do NADIA ao Moodle e executá-lo sempre que o aluno desejar avaliar a corretude da prova escrita no editor de texto. Disponibilizamos um vídeo no Youtube[10]com o passo a passo para efetivar essa integração. A Figura 22 ilustra a visão do ambiente de programação para o aluno. No lado esquerdo da figura está disponível o editor no qual o aluno deve escrever a prova. Dentre os botões disponíveis na parte superior do editor, há o botão de executar o código. Quando o aluno clica neste botão, o código fonte do NADIA é executado, recebendo como entrada o texto que foi escrito no editor e emitindo como saída o resultado da avaliação da prova no lado direito da tela.

Após a integração ser concluída, os professores elaboraram um *banco de questões* de Dedução Natural no Moodle. Os exercícios no banco podem ser categorizados segundo conteúdos, nível de dificuldade, dentre outros critérios. Assim, a partir do banco de questões elaborado, o professor pode selecionar exercícios para compor listas de exercícios e disponibilizá-las aos alunos. Em cada exercício, o aluno pode ler o enunciado, escrever a solução e *avaliá-la automaticamente* no próprio AVA, sem a necessidade de acessar um outro ambiente.

Com a integração efetivada, foi possível que os professores pudessem, além de propor as listas de exercícios, acompanhar o desempenho dos alunos na resolução dos exercícios propostos. Com a correção automática dos exercícios, são gerados relatórios disponibilizados pelo Moodle nos quais é possível obter desde informações gerais - como quais alunos estão resolvendo os

[9]Disponível em: https://moodle2.quixada.ufc.br/

[10]Vídeo de integração do NADIA ao Moodle disponível em https://youtu.be/-xz_lCXGkD8

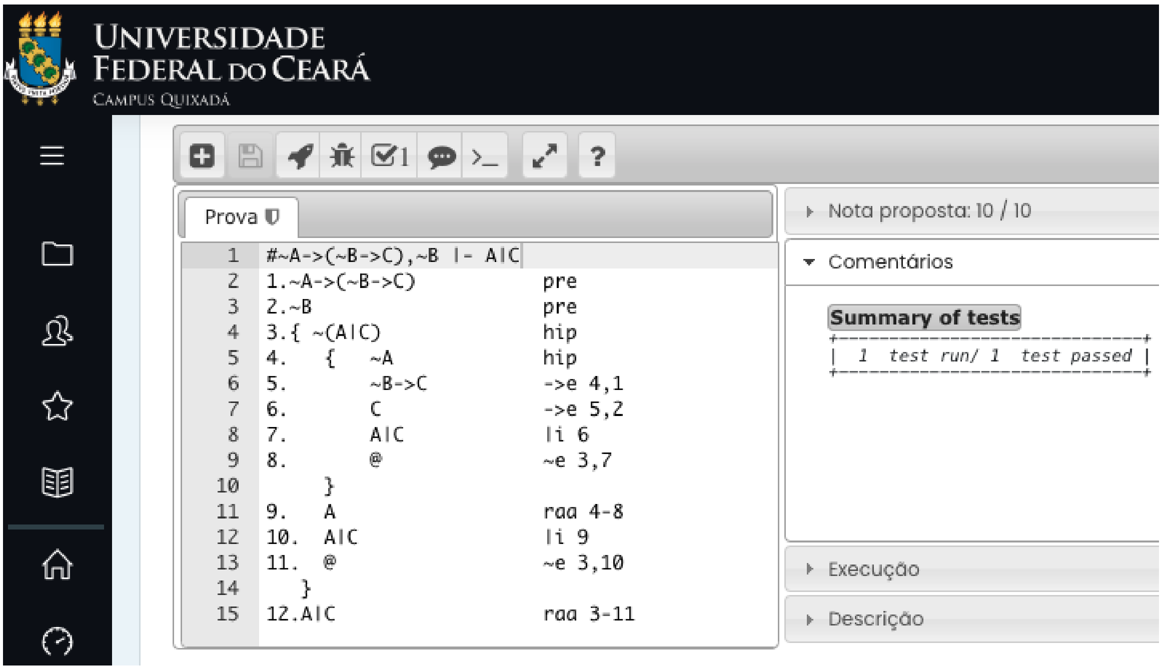


Figura 22: Exemplo de exercício resolvido no Moodle após a integração com o NADIA.

exercícios a medida que o conteúdo está sendo ministrado; quantas tentativas de submissões efetuam até conseguir uma resposta correta e quais os principais erros cometidos; quais alunos estão tentando solucionar os exercícios e não estão conseguindo; quais alunos nem ao menos tentam resolver os exercícios - até informações mais específicas como dias e horários que os alunos se dedicam a solução dos exercícios. Assim, é possível identificar o grupo de alunos que precisa de uma atenção individualizada do professor. A ideia é incentivar que o conteúdo seja estudado e exercitado ao longo do período em que está sendo ministrado, desencorajando que os alunos deixem para estudar o conteúdo apenas nas vésperas das avaliações.

Além das listas de exercícios, as avaliações parciais no ano de 2022 foram também realizadas em laboratório no ambiente do Moodle. Cada questão da avaliação era sorteada aleatoriamente a partir do banco de questões, possibilitando a geração de diferentes versões de provas para o grupo de alunos. No sorteio das questões para a avaliação, utilizamos para cada questão, exercícios com nível de dificuldade similar.

Com o sucesso desta experiência, algumas outras ferramentas de correção automática foram desenvolvidas e integradas ao Moodle para outros conteúdos da disciplina de Lógica para Computação, como é o caso da ferramenta Anita D. R. Vasconcelos, 2023 para correção automática de provas utilizando Tableaux Analíticos[11].

[11]Disponível em https://sistemas.quixada.ufc.br/anita/

6 Resultados da Avaliação da Experiência do Estudante e Análise

O uso do NADIA como ferramenta de ensino e aprendizagem do conteúdo de Dedução Natural iniciou no ano de 2021. Neste ano, por conta do isolamento social devido à pandemia de COVID-19, o contexto era de ensino remoto e a ferramenta estava implementada apenas para a escrita e avaliação de provas de Dedução Natural em Lógica Proposicional. No ano de 2022, o contexto passou ser o de aulas presenciais e a ferramenta passou por algumas melhorias, tais como: foi ampliada para permitir a escrita e avaliação de provas de Dedução Natural em Lógica de Predicados; o módulo de identificação e correção de erros foi aprimorado e; o NADIA foi integrado ao Moodle.

O formulário de avaliação foi disponibilizado para que os estudantes respondessem a perguntas divididas em dois blocos: (i) perguntas sobre a utilização do sistema como ferramenta de apoio ao estudo de Dedução Natural e (ii) perguntas sobre a usabilidade do sistema. As perguntas do primeiro bloco foram: "*Você utilizou o NADIA como ferramenta de estudo de Dedução Natural em Lógica Proposicional?*"; "*Caso tenha utilizado o NADIA, em quais momentos a ferramenta serviu como suporte às suas atividades?*"; "*Com que frequência você utilizou o NADIA no período em que o conteúdo de Dedução Natural em Lógica Proposicional estava sendo ministrado?*"; "*Você considera que o NADIA ajudou a exercitar o conteúdo de Dedução Natural?*". Já as perguntas sobre o segundo bloco foram em relação à usabilidade do sistema, a saber:

- **Q1**: facilidade de uso do sistema;
- **Q2**: compreensão das mensagens;
- **Q3**: organização da ferramenta e;
- **Q4**: identificação e correção de erros:

Além destas perguntas, o formulário apresentava um campo aberto para sugestões de melhorias.

6.1 Resultado da avaliação com turmas remotas no ano de 2021

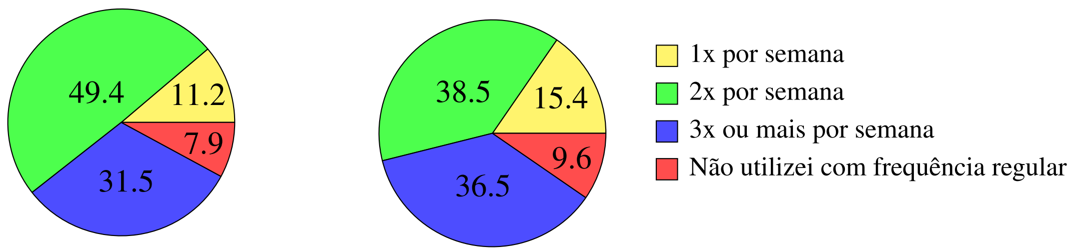
A avaliação da experiência de uso do sistema foi realizada com alunos das turmas de Lógica para Computação dos semestres 2021.1 e 2021.2 do Campus da UFC em Quixadá. Neste período o NADIA foi usado como ferramenta de ensino e aprendizagem do conteúdo de *Dedução Natural em Lógica Proposicional*.

Em 2021.1, havia 3 turmas da disciplina, contabilizando 122 alunos matriculados e obtivemos 90 respostas ao formulário de avaliação[12]. Neste semestre 98.9% dos respondentes utilizaram o NADIA como ferramenta de estudo de Dedução Natural em Lógica Proposicional. A Figura 23(a) mostra que a maioria dos estudantes mencionaram utilizar a ferramenta duas ou mais vezes por semana durante o período que o conteúdo estava sendo ministrado. Neste semestre também 98.9% dos alunos responderam que a ferramenta ajudou a exercitar o conteúdo. Em relação à forma como os alunos utilizaram a ferramenta, 91.0% responderam que utilizaram para resolver

[12] Disponível em https://forms.gle/gPDrWmP5WpRouF3U7

exercícios das *atividades assíncronas* propostas; 75.3% responderam que utilizaram para resolver exercícios das *atividades síncronas* e; 46.1% relataram que utilizaram para resolver outros exercícios além dos propostos pelos professores.

Em 2021.2, havia 2 turmas, contabilizando 101 alunos matriculados e obtivemos 52 respostas ao formulário de avaliação[13]. Neste semestre 98.1% dos respondentes utilizaram o NADIA como ferramenta de estudo do conteúdo de Dedução Natural em Lógica Proposicional. A Figura 23(b) mostra que, também no semestre 2021.2, a maioria dos estudantes mencionaram utilizar a ferramenta duas ou mais vezes por semana durante o período que o conteúdo estava sendo ministrado. Neste semestre ainda 96.2% dos alunos responderam que a ferramenta ajudou a exercitar o conteúdo. Em relação à forma como os alunos utilizaram a ferramenta, 96.1% responderam que utilizaram para resolver exercícios das *atividades assíncronas*; 80.4% responderam que utilizaram para resolver exercícios das *atividades síncronas* e; 47.1% relataram que utilizaram para resolver outros exercícios além dos propostos pelos professores.



(a) 2021.1 (b) 2021.2

Figura 23: Frequência semanal de utilização do sistema durante o período que o conteúdo de dedução Natural estava sendo ministrado.

A Figura 24 ilustra o gráfico com as respostas sobre usabilidade do sistema em 2021.1. Neste semestre, 34.8% dos respondentes avaliou como “muito bom” a facilidade de uso do sistema; 56.2% avaliou como “bom”; 7.9%, como regular e 1.1%, como ruim. Em relação à compreensão das mensagens reportadas, 17% avaliou como “muito bom”; 35.2% avaliou como bom; 45.5%, como “regular”; e 2.3%, como “ruim”. Em relação à organização da ferramenta: 40.4% avaliou como “muito bom”; 44.9% avaliou como “bom”; 13.5%, como regular e 1,1% avaliou como ruim. Já quanto à identificação e correção de erros, 59,2% dos alunos classificaram a ferramenta como "muito boa"ou "boa".

Dentre as sugestões de melhorias apontadas, em campo aberto, pelos alunos deste semestre podemos citar: construção de um manual interativo; mensagens de erro mais precisas e; inserção automática das numerações das linhas.

A Figura 25 ilustra o gráfico com as respostas dos alunos do semestre 2021.2 às questões de usabilidade do sistema. Neste semestre, 34.6% dos respondentes avaliou como “muito bom” a facilidade de uso do sistema; 46.2% avaliou como “bom”; 19.2%, como regular. Em relação à compreensão das mensagens reportadas, 11.5% avaliou como “muito bom”; 36.5% avaliou como bom; 34.6%, como “regular”; e 17.3%, como “ruim”. Em relação à organização da ferramenta: 44.2% avaliou como “muito bom”; 42.3% avaliou como “bom” e 13.5%, como regular. Já em

[13] Disponível em https://forms.gle/G4MGbRugNgWPp17W9

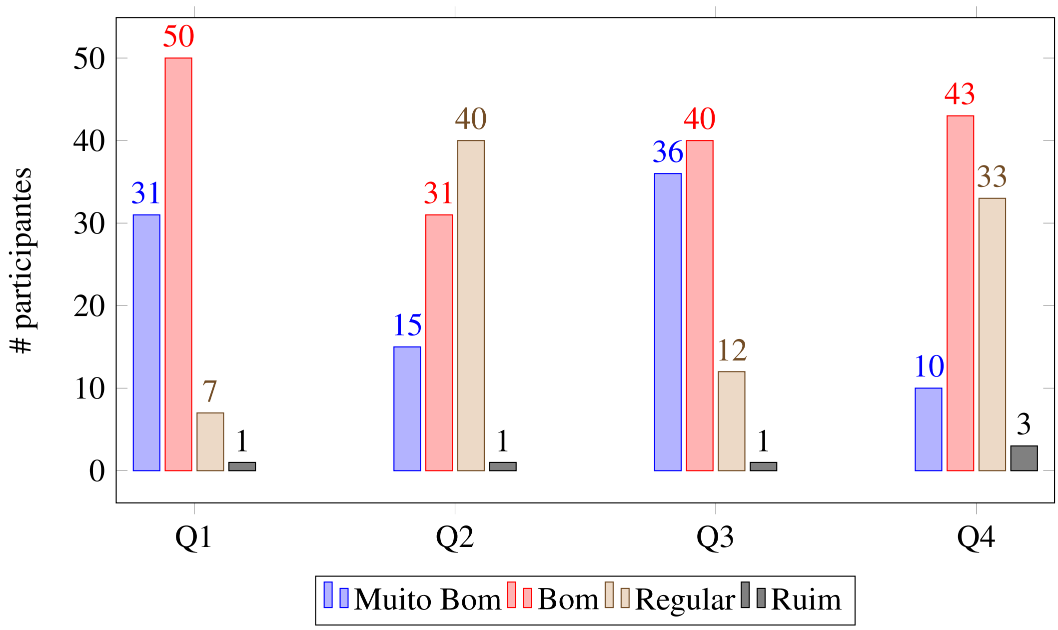


Figura 24: Respostas às questões de usabilidade no período 2021.1.

relação à identificação e correção de erros, 48,1% classificaram a ferramenta como "muito boa"ou "boa". Este foi um indicativo de que esta funcionalidade precisava ser melhorada para o próximo ano letivo.

Dentre as sugestões de melhorias apontadas, em campo aberto, pelos alunos deste semestre podemos citar: mostrar na página a correspondência entre os símbolos da lógica e os do NADIA; melhorar as mensagens de erro e; colorir as chaves que representam as caixas.

6.2 Resultado da avaliação com turmas presenciais no ano de 2022

A avaliação da experiência de uso do sistema foi realizada com alunos das turmas de Lógica para Computação dos semestres 2022.1 e 2022.2 do Campus da UFC em Quixadá. Neste período o NADIA foi usado como ferramenta de ensino e aprendizagem do conteúdo de *Dedução Natural em Lógica Proposicional e Lógica de Predicados*. Além disso, a ferramenta também foi integrada ao Moodle neste período, conforme relatado na seção anterior.

Em 2022.1, o NADIA foi avaliado em duas turmas de Lógica para Computação, sendo utilizado por um total de 74 alunos. Destes, 36 responderam o questionário de avaliação da ferramenta. Neste semestre 100% dos respondentes indicaram utilizar o NADIA para resolução de exercícios propostos pelo professor e 58% informaram usar a ferramenta para resolução de outros exercícios, além dos propostos pelo professor.

A Figura 26 ilustra o gráfico com as respostas às perguntas sobre usabilidade do sistema. Neste semestre, 75.0% dos respondentes avaliou como “muito bom” a facilidade de uso do sistema; 25.0% avaliou como “bom”; Em relação à compreensão das mensagens reportadas, 41.66% avaliou como “muito bom”; 36.11% avaliou como bom; 22.22%, como “regular”. Em relação à organização da ferramenta: 83.33% avaliou como “muito bom” e 16.66% avaliou como “bom”. Em relação à identificação e correção de erros: nenhum aluno reportou a resposta “muito ruim” e

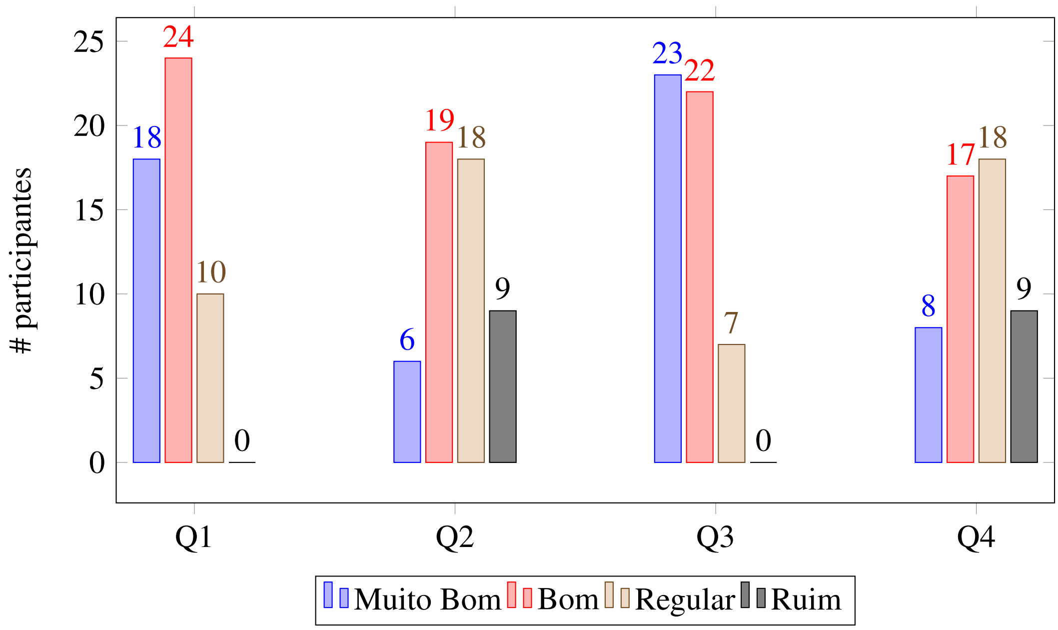


Figura 25: Respostas às questões de usabilidade no período 2021.2

69,3% classificaram como “muito bom” ou “bom”.

Dentre as melhorias apontadas, em campo aberto, pelos alunos deste semestre podemos citar: mostrar na página a correspondência entre os símbolos da lógica e os do NADIA; implementar a contagem automática das linhas e; implementar sistema de cores.

Em 2022.2, o NADIA foi avaliado em uma turma de Lógica para Computação com 34 alunos. Destes, 8 alunos responderam ao formulário de avaliação[14]. Todos os alunos informaram utilizar o NADIA para resolução dos exercícios propostos pelo professor e 37,5% informaram utilizar a ferramenta para resolução de exercícios, além dos propostos pelo professor. Metade dos respondentes disseram utilizar o NADIA duas vezes por semana para resolução de exercícios e metade disseram utilizar a ferramenta três ou mais vezes por semana.

A Figura 27 ilustra o gráfico com as respostas dos usuários quanto à usabilidade do sistema em 2022.2. Neste semestre, 100.0% dos respondentes avaliou como “muito bom” a facilidade de uso do sistema. Em relação à compreensão das mensagens reportadas, 62.5% avaliou como “bom”; 25.0% avaliou como bom; 12.5%, como “regular”. Em relação à organização da ferramenta: 87.5% avaliou como “muito bom” e 12.5% avaliou como “bom”. Em relação à identificação e correção de erros, 75% dos alunos reportaram como "muito bom"ou "bom".

Dentre as melhorias apontadas, em campo aberto, pelos alunos deste semestre podemos citar: implementação de funcionalidade de desfazer ações (CTRL + Z), inclusão de numeração automática das linhas e implementação de esquema de cores para ajudar na visualização das fórmulas.

[14]Disponível em https://forms.gle/vrFWV39afxM8Hgtk9

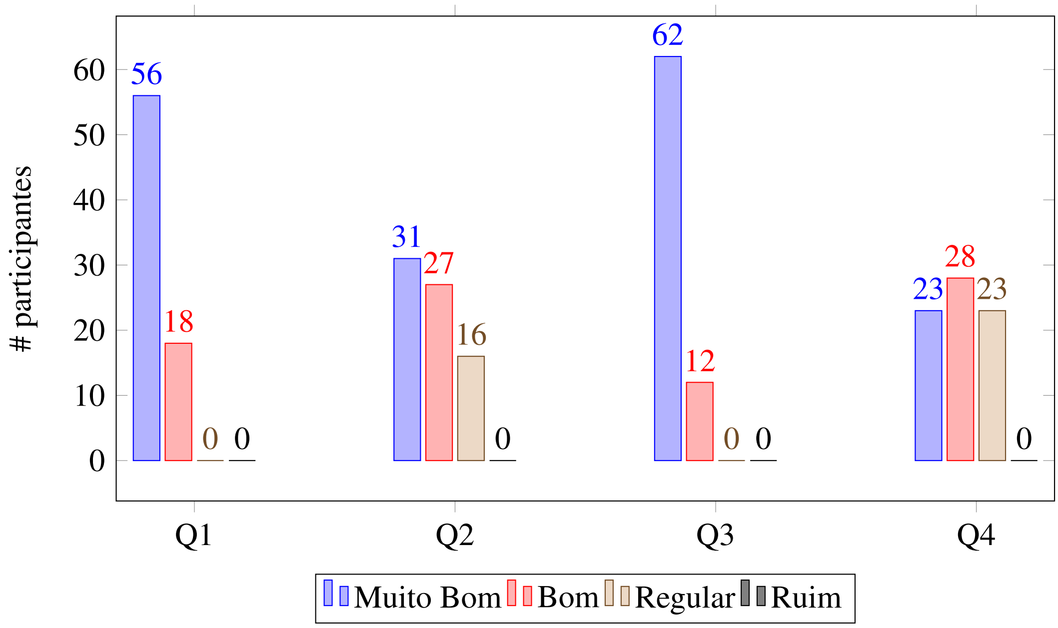


Figura 26: Respostas às questões de usabilidade no período 2022.1

6.3 Análise da Avaliação da Experiência do Usuário

Os resultados mostraram a eficácia do NADIA como ferramenta para ensino e aprendizagem de Dedução Natural nos contextos de aulas remotas e aulas presenciais e ao longo dos anos de 2021 e 2022.

No ano de 2021, em um contexto de aulas remotas, a taxa de adesão dos estudantes à ferramenta foi notável. A frequência de uso também foi destacada, uma vez que a maioria dos alunos relatou utilizar a ferramenta duas ou mais vezes por semana. Os resultados apontaram que o NADIA desempenhou um papel significativo no exercício e prática do conteúdo de Dedução Natural. No quesito usabilidade, a facilidade de uso do sistema recebeu avaliações positivas, com a maioria dos alunos classificando-a como "muito boa" ou "boa". No entanto, as avaliações sobre a compreensão das mensagens fornecidas pela ferramenta e a organização geral da mesma revelaram uma distribuição mais variada entre as categorias, indicando áreas para possíveis melhorias.

No ano de 2022, com o retorno às aulas presenciais, o NADIA passou a abranger também o conteúdo de Lógica de Predicados. Nesse contexto, a taxa de utilização da ferramenta permaneceu alta. As avaliações de usabilidade neste ano foram mais positivas, com a maioria dos alunos considerando a facilidade de uso, a compreensão das mensagens e a organização da ferramenta como "muito boas". Isso sugere que as melhorias implementadas entre os anos de 2021 e 2022 tiveram um impacto positivo na experiência dos alunos. Quanto à identificação e correção de erros, observou-se uma melhoria notável nos resultados em relação ao ano anterior, indicando que as aprimoramentos implementados foram bem-sucedidos em abordar as preocupações dos alunos quanto a esse aspecto específico da ferramenta.

A análise dos resultados da avaliação do uso do NADIA como ferramenta de ensino e aprendizagem sugere que a ferramenta desempenhou um papel eficaz no apoio à prática de Dedução Natural. A adesão significativa dos alunos, juntamente com a alta frequência de uso, indica que a

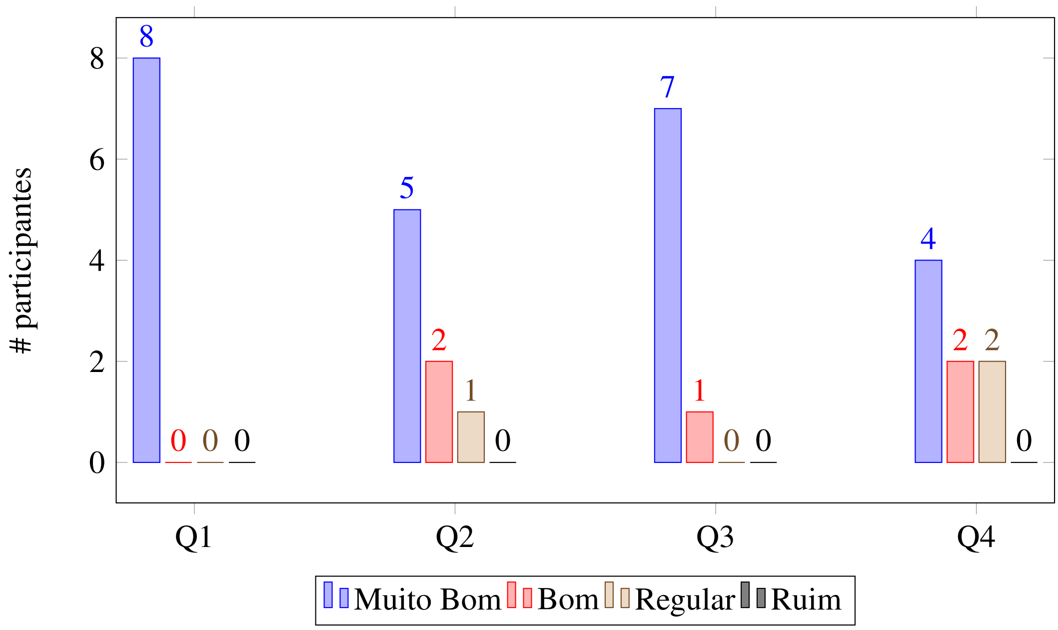


Figura 27: Respostas às questões de usabilidade no período 2022.2

ferramenta se tornou uma parte valiosa das atividades de aprendizagem dos estudantes.

As avaliações positivas sobre usabilidade, compreensão das mensagens e organização da ferramenta são indicativos de que as melhorias implementadas tiveram um impacto positivo na experiência dos alunos. No entanto, ainda há espaço para aprimoramentos contínuos, especialmente com base nas sugestões fornecidas pelos próprios alunos. Melhorias como a inclusão de uma correspondência clara entre os símbolos da lógica e os do NADIA, numeração automática das linhas e um sistema de cores podem enriquecer ainda mais a experiência do usuário.

7 Conclusão e Trabalhos Futuros

Este trabalho apresentou um assistente de provas para auxiliar no ensino de Dedução Natural no estilo de Fitch (caixas). A ferramenta foi desenvolvida em Python, integrada ao Moodle e testada em cinco turmas da disciplina de Lógica para Computação, nos anos de 2021 e 2022.

NADIA permite que o estudante escreva as provas de Dedução Natural em Lógica Proposicional no estilo Fitch de uma forma bem semelhante ao que se aprende na literatura e ao que usualmente se faz no papel. A ferramenta foi avaliada por estudantes neste período e a grande maioria mencionou: utilizar a ferramenta duas ou mais vezes por semana durante o período que o conteúdo de Dedução Natural estava sendo ministrado; que a ferramenta ajudou a exercitar o conteúdo de Dedução Natural; utilizar a ferramenta para resolver exercícios das atividades propostas. Em relação à usabilidade da ferramenta, o principal problema apontado pelos alunos no ano de 2021 foi em relação à compreensão das mensagens de erros reportadas. No ano de 2022 implementamos melhorias no módulo de apresentação das mensagens de erros e esta funcionalidade passou a ser melhor avaliada pelos alunos.

Como trabalhos futuros pretendemos incorporar o NADIA em outras ferramentas que podem ser utilizadas em sala de aula, como é o caso do Google Colaboratory. Também pretendemos ampliar o repertório de ferramentas para outros sistemas dedutivos, como é o caso da implementação e avaliação da ferramenta Anita para o sistema dedutivo de Tableaux Analíticos.

Artigo Premiado Estendido

Esta publicação é uma versão estendida de artigo premiado no XXX Workshop sobre Educação em Computação (WEI 2022), intitulado "NADIA - Natural DeductIon proof Assistan", DOI: 10.5753/wei.2022.222875.

Referências

Bornat, R., & Sufrin, B. (1996). Jape's quiet interface. *User Interfaces for Theorem Provers (UITP'96), Technical Report*, 25–34. [GS Search].

Chang, C.-L., & Lee, R. C.-T. (1976). *Symbolic logic and mechanical theorem proving*. Academic press. [GS Search].

da Costa, N. (2008). *Introdução aos Fundamentos da Matemática* (4a). Hucitec. [GS Search].

da Silva, F., Finger, M., & de Melo, A. (2006). *Lógica para Computação*. Cengage Learning. https://books.google.com.br/books?id=w27uOgAACAAJ [GS Search].

de Souza, J. (2008). *Logica Para Ciencia da Computação*. Elsevier. https://books.google.com.br/books?id=Y8GEsUoRKiEC [GS Search].

Ferreirós, J. (2001). The road to modern logic—an interpretation. *Bulletin of Symbolic Logic*, *7*(4), 441–484. https://doi.org/10.2307/2687794 [GS Search].

Gasquet, O., Schwarzentruber, F., & Strecker, M. (2011). Panda: A Proof Assistant in Natural Deduction for All. A Gentzen Style Proof Assistant for Undergraduate Students. Em P. Blackburn, H. van Ditmarsch, M. Manzano & F. Soler-Toscano (Ed.), *Tools for Teaching Logic* (pp. 85–92). Springer Berlin Heidelberg. [GS Search].

Gentzen, G. (1969). *The collected papers*. North-Holland Publishing Company. https://doi.org/10.2307/2272429 [GS Search].

Hammer, E. M. (1998). Semantics for existential graphs. *Journal of Philosophical Logic*, *27*, 489–503. https://doi.org/10.1023/A:1017908108789 [GS Search].

Huth, M., & Ryan, M. (2004). *Logic in Computer Science: Modelling and Reasoning about Systems (2nd Ed.)* Cambridge University Press. https://doi.org/10.1017/CBO9780511810275 [GS Search].

Leach-Krouse, G. (2018). Carnap: An Open Framework for Formal Reasoning in the Browser. Em P. Quaresma & W. Neuper (Ed.), Proceedings 6th International Workshop on *Theorem proving components for Educational software,* Gothenburg, Sweden, 6 Aug 2017 (pp. 70–88, Vol. 267). Open Publishing Association. https://doi.org/10.4204/EPTCS.267.5 [GS Search].

Maxim, H., Kaliszyk, C., van Raamsdonk, F., & Wiedijk, F. (2010). Teaching logic using a state-of-art proof assistant. *Acta Didactica Napocensia*, *3*. https://files.eric.ed.gov/fulltext/EJ1056118.pdf [GS Search].

Pelletier, F. J. (1999). A brief history of natural deduction. *History and Philosophy of Logic*, *20*(1), 1–31. https://doi.org/10.1080/014453499298165 [GS Search].

Pelletier, F. J. (2000). A history of natural deduction and elementary logic textbooks. *Logical consequence: Rival approaches*, *1*, 105–138. [GS Search].

Rodríguez-del-Pino, J. C., Rubio Royo, E., & Hernández Figueroa, Z. (2012). A Virtual Programming Lab for Moodle with automatic assessment and anti-plagiarism features. [GS Search].

van Dalen, D. (2013). *Logic and Structure (5th Ed.)* Springer London. https://doi.org/10.1007/978-1-4471-4558-5 [GS Search].

Vasconcelos, D., Paula, R., & Menezes, M. (2022). NADIA - Natural DeductIon proof Assistant. *Anais do XXX Workshop sobre Educação em Computação*, 427–438. https://doi.org/10.5753/wei.2022.222875 [GS Search].

Vasconcelos, D. R. (2023). ANITA: Analytic Tableau Proof Assistant. Em P. Quaresma, J. Marcos & W. Neuper (Ed.), Proceedings 11th International Workshop on *Theorem Proving Components for Educational Software,* Haifa, Israel, 11 August 2022 (pp. 38–53, Vol. 375). Open Publishing Association. https://doi.org/10.4204/EPTCS.375.4 [GS Search].

Villadsen, J., From, A. H., & Schlichtkrull, A. (2018). Natural Deduction and the Isabelle Proof Assistant. Em P. Quaresma & W. Neuper (Ed.), Proceedings 6th International Workshop on *Theorem proving components for Educational software,* Gothenburg, Sweden, 6 Aug 2017 (pp. 140–155, Vol. 267). Open Publishing Association. https://doi.org/10.4204/EPTCS.267.9 [GS Search].