

UNIVERSIDADE FEDERAL DO PARANÁ

FABIO HENRIQUE GIL

**STI - LÓGICA LIVRE: UM CONTRIBUTO À APRENDIZAGEM DE  
EQUIVALÊNCIA LÓGICA PROPOSICIONAL NO ÂMBITO DO  
ENSINO SUPERIOR**

PALOTINA

2018

FABIO HENRIQUE GIL

**STI - LÓGICA LIVRE: UM CONTRIBUTO À APRENDIZAGEM DE  
EQUIVALÊNCIA LÓGICA PROPOSICIONAL NO ÂMBITO DO  
ENSINO SUPERIOR**

Trabalho de Conclusão de Curso apresentado ao Curso de Licenciatura em Computação da Universidade Federal do Paraná como requisito para a obtenção do título Licenciado em Computação

Orientador: Prof. Me. Marcos Antonio Schreiner  
Coorientadora: Profa. Dra. Eliana Santana Lisbôa

PALOTINA  
2018

# TERMO DE APROVAÇÃO

FABIO HENRIQUE GIL

## STI - LÓGICA LIVRE: UM CONTRIBUTO À APRENDIZAGEM DE EQUIVALÊNCIA LÓGICA PROPOSICIONAL NO ÂMBITO DO ENSINO SUPERIOR

Trabalho de Conclusão de Curso apresentado ao Curso de Licenciatura em Computação da Universidade Federal do Paraná como requisito à obtenção do título de Licenciado em Computação, e aprovado pela seguinte banca examinadora:

---

Orientador: Prof. Me. Marcos Antonio Schreiner  
Setor Palotina, UFPR

---

Coorientadora: Profa. Dra. Eliana Santana Lisbôa  
Setor Palotina, UFPR

---

Profa. Me. Raquel Angela Speck  
Setor Palotina, UFPR

---

Prof. Me. Helio Henrique Lopes Costa Monte-Alto  
Setor Palotina, UFPR

---

Prof. Me. Wander Mateus Branco Meier  
Setor Palotina, UFPR

Palotina, 29 de junho de 2018

Este trabalho é dedicado aos que acreditam na própria capacidade, que o esforço e dedicação é o que nos permitem vencer desafios e almejar ir ainda mais longe.

## **AGRADECIMENTOS**

A Deus, por ter concedido saúde para vencer os desafios.

A Universidade Federal do Paraná (UFPR) e seu corpo docente, pela oportunidade de realizar um curso com ótima qualidade de ensino, especialmente, ao meu orientador Marcos Antonio Schreiner e minha coorientadora Eliana Santana Lisbôa pelo suporte que me deram durante a elaboração deste trabalho.

Agradeço aos meus pais, Devair Gil e Lucinda Gil, por terem acreditado e incentivado este ideal e minha irmã Angela Gil, por apoiar e permanecer ao meu lado nos momentos mais difíceis.

A todos os meus irmãos e amigos que participaram direta ou indiretamente dessa formação, muito obrigado.

*“Se definir é se limitar.  
Você é um eterno parênteses em aberto,  
enquanto sua eternidade durar”.*  
*(Machado de Assis)*

## RESUMO

O presente trabalho aborda o desenvolvimento e a avaliação de um Sistema Tutor Inteligente (STI) a ser utilizado no ensino de Equivalência Lógica Proposicional, usufruindo de tecnologias na área da Inteligência Artificial (IA), aplicadas à Educação. A usabilidade do STI desenvolvido é baseada na solução de desafios, provendo *feedbacks* personalizados, paralelos ao progresso do aluno. Porém, a principal característica do sistema é levar em conta a individualidade do aluno no que se refere ao ritmo e a forma com que ele aprende, de modo que o *software* poderá trabalhar na Zona de Desenvolvimento Proximal (ZDP) do aprendiz, por meio de um grafo de conteúdos. A metodologia utilizada no trabalho é denominada *Development Research*, na qual buscamos a solução de um problema vivenciado em cursos de Computação. O STI - Lógica Livre foi submetido a uma série de testes e avaliações, com o intuito de validá-lo enquanto ferramenta educacional, tais avaliações consistem em (i) avaliação por especialistas nas áreas de Computação, Educação e Matemática; (ii) utilização do STI por um grupo de alunos de graduação, com pré e pós teste (sendo pré teste antes da utilização e pós teste depois dela) para avaliar o desenvolvimento dos alunos; (iii) e uma avaliação do sistema pelos alunos utilizadores. Como resultado obtivemos uma melhora considerável no desempenho dos alunos que fizeram uso do *software*, além de considerações pertinentes nos questionários de avaliação. Ao fim da pesquisa, é possível afirmar que o STI - Lógica Livre contribuiu na aprendizagem dos alunos que o utilizaram.

Palavras-chave: STI; Ensino de Lógica; Inteligência Artificial.

## ABSTRACT

*The present work deals with the development and evaluation of an Intelligent Tutoring System (ITS) to be used in the teaching of Propositional Logical Equivalence, using technologies in the field of Artificial Intelligence (AI) applied to education. The usability of the STI developed is based on the solution of challenges, providing personalized feedbacks, parallel to the progress of the student. However, the main characteristic of the system is to take into account the individuality of the student with regard to the rhythm and way in which he/she learns, so that software can work in the learner's Zone of Proximal Development (ZPD). To develop such a tool, we use the methodology Development Research, which seeks to solve a problem experienced in courses of Computing. The STI - Lógica Livre has undergone a series of tests and evaluations, with the purpose of validating its effectiveness as an educational tool, such as evaluation by specialists in the fields of Computing, Education and Mathematics; use of STI by a group of undergraduate students, with pre and post test (being pre test before use and post test after) to evaluate students progress; and finally an evaluation of the system by users. As a result we have achieved a considerable improvement in the performance of the students who made use of the software, as well as pertinent considerations in the evaluation questionnaires. At the end of the research, it is possible to affirm that the STI - Lógica Livre contributed to the learning of the students who used it.*

Palavras-chave: STI; Logic Learning; Artificial Intelligence.



## LISTA DE FIGURAS

FIGURA 1 – ARQUITETURA TRIPARTIDA PARA UM STI . . . . .	29
FIGURA 2 – DIAGRAMA DE ENTIDADE E RELACIONAMENTO (DER) DO STI - LÓGICA LIVRE . . . . .	39
FIGURA 3 – GRAFO DE REPRESENTAÇÃO DO CONHECIMENTO . . . . .	42
FIGURA 4 – A PONTUAÇÃO E O ÍNDICE DE ACERTOS EM DETERMINADA REGRA	43
FIGURA 5 – QUANTIDADE DE DESAFIOS RESOLVIDOS PARA DETERMINADA REGRA . . . . .	43
FIGURA 6 – PESO RETORNADO PELO SISTEMA <i>FUZZY</i> . . . . .	45
FIGURA 7 – EQUIVALÊNCIA LÓGICA PROPOSICIONAL . . . . .	46
FIGURA 8 – PERSONAGEM LOGEEK . . . . .	48
FIGURA 9 – DIFERENTES FORMAS DE APRESENTAÇÃO DO LOGEEK . . . . .	49
FIGURA 10 – EXPLICAÇÃO INICIAL SOBRE PROPOSIÇÕES . . . . .	50
FIGURA 11 – EXPLICAÇÃO INICIAL SOBRE CONECTIVOS . . . . .	50
FIGURA 12 – EXPLICAÇÃO INICIAL SOBRE EXPRESSÕES . . . . .	51
FIGURA 13 – EXPLICAÇÃO INICIAL SOBRE EQUIVALÊNCIA . . . . .	51
FIGURA 14 – DESAFIO DE SELECIONAR REGRA . . . . .	52
FIGURA 15 – DESAFIO DE SELECIONAR EQUIVALÊNCIAS . . . . .	52
FIGURA 16 – DESAFIO DE DIGITAR EQUIVALÊNCIA . . . . .	53
FIGURA 17 – DICAS SOBRE AS REGRAS . . . . .	54
FIGURA 18 – DICA COM REGISTRO SEMIÓTICO . . . . .	54
FIGURA 19 – CALCULADORA LÓGICA . . . . .	55
FIGURA 20 – TOKENS - LIBERAR USO DA CALCULADORA . . . . .	56
FIGURA 21 – RELATÓRIO DE DESEMPENHO DO ALUNO . . . . .	56
FIGURA 22 – NÍVEL DE DIFICULDADE DA DISCIPLINA DE LÓGICA . . . . .	58
FIGURA 23 – RESULTADOS DO PRÉ TESTE DO GRUPO DE APOIO . . . . .	64
FIGURA 24 – RESULTADOS DO PÓS TESTE DO GRUPO DE APOIO . . . . .	64
FIGURA 25 – RESULTADOS DO PRÉ TESTE DO GRUPO DE ESTUDO . . . . .	65
FIGURA 26 – RESULTADOS DO PÓS TESTE DO GRUPO DE ESTUDO . . . . .	66
FIGURA 27 – RELAÇÃO ENTRE ACERTOS DO GRUPO DE APOIO . . . . .	66
FIGURA 28 – RELAÇÃO ENTRE ACERTOS DO GRUPO DE ESTUDO . . . . .	66
FIGURA 29 – RELAÇÃO ENTRE DESEMPENHO NOS TESTES E QUANTIDADE DE UTILIZAÇÃO DO STI - LÓGICA LIVRE . . . . .	68
FIGURA 30 – SEXTA QUESTÃO DO FORMULÁRIO SOBRE A EXPERIÊNCIA DE USO	69
FIGURA 31 – SÉTIMA QUESTÃO DO FORMULÁRIO SOBRE A EXPERIÊNCIA DE USO . . . . .	70
FIGURA 32 – NONA QUESTÃO DO FORMULÁRIO SOBRE A EXPERIÊNCIA DE USO	71

## LISTA DE TABELAS

TABELA 1 – REGRAS DE PRODUÇÃO PARA TOMADA DE DECISÃO . . . . .	44
TABELA 2 – RESULTADOS DO QUESTIONÁRIO DO ESPECIALISTA EM COMPUTAÇÃO . . . . .	60
TABELA 3 – RESULTADOS DO QUESTIONÁRIO DO ESPECIALISTA EM EDUCAÇÃO	61
TABELA 4 – RESULTADOS DO QUESTIONÁRIO DO ESPECIALISTA EM MATEMÁTICA . . . . .	62
TABELA 5 – DADOS DE UTILIZAÇÃO DO STI - LÓGICA LIVRE PELOS ALUNOS .	67

## LISTA DE SIGLAS

AG	Algoritmos Genéticos
ASSOC	Associação
BICOND	Bicondicional
CAI	Instrução de Assistência por Computador
COM	Comutação
COND	Condicional
CP	Contraposição
CSS	Folhas de Estilo em Cascata
DER	Diagrama Entidade e Relacionamento
DIST	Distributiva
DM	De Morgan
DN	Dupla Negação
EC	Engenharia do Conhecimento
EI	Exportação-Importação
FNC	Forma Normal Conjuntiva
FND	Forma Normal Disjuntiva
FOL	Linguagem de Primeira Ordem
IA	Inteligência Artificial
ICAI	Instrução de Assistência por Computadores Inteligentes
ID	Idempotência
LN	Linguagem Natural
MD	Mineração de Dados
OD	Orientação a Dados
OO	Orientação a Objetos

RC	Representação de Conhecimento
SBC	Sistemas Baseados em Conhecimento
SE	Sistema Especialista
STI	Sistema Tutor Inteligente
TFC	Teoria dos Conjuntos <i>Fuzzy</i>
TI	Tecnologia da Informação
UFPR	Universidade Federal do Paraná
ZDP	Zona de Desenvolvimento Proximal
ZDR	Zona de Desenvolvimento Real

## LISTA DE SÍMBOLOS

$\neg$	Negação
$\vee$	Disjunção (Ou)
$\wedge$	Conjunção (E)
$\rightarrow$	Implicação
$\leftrightarrow$	Bi implicação

## SUMÁRIO

<b>1</b>	–	<b>INTRODUÇÃO</b> . . . . .	<b>15</b>
<b>2</b>	–	<b>OBJETIVOS</b> . . . . .	<b>18</b>
2.1		OBJETIVO GERAL . . . . .	18
2.2		OBJETIVOS ESPECÍFICOS . . . . .	18
<b>3</b>	–	<b>FUNDAMENTAÇÃO TEÓRICA</b> . . . . .	<b>19</b>
3.1		TEORIAS DE APRENDIZAGEM . . . . .	19
3.2		O ENSINO DE LÓGICA . . . . .	20
3.3		CRITÉRIOS ERGONÔMICOS PARA O DESENVOLVIMENTO DE <i>SOFTWARE</i> . . . . .	21
3.4		PRINCÍPIOS GERAIS PARA ENGENHARIA DE <i>SOFTWARE</i> . . . . .	23
3.5		REPRESENTAÇÃO DO CONHECIMENTO E CONCEITOS INERENTES . . . . .	24
3.6		TECNOLOGIA DOS COMPILADORES E PROCESSOS DE ANÁLISE . . . . .	27
3.7		ARQUITETURAS DE SISTEMAS TUTORES INTELIGENTES (STI) . . . . .	28
3.8		TRABALHOS RELACIONADOS . . . . .	30
<b>4</b>	–	<b>PROCEDIMENTOS METODOLÓGICOS</b> . . . . .	<b>33</b>
4.1		METODOLOGIA . . . . .	33
4.2		REVISÃO DA LITERATURA . . . . .	33
4.3		VERIFICAÇÃO DOS PROBLEMAS NA APRENDIZAGEM DE LÓGICA E NECESSIDADE DE AUXÍLIO TECNOLÓGICO . . . . .	34
4.4		DESENVOLVIMENTO DO <i>SOFTWARE</i> . . . . .	34
4.4.1		Levantamento de Requisitos . . . . .	34
4.4.2		Ciclo de Vida do Projeto . . . . .	35
4.4.3		Tecnologias Utilizadas . . . . .	35
4.5		VALIDAÇÃO DO <i>SOFTWARE</i> POR ESPECIALISTAS DAS ÁREAS DE COMPUTAÇÃO, EDUCAÇÃO E MATEMÁTICA . . . . .	36
4.6		APLICAÇÃO DO PRÉ TESTE . . . . .	36
4.7		UTILIZAÇÃO DO <i>SOFTWARE</i> PELO GRUPO DE ESTUDOS . . . . .	36
4.8		APLICAÇÃO DO PÓS TESTE . . . . .	37
4.9		QUESTIONÁRIO SOBRE A EXPERIÊNCIA DE USO DO SISTEMA . . . . .	37
4.10		TÉCNICAS DE ANÁLISE DE DADOS . . . . .	37
<b>5</b>	–	<b>O SISTEMA STI - LÓGICA LIVRE</b> . . . . .	<b>39</b>
5.1		BASE DE FATOS E BASE DE CONHECIMENTO . . . . .	39
5.2		ARQUITETURA . . . . .	41
5.2.1		Módulo do Aluno . . . . .	41
5.2.2		Módulo de Domínio . . . . .	45
5.2.3		Módulo Tutor . . . . .	47
5.2.4		Módulo de Interface . . . . .	48

5.3	FUNCIONALIDADES . . . . .	48
6	– ANÁLISE E DISCUSSÕES . . . . .	58
7	– CONCLUSÕES E TRABALHOS FUTUROS . . . . .	72
	REFERÊNCIAS . . . . .	74
	APÊNDICE A – PRÉ TESTE . . . . .	78
	APÊNDICE B – PÓS TESTE . . . . .	79
	APÊNDICE C – OBSERVAÇÕES QUALITATIVAS DO ESPECI- ALISTA EM EDUCAÇÃO . . . . .	80
	APÊNDICE D – OBSERVAÇÕES QUALITATIVAS DO ESPECI- ALISTA EM MATEMÁTICA . . . . .	81

## 1 INTRODUÇÃO

No contexto atual, permeado por tecnologias digitais, surge o grande interesse de adolescentes e jovens pelas novidades tecnológicas. Um grande desafio é apresentado aos professores, pois precisam lidar com uma geração que necessita de maiores estímulos para aprender (VEEN; VRAKING, 2009).

Outro fator que desafia o professor em sala de aula é a individualidade dos alunos. Segundo Pimentel, França e Omar (2003), um grande obstáculo encontrado pelas instituições de ensino é saber lidar com competências e estilos de aprendizagem diferenciados dos alunos em uma mesma sala, pois a não observação a esses fatores pode causar sérios prejuízos à aprendizagem, gerando sucessivas reprovações e/ou colocando no mercado de trabalho profissionais sem uma formação consistente (com deficiência no domínio dos conteúdos obrigatórios).

No caso específico do nível superior, as diferenças que os alunos apresentam entre os conhecimentos prévios nos indicam ter um impacto mais forte, uma vez que, dependendo da disciplina, pode ser gerada para alguns, uma dificuldade maior em adquirirem novos conhecimentos. Como exemplo, citamos a disciplina “Lógica Matemática” que é ministrada nos cursos de Computação de nível superior, de modo que muitos alunos têm dificuldades em compreendê-la (GRIVOKOSTOPOULOU; PERIKOS; HATZILYGEROUDIS, 2013; LODDER; PASSIER; STUURMAN, 2008).

Esse é também um problema recorrente entre muitos alunos do Curso de Licenciatura em Computação da Universidade Federal do Paraná (UFPR) - Setor Palotina, os quais parecem sentir-se inseguros, desenvolvendo uma espécie de aversão a esta disciplina que, pensamos ser, talvez, pelo nível de abstração que há nos conteúdos (altamente teóricos) ou pelos métodos tradicionais de ensino. Um desses conteúdos é o de “Equivalência Lógica Proposicional”, o qual é fundamental para a compreensão do método de dedução natural e de resolução.

Dessa forma, percebemos assim, o quanto o conhecimento prévio do aluno influencia na sua aprendizagem e, ao mesmo tempo, desafia a tarefa do professor de ensinar múltiplos alunos com suas respectivas particularidades. A explicação para isso, reside no fato de que o conhecimento prévio de cada aluno é influenciado por múltiplos fatores, os quais vão desde suas experiências pessoais até às escolares e/ou acadêmicas e que, por esse motivo, gera uma divergência entre os alunos.

De acordo com Nwana (1990), a forma com que cada aluno aprende constitui uma temática de estudos de várias áreas, como por exemplo, a Ciência da Computação, a Educação e a Psicologia. Contudo, essa não é uma temática nova, pois os estudos sobre a aprendizagem individual (com Sistemas tutores Inteligentes) remontam desde o clássico Scholar<sup>1</sup> (CARBONELL, 1970). Tais estudos contribuíram para o desenvolvimento e implantação de inúmeras ferramentas educacionais, dentre elas, citamos os Sistemas Tutores Inteligentes (STI). Os STI

---

<sup>1</sup>Sistema Tutor Inteligente desenvolvido para o ensino de geografia, que utilizava modelagem de conhecimento e podia dialogar com o aluno.



possuem um conhecimento do domínio a ser ensinado, e de um modelo do aprendizado do aluno, buscando direcionar o aprendizado de acordo com as suas características individuais, além de estratégias pedagógicas que levam em conta estes conhecimentos prévios. De acordo com Elsom-Cook (1987), a vantagem preliminar de um STI é a possibilidade de fornecer um ensino um-para-um<sup>2</sup>. Este método de ensino é dificilmente alcançado em uma sala de aula, dado que um professor leciona para vários alunos ao mesmo tempo e pela diversidade de níveis de conhecimento entre esses alunos.

De acordo com SILVA (2006), a aplicação dos STI na Educação contempla desde ações simples até as mais sofisticadas, as quais devem ser pensadas tendo como pressuposto a adoção de estratégias pedagógicas planejadas, considerando as especificidades do conteúdo. Partindo desse pressuposto, “um STI pode realizar o levantamento do perfil do aluno (conhecimentos prévios, preferências, interesses, necessidades de formação etc), com base no qual possa negociar com ele uma estrutura de curso [...] personalizada” (SILVA, 2006, p.7).

Ao falar sobre levantamento do perfil do aluno é importante ressaltar que os conhecimentos prévios funcionam como pontes cognitivas para aquisição de novos conhecimentos. Ausubel (2003) pressupõe que o material de aprendizagem deva estar relacionado de forma não arbitrária e não literal, isso porque a estrutura cognitiva particular do aprendiz deve conter ideias ancoradas relevantes, com as quais se possa relacionar o novo material (AUSUBEL, 2003).

Ao perceber a dificuldade e a importância que existe nos conteúdos de lógica do ensino superior, mais especificamente no curso de Licenciatura em Computação da UFPR - Setor Palotina, nos propomos a desenvolver e utilizar um STI com propósito de verificar na prática, se ele constitui uma ferramenta eficaz para minimizar as dificuldades de aprendizagem dos alunos no conteúdo de Equivalência Lógica Proposicional.

Embora as pesquisas sobre uso de ferramentas computacionais na Educação não seja um assunto novo, assim como a criação de ferramentas para o ensino de Lógica não se mostra como novidade na comunidade científica (LODDER; PASSIER; STUURMAN, 2008; GIRAFFA, 1999; MOSTAFAVI; BARNES, 2017; GRIVOKOSTOPOULOU; PERIKOS; HATZILYGEROUDIS, 2013; VICCARI, 1990), nossa pesquisa se atentou aos conteúdos iniciais do ensino de Lógica, especificamente, Equivalência Lógica Proposicional, assuntos estes que têm sido pouco abordados por pesquisadores da área. Vemos este conteúdo como essencial para que se possa avançar, não só na disciplina em questão, como também em conteúdos posteriores do curso.

Nesse contexto, o escopo do trabalho é delimitado à pesquisa e desenvolvimento de um Sistema Tutor Inteligente, denominado STI - Lógica Livre. Esta ferramenta objetiva auxiliar alunos e professores no processo de ensino e aprendizagem do conteúdo de Equivalência Lógica Proposicional no âmbito do ensino superior.

Para desenvolver o *software* intitulado “STI - Lógica Livre”, tomamos como ponto de partida a individualidade dos alunos (no que tangem seus conhecimentos prévios de Equiva-

---

<sup>2</sup>Ensino personalizado, onde cada aluno tem seu professor, que ensina conforme suas necessidades.

lência Lógica Proposicional e a forma como cada um aprende). Para isso, criamos formas de representação de conhecimento para que o ensino seja aplicado em torno desta representação. Foi considerada também a complexidade e formalidade dos conteúdos, provendo *feedback* passo a passo para o aluno conforme ele evolui nas atividades, se atentando também à realidade tecnológica em que os jovens estão inseridos, bem como a facilidade de distribuição e a atualização do *software*.

Em nosso STI, utilizamos de conhecimentos pedagógicos e tecnologias computacionais. O sistema opera em qualquer plataforma, ou seja, pode ser acessado de qualquer lugar com acesso à internet, além de ser de fácil distribuição e atualização. Logo, o STI desenvolvido é um sistema *Web* com tela responsiva<sup>3</sup>. Todavia, a principal característica do STI - Lógica Livre, é o fato de considerar a individualidade dos usuários. Vygotsky (1978) diz que é preciso analisar o que o aluno já conhece, para que se possa disponibilizar o conteúdo ótimo para ele, de acordo com sua Zona de Desenvolvimento Proximal<sup>4</sup> (ZDP). Por esse motivo, o sistema possibilita guardar as informações de desempenho de cada aluno, que são consideradas na análise do conhecimento e da evolução individual dos mesmos.

Sendo assim, o presente estudo tem sua relevância, no sentido em que poderá auxiliar os alunos do primeiro período do curso de Licenciatura em Computação da UFPR - Setor Palotina. A disciplina de Introdução à Lógica constitui um conteúdo base para que o aluno obtenha êxito nas disciplinas posteriores, como por exemplo: Algoritmos, Arquitetura de Computadores, Redes de computadores e sistemas distribuídos, Orientação a Objetos, Banco de Dados e Inteligência Artificial. Além disso, os conhecimentos adquiridos nessas disciplinas são fundamentais para os futuros profissionais da Computação. Partindo desse pressuposto, consideramos que o STI desenvolvido constitui uma ferramenta que poderá funcionar como aporte tecnológico e pedagógico para aprendizagem de Lógica.

As próximas seções deste trabalho estão organizados da seguinte forma: na seção 2 apresentamos os objetivos do projeto (geral e específicos); na seção 3 a fundamentação teórica, na qual apresentamos as pesquisas realizadas na literatura sobre os assuntos pertinentes ao estudo; na seção 4, os procedimentos metodológicos utilizados no trabalho; na seção 5, apresentamos o sistema STI - Lógica livre, bem como suas características funcionais e não funcionais; na seção 6 realizamos as análises e discussões sobre os dados obtidos durante os experimentos; e a seção 7 consiste nas conclusões e apresentação dos trabalhos futuros.

---

<sup>3</sup>*Layout* que se ajusta conforme o tamanho e a resolução da tela.

<sup>4</sup>Conteúdo que o aluno é capaz de assimilar de imediato, desde que tenha um auxílio externo (VYGOTSKY, 1978)

## 2 OBJETIVOS

### 2.1 OBJETIVO GERAL

Desenvolver e aplicar um STI para auxiliar os alunos do ensino superior no processo de aprendizagem de Equivalência Lógica Proposicional, avaliando-o enquanto ferramenta educacional.

### 2.2 OBJETIVOS ESPECÍFICOS

- Fazer uma revisão da literatura para fundamentar o desenvolvimento do STI, bem como um levantamento dos estudos já realizados sobre a temática.
- Construir um STI para o ensino de Equivalência Lógica Proposicional que leve em conta a individualidade dos alunos.
- Avaliar o STI desenvolvido e suas potencialidades através do *feedback* de especialistas da área de Matemática, Computação e Educação, assim como dados recolhidos junto aos alunos e ao professor.
- Aplicar o STI junto ao ensino presencial de Lógica aos alunos do primeiro semestre do curso de Licenciatura em Computação da UFPR, Setor Palotina.

### 3 FUNDAMENTAÇÃO TEÓRICA

#### 3.1 TEORIAS DE APRENDIZAGEM

O presente trabalho foi embasado principalmente nas teorias de aprendizagem mediacionais: aprendizagem significativa de Ausubel (2003) e teoria socio-interacionista de Vygotsky (1978), uma vez que a proposta inclui trabalhar o ensino de Lógica na Zona de Desenvolvimento Proximal (ZDP) do aluno, utilizando inicialmente conceitos básicos para que se possa evoluir por meio da interação com o Tutor. Porém, o trabalho carrega alguns traços de outras teorias, como, a teoria da construção do conhecimento de Piaget (1975), uma vez que apresentamos o conteúdo por meio de desafios, com a intenção de causar o desequilíbrio cognitivo<sup>1</sup>. Também nos baseamos em traços da teoria do condicionamento operante de Skinner (1963), uma vez que, junto com os *feedbacks*, o sistema provê recompensas visuais ao aluno, transmitindo emoções e cores diferenciadas conforme sua resposta. Outro ponto trabalhado pelo tutor é o fornecimento de conteúdos que correlacione com conceitos que o aluno já conhece, processo este que é definido por Ausubel (2003) como aprendizagem significativa.

O condicionamento clássico estuda basicamente o comportamento observável, sendo portanto, um estudo conhecido como behaviorista, no qual não distingue o homem e o bruto (WATSON, 1913), ou seja, a aprendizagem é diretamente observada mediante a resposta do aluno. Os estudos de Watson (1913) tiveram base nos pensamentos de Pavlov e Gantt (1941) sobre reflexos condicionados (considerados uma das primeiras abordagens realmente objetivas e científicas sobre o estudo da aprendizagem), no qual supõe-se a inclusão de respostas pelo próprio comportamento, tais respostas podem ser observadas e relacionadas com eventos que as precedem e sucedem (estímulos e consequências, respectivamente).

Skinner (1963) tinha como objetivo criar as leis e princípios do comportamento humano. Apesar de seus estudos não terem atingido aquilo que almejava, teve grande influência nos estudos iniciais sobre *Psicologia*, além de ter se tornado referência dentre as teorias associacionistas de condicionamento com sua teoria do condicionamento operante. Este se diferencia do behaviorismo radical pelo fato de levar em conta também alguns eventos não observáveis, isto é, eventos internos como pensamentos, sentimentos, cognições, sonhos, fantasias, etc (SKINNER, 1963).

Dentre as teorias mediacionais, temos a psicologia genético-cognitiva, onde destacamos Piaget (1975) e Ausubel (2003), além da psicologia genético-dialética que tem Vygotsky (1978) como principal pesquisador. Neste sentido, a teoria de Piaget (1975) considera que, para que ocorra a aprendizagem, é necessário existir situações de desequilíbrio para o aluno, já que o processo de construção do conhecimento acontece por meio dos processos de assimilação, acomodação e adaptação. A assimilação é o primeiro contato com o objeto do conhecimento,

---

<sup>1</sup>Tentativa de acomodação do conhecimento pelas estruturas cognitivas.

uma maneira de acontecer este contato é por meio de desafios, que nos levam ao desequilíbrio cognitivo, ou seja, as estruturas cognitivas estão em tentativa de acomodar o novo conhecimento. A adaptação é o ponto de equilíbrio entre os dois processos anteriores e o indivíduo precisa manter uma estabilidade entre eles para que continue aprendendo. Sendo assim, podemos deduzir que o processo de ensino e aprendizagem acontece pela reorganização cognitiva, logo, o ideal para que aconteça o aprendizado é que existam desafios, provocando assim o desequilíbrio.

Ainda na psicologia genético-cognitiva, Ausubel (2003) utiliza o conceito de aprendizagem significativa como ponto central de sua teoria, a qual se baseia na compreensão das estruturas cognitivas, na organização dos conteúdos, nas ideias e nos conceitos já assimilados pelo indivíduo para explicar o processo de aprendizagem. O conhecimento é organizado de forma hierárquica, de modo que conhecimentos específicos são assimilados conforme o indivíduo os correlacionam a conhecimentos mais amplos, obtendo uma rede de conceitos, que são chamadas estruturas cognitivas. Em outras palavras, aprendizagem significativa se dá quando ancoram-se conceitos do objeto de aprendizagem à conhecimentos prévios do aprendiz (AUSUBEL, 2003). De acordo com Sacristán e Gómez (2009, p.38) “a chave da aprendizagem significativa está na vinculação substancial de novas ideias e conceitos com a bagagem cognitiva do indivíduo”.

A teoria sócio-interacionista de Vygotsky (1978), defende que o indivíduo humano nasce com potencialidades de aprendizado diferenciadas dos outros animais, porém para que sejam desenvolvidas é necessário que seja socialmente mediado, ou seja, as relações sociais estão diretamente ligadas ao processo de desenvolvimento intelectual. Outro ponto relevante de sua teoria é o conceito de Zona de Desenvolvimento Proximal (ZDP) e Zona de Desenvolvimento Real (ZDR). A primeira representa a diferença entre a capacidade do indivíduo de resolver problemas por si própria e a capacidade de resolvê-los com ajuda de alguém e a segunda, portanto, abrange todas as funções e atividades que o indivíduo consegue desempenhar por seus próprios meios (VYGOTSKY, 1978). Ou seja, ZDR é o conteúdo que o aluno domina e ZDP é o conteúdo que ele é capaz de dominar com auxílio externo.

Diferente da teoria piagetiana, a teoria socio-interacionista de Vygotsky (1978) defende que uma característica primordial da aprendizagem é a necessidade de interação com o ambiente de convívio, o que desperta vários processos de desenvolvimento internos. As diversas teorias de aprendizagem, têm suas devidas relevâncias na literatura. Dessa forma, utilizaremos algumas das características citadas acima, de maneira que possamos fazer uma mescla entre elas para que o STI - Lógica Livre consiga mediar o ensino de forma eficaz.

### 3.2 O ENSINO DE LÓGICA

A disciplina de Lógica está presente em muitos cursos no âmbito do ensino superior, porém, segundo Lodder, Passier e Stuurman (2008), os conteúdos da disciplina “Lógica Matemática” assumem um valor elevado quando se trata de cursos relacionados a Computação. Mesmo sabendo de tal importância, segundo Grivokostopoulou, Perikos e Hatzilygeroudis (2013) muitos alunos tem dificuldade de compreendê-la, talvez pelo nível de abstração dos conteúdos

formais ou pelos métodos tradicionais de ensino.

Ao ensinar conteúdos matemáticos, não podemos ignorar a formalidade dos conteúdos, pois ela é necessária para que não tenhamos ambiguidade nas resoluções. Acreditamos que o conceito de Transposição Didática proposto por Chevallard (1991) seja uma solução que minimize esse *gap* semântico<sup>2</sup>. O conceito de Transposição Didática consiste em submeter um conteúdo a um conjunto de transformações adaptativas que irão torná-lo apto a ser ensinado e aprendido com mais facilidade (CHEVALLARD, 1991). Dessa forma não é necessário abandonar a formalidade dos conteúdos, mantendo assim o rigor científico do ensino.

A Transposição Didática pode ser feita de inúmeras formas, mas é importante que seja de forma significativa. Para isso Ausubel (2003) propõe aproximar-se da realidade do aluno, ancorando o conhecimento com o que o aluno já conhece.

Tendo isso em mente, Duval (1993) desenvolveu a Teoria dos Registros de Representação Semiótica, que estuda a influência de representações de objetos matemáticos durante o aprendizado, objetos estes que são denominados Registros de Representação Semióticas. Os objetos matemáticos são estritamente abstratos e diferentemente de outros domínios científicos, não são percebidos ou vistos (por meio de microscópio, telescópio, etc) (DUVAL, 1993). Isso torna os Registros Semióticos uma maneira típica de representar objetos matemáticos. Geralmente os Registros Semióticos são apresentados de forma simples e direta, de modo que o aprendiz possa fazer rapidamente a ligação entre o objeto e a representação.

Outra característica dos Registros de Representação Semiótica é a possibilidade de serem substituídos facilmente. A ideia é que, com o tempo, o aluno consiga enxergar várias formas de aplicação para o mesmo objeto. Substituí-los é uma boa prática para que se possa entender essa dinâmica dos objetos Matemáticos (DUVAL, 1993).

Para que um STI obtenha sucesso no ensino de “Lógica Matemática”, acreditamos que é necessário manter as formalidades dos termos técnicos e o rigor científico dos conteúdos, no entanto, convém utilizar-se da Transposição Didática, contextualizando os conteúdos.

### 3.3 CRITÉRIOS ERGONÔMICOS PARA O DESENVOLVIMENTO DE *SOFTWARE*

Segundo Catapan et al. (1999), a potencialidade pedagógica de um *Software* está diretamente ligada a sua qualidade ergonômica, ou seja, é necessário que se tenha o máximo de conforto, segurança e eficácia no que se refere à usabilidade do sistema.

Scapin e Bastien (1997) elencam importantes critérios para avaliação da qualidade ergonômica da interação com os sistemas computacionais. Seguem abaixo tais critérios, sobre os quais iremos explicar posteriormente:

---

<sup>2</sup>diferença entre o entendimento do interlocutor e a mensagem passada pelo locutor.

1. *Guidance*
  - a) *Prompting*
  - b) *Grouping and distinguishing items*
    - i. *Grouping and distinguishing items by location*
    - ii. *Grouping and distinguishing items by format*
  - c) *Immediate feed-back*
  - d) *Legibility*
2. *Workload*
  - a) *Brevity*
    - i. *Conciseness*
    - ii. *Minimal actions*
  - b) *Information density*
3. *Explicit control*
  - a) *Explicit user actions*
  - b) *User control*
4. *Adaptability*
  - a) *Flexibility*
  - b) *User's experience*
5. *Error management*
  - a) *Error protection*
  - b) *Quality of error messages*
  - c) *Error correction*
6. *Consistency*
7. *Significance of codes*
8. *Compatibility*

O critério 1, *Guidance* (orientação), está relacionado aos meios disponíveis para instruir o usuário ao longo de sua interação com o sistema, facilitando a utilização e o aprendizado. Esse critério está subdividido em outros quatro itens: *Prompting* (Entrada de dados), *Grouping and distinguishing items* (Agrupar e distinguir itens), *Immediate Feedback* (Retorno Imediato) e *Legibility* (Legibilidade).

As orientações devem possibilitar ao usuário saber quais ações possíveis e suas consequências a qualquer momento da interação. A facilidade de uso leva a melhores desempenhos, diminuindo os erros.

O critério 2, *Workload* (Carga de Trabalho), refere-se às ações que o usuário precisa realizar ao utilizar o *software*, este critério está subdividido em dois itens: *Brevity* (Brevidade) e *Information Density* (Densidade de Informação). O ideal é que o sistema ofereça o máximo de informação em poucas ações, pois, quanto maior a carga de trabalho, maior a probabilidade

do usuário cometer erros, além de diminuir o desempenho de uso.

O critério 3, *Explicit control* (Controle explícito), divide-se em dois itens: *Explicit user actions* (Ações explícitas do usuário) e *User control* (Controle de usuário). O sistema será melhor aceito pelos usuários se estes tiverem controle e conseguirem visualizar os impactos de suas ações. Ou seja, o sistema deve obedecer aos controles do usuário, evitando comportamentos que não lhe foram solicitados.

Quanto ao critério 4, *Adaptability* (Adaptabilidade), a interface deve se adaptar aos usuários, contextualizando-se com suas necessidades e experiências. Este critério se divide em dois itens: *Flexibility* (Flexibilidade) e *User's experience* (Experiência dos usuários).

No contexto tecnológico atual, surge a necessidade de adaptação dos sistemas a diferentes plataformas, incluindo dispositivos móveis (*mobile*), para que possam ser atendidas as demandas de uma quantidade cada vez maior de pessoas.

O critério 5, *Error management* (Gerenciamento de erros), é subdividido em três itens: *Error protection* (Proteção contra erros), *Quality of error messages* (Qualidade das mensagens de erro) e *Error correction* (Correção de erros). Baseia-se, portanto, em meios de prevenir, reduzir e corrigir os erros, quando ocorrem. Interrupções causadas por erros têm consequências negativas, aumentando o número de interações e dificultando a realização das tarefas.

Para atender ao critério 6, *Consistency* (Consistência), o *design* da interface deve manter um padrão de comandos, procedimentos e rótulos que façam sentido ao usuário, se adaptando ao contexto da tela ou da seção atual. O sistema deve ser previsível, diminuindo a curva de aprendizagem<sup>3</sup> e, conseqüentemente, o número de erros do usuário. A falta de consistência de uma interface diminui o desempenho de uso.

O critério 7, *Significance of codes* (Significância de códigos), propõe utilizar nomes e ícones significativos e apropriados às ações as quais se referem. Códigos inapropriados podem culminar em erros por parte dos usuários.

O critério 8, *Compatibility* (Compatibilidade), diz respeito à coerência entre os ambientes e suas aplicações, nos quais linguagem de comunicação, *design* e outras características do sistema devem ser compatíveis com o público alvo, de modo a diminuir a quantidade de informações a serem interpretadas pelo usuário.

Em seu trabalho, Scapin e Bastien (1997) salientam que “O conjunto de critérios ergonômicos apresentados, foi extensivamente testado e tem sido usado como uma estrutura para o diagnóstico e o relato de avaliações de interface”.

### 3.4 PRINCÍPIOS GERAIS PARA ENGENHARIA DE SOFTWARE

Para que um *software* seja bem sucedido, ele deve atender às necessidades do usuário, operar perfeitamente durante um longo período, ter fácil modificação e fácil utilização (PRESSMAN, 2005). Para satisfazer estes requisitos, o *software* deve contemplar os sete

---

<sup>3</sup>Esforço necessário para aprender determinado conteúdo.



princípios gerais propostos por Hooker (1996 apud PRESSMAN, 2005) que serão abordados nos parágrafos a seguir.

O primeiro e mais importante dos princípios é a **razão de existir**. Um software precisa agregar valor aos seus usuários, caso isso não seja verdade, seu desenvolvimento não se faz necessário.

No segundo princípio, o autor propõe que o *software* seja **tão simples quanto possível**. Um sistema simples é mais fácil de compreender e manter, porém, isso não quer dizer que características devam ser descartadas em nome da simplicidade. Ou seja, simplificar exige análise e trabalho. Uma boa forma de simplificar um projeto sem perder suas características é organizá-lo, dividindo em módulos.

O terceiro princípio alerta-nos para **manter a visão**, pois sem isso o projeto pode se tornar uma “colcha de retalhos”, unindo pedaços incompatíveis. Apesar de que a divisão do software em módulos é uma boa prática, não podemos comprometer a visão arquitetônica geral.

O quarto princípio reforça que **o que um produz, outros consomem**, ou seja, devemos projetar *software* para que outras pessoas também consigam entendê-los. Um *software* de qualidade deve ter uma vida útil longa, e para que isso ocorra, é necessário que tenha fácil manutenção, pois outras pessoas irão manter e estender o sistema.

Quanto maior o tempo de vida, maior o valor agregado do sistema, portanto o quinto princípio sugere **estar aberto para o futuro**, o ideal é que o sistema esteja preparado para mudanças e adaptações, até mesmo de tecnologias que ainda não existem, isso provavelmente conduzirá à reutilização e melhorias futuras.

**Planejar com antecedência, visando a reutilização** é o sexto princípio proposto pelo autor. Esta prática reduz o custo e aumenta o valor tanto dos componentes projetados quanto dos sistemas aos quais serão incorporados.

O sétimo princípio nos conduz a **pensar** antes de agir. Analisar algumas coisas diminui a probabilidade de erros e aumenta a confiabilidade no que se está fazendo.

Ao vivenciar estes princípios, muitas das dificuldades encontradas no desenvolvimento de sistemas complexos podem ser eliminadas, pois eles ajudam a estabelecer uma forma de praticar a engenharia de *software* de forma segura.

### 3.5 REPRESENTAÇÃO DO CONHECIMENTO E CONCEITOS INERENTES

A Inteligência pode ser entendida como a capacidade de compreender e resolver novos problemas e conflitos adaptando-se a novas situações. Mais precisamente, Sobral (2013) define que inteligência é a ligação do concreto ao abstrato, capacitando o sujeito à enfrentar novas situações a fim de resolver problemas. Assim pode-se entender que inteligência é equivalente a capacidade de aprender, inventar ou criar.

Mas o que seria, então, Inteligência Artificial? Segundo Nilsson (1998) Inteligência Artificial (IA) é o desempenho inteligente de artefatos. Apesar de simples, essa definição se

mostra bem completa, entendendo que artefato é um objeto, ou parte de um objeto feito pelo homem, a Inteligência Artificial é a designação para o comportamento inteligente de qualquer artefato. Porém, os computadores vêm sendo o artefato mais atrativo para esse tipo de implementação.

Conforme a prática de construir artefatos inteligentes foi crescendo, descobriu-se a necessidade de formalizar paradigmas com o propósito de relacionar o campo da cognição com os artefatos utilizados. Dentre esses paradigmas destacam-se:

- **Paradigma Simbólico** - seu principal conceito é o desenvolvimento de modelos através da manipulação de símbolos. Tais modelos computacionais são baseados em representações explícitas com estruturas agregadas, que são construídas através dos símbolos e combinações sintáticas desses símbolos (SUN, 1999). Exemplos de representações simbólicas são as árvores de decisões, as regras de produção e a Lógica.
- **Paradigma Conexionista** - estabelecido mais recentemente, o paradigma conexionista se dá por vários pontos insatisfatórios nos modelos simbólicos, especialmente pelo fato do simbólico não ter um bom desempenho com processamento massivo e robusto de maneira eficiente. Um exemplo clássico de aplicação conexionista são as redes neurais. Como apontam Russell e Norvig (1995), o modelo conexionista foi, por muito tempo, visto como concorrente do simbólico, questionando o segundo quanto à veracidade da manipulação de símbolos ter qualquer função realmente explicativa a respeito da cognição. Entretanto, nos dias atuais, ambos os paradigmas são vistos como complementares.
- **Computação Evolutiva** - é entendida por Giusti (2010) como um conjunto de técnicas de IA que utiliza computação para solucionar problemas de otimização<sup>4</sup>. Tais técnicas aparecem nas mais diversas formas como, por exemplo, o Algoritmo Genético (AG). Este, utiliza da busca em feixe local<sup>5</sup> por meio de técnicas de cruzamento e mutações de possíveis soluções. A partir de um conjunto de estados gerados aleatoriamente (ou com algum outro método estocástico), denominado população, são gerados novos estados que serão avaliados pela função de adaptação (ou função *fitness*). Esta função deve retornar valores que mensuram a qualidade do estado a fim de preservar os melhores e mais aptos à se reproduzirem (RUSSELL; NORVIG, 1995). Essa abordagem foi inspirada na teoria da seleção natural dos seres vivos de Darwin (1859).

Além desses paradigmas, a IA teve início com mecanismos de busca para solucionar problemas. Entretanto, Russell e Norvig (1995) discorrem sobre tais abordagens serem considerados métodos fracos, por não serem capazes de resolver problemas complexos, mas apenas alguns casos típicos.

---

<sup>4</sup>Problemas que visam determinar pontos extremos de uma função, com intuito de encontrar máximos ou mínimos.

<sup>5</sup>Analisa um feixe de soluções prováveis e seleciona as melhores, repete este procedimento criando novos feixes a partir destas seleções e melhorando as soluções.

A fim de melhorar este quadro, surgiram os Sistemas Especialistas (SE), os quais têm como objetivo simular o conhecimento ao nível comparável de um profissional especialista em determinada área. Esses sistemas já não resolvem problemas genéricos, sendo notável a maior precisão com que solucionam problemas específicos. Os SE estão contidos em um grupo de sistemas denominados Sistemas Baseados em Conhecimento (SBC), que consistem na utilização do conhecimento representado explicitamente para resolver problemas, os SBC são desenvolvidos com o propósito de serem aplicados em problemas que requerem uma quantidade considerável de conhecimento humano e de perícia para serem resolvidos.

Os STI são um tipo de SBC. Suas principais características são a capacidade de interagir com o aprendiz e a de adaptar-se conforme as necessidades do mesmo, podendo assim, oferecer instruções diretas e conteúdos personalizados.

O principal diferencial dos STI em relação a outros sistemas computacionais de ensino é a possibilidade de fornecer um ensino personalizado (ELSOM-COOK, 1987). Porém, para que isso seja possível, não basta somente personalizar o conteúdo: é importante também ter um método de avaliação que possibilite essa personalização. Conforme Castro (2005 apud RISSOLI; GIRAFFA; MARTINS, 2006, p. 42) “Não se melhora o ensino sem melhorar a sua avaliação”. Para a implementação desses processos avaliativos personalizados, podemos fazer uso, por exemplo, de mecanismos de Representação de Conhecimento.

Alguns mecanismos de Representação de Conhecimento são constituídos de sistemas baseados em regras, pois geralmente são utilizados para fornecer recomendações ou diagnósticos, tanto para tomada de decisão como para avaliação. Esses sistemas são compostos por uma base de conhecimento<sup>6</sup>, uma base de fatos<sup>7</sup> e um mecanismo de inferência<sup>8</sup> (COPPIN, 2010).

Segundo Py (2009), são três as principais teorias de Representação do Conhecimento: (i) a Lógica, que utiliza representações específicas interpretadas, tais como proposições ou predicados lógicos; (ii) as Redes Semânticas, que são constituídas por nós conectados de forma hierárquica a um conjunto de arcos, onde os nós representam objetos ou predicados e os arcos representam relações binárias entre estes objetos; e (iii) os Quadros (*Frames*), também conhecida como *Scripts*, que além de possibilitarem estruturas hierárquicas (como as Redes Semânticas), permitem também representar as estruturas internas dos objetos.

Pela sua simplicidade, a Lógica permite representação de forma explícita, por isso este método está presente na maioria dos SBC conhecidos. A linguagem Prolog<sup>9</sup> se baseia no método lógico de representação de conhecimento, utilizando-se de predicados lógicos sua sintaxe possui alto nível de abstração, tornando assim a representação do conhecimento clara e objetiva.

---

<sup>6</sup>Base que representando computacionalmente a informação que um especialista utiliza

<sup>7</sup>Base que comporta os acontecimentos, informações que o sistema entenderá como verdade enquanto estiverem armazenadas

<sup>8</sup>Responsável pelas deduções do sistema, para isso utiliza as bases de fato e conhecimento

<sup>9</sup>Linguagem baseada no paradigma de Lógica Matemática, amplamente utilizado para desenvolvimento de sistemas inteligentes com aspectos linguísticos computacionais.

Quando o domínio do conhecimento apresenta necessidades de representar incertezas, a Lógica *Fuzzy* é uma das técnicas utilizadas para representá-lo. Ela fornece modelos matemáticos nos quais as incertezas possam ser tratadas de maneira precisa e rigorosa (FERNANDES et al., 1996).

A Lógica clássica trata as proposições ou predicados de forma discreta, ou seja, as proposições ou predicados são verdadeiros ou falsos (não outro). Já a lógica *Fuzzy* nos permite representar valores reais dentro do intervalo  $[0,1]$ , assim um predicado pode ser verdadeiro e falso, ou mais verdadeiro do que falso, ou ainda, mais falso do que verdadeiro. Dessa forma, temos o conceito de graus de pertinência. O grau de pertinência para cada conjunto pode ser entendido como grau de pertencimento, pois um elemento pode não pertencer inteiramente a um conjunto, mas a vários ao mesmo tempo. Com isso, cada elemento tem seu grau de pertinência para cada conjunto.

A lógica *Fuzzy* altera a pertinência dos elementos, fazendo com que estes se adéquem aos conjuntos *Fuzzy* em que seu grau de pertinência varia em um intervalo  $[0,1]$  (KLIR; YUAN, 1995). Com esta modelagem, a Lógica *Fuzzy* permite representar os elementos de forma qualitativa, como por exemplo: Pouco; Médio; Alto; Muito alto, etc.

Esta Lógica tem se mostrado mais adequada no tratamento de determinados tipos de imperfeições da informação, podendo, de maneira preliminar e mais objetiva, ser definida como uma metodologia capaz de capturar informações vagas, em geral descritas em linguagem natural, e convertê-las para um formato mais rígido e numérico (RISSOLI; GIRAFFA; MARTINS, 2006, p.42).

Este processo de transformação de informações incertas em modelos numéricos, é denominado *Fuzzyficação*. Neste, o sistema faz um mapeamento dos valores incertos para valores numéricos dentro dos conjuntos *Fuzzy*. Análogo a isso e baseados nas regras de produção *Fuzzy*, existe o processo de *Defuzzyficação*, que consiste em retornar os dados numéricos em um formato próximo de linguagem natural.

O presente trabalho faz uso do paradigma simbólico da Inteligência Artificial, de modo que não se fez necessário realizar processamentos massivos e paralelos, tampouco tivemos problemas de otimização. A Representação do Conhecimento do Sistema Especialista é implementada por meio da programação lógica com a Linguagem Prolog, utilizando ainda, a Lógica *Fuzzy* para possibilitar as deliberações do Tutor em ambientes de incerteza, como veremos na seção 5.

### 3.6 TECNOLOGIA DOS COMPILADORES E PROCESSOS DE ANÁLISE

Os STI podem utilizar os conceitos dos compiladores para analisar e interpretar a entrada de dados vinda dos usuários. A fase responsável por auxiliar o STI neste processo, é chamada de análise gramatical. O processo de análise gramatical em computadores surgiu por volta dos anos 1950, nesta mesma época começaram a surgir os primeiros compiladores, dentre

eles o do FORTRAN (BACKUS et al., 1957). Um compilador é um programa que traduz algo escrito em determinada linguagem para outra linguagem, de modo que, falar de compilador, nos remete a pensar em linguagens de programação sendo transformadas em linguagens de máquina. Porém, existem inúmeras áreas que não estão diretamente ligadas à execução de códigos, mas que utilizam conceitos de compiladores (AHO; SETHI; ULLMAN, 1986), por exemplo, formatadores de texto, compiladores de silício e interpretadores de *queries*<sup>10</sup>. A análise do código fonte<sup>11</sup> é subdividida em três partes:

- A análise léxica recebe um fluxo de caracteres e têm como objetivo dividi-los em *tokens* (sequências de caracteres que contém um significado) e validar a correteude individual desses *tokens*.
- A análise sintática é feita de forma hierárquica, de modo que os *tokens* são organizados em forma de árvore gramatical. Esta árvore deve validar a escrita dentro da linguagem, levando em conta a precedência de operadores e sinais, caso existam. No entanto, não se preocupa com o significado da escrita.
- A análise semântica utiliza a estrutura hierárquica gerada na análise sintática para identificar os operadores e operandos das expressões e enunciados. O objetivo desta análise é justamente extrair o significado da expressão. Em qualquer uma das análises, se o compilador encontrar um erro no código fonte, ele deve parar o processo e informar o erro.

Esse processo de análise é amplamente utilizado, permitindo definir o próprio protocolo de divisão dos *tokens*, gramática e valores semânticos. Ao fazer isso, é possível criar, por exemplo, uma nova linguagem de programação, ou então a própria gramática para entrada dos dados de um STI, no nosso caso, Expressões da Lógica Proposicional.

### 3.7 ARQUITETURAS DE SISTEMAS TUTORES INTELIGENTES (STI)

A ideia de utilizar a inteligência artificial na forma de ferramentas computacionais de ensino transformou, ainda na década de 1970, os Sistemas de Instrução Assistida por Computador (CAI) em Sistemas de Instruções Assistidas por Computadores Inteligentes (ICAI), também denominados Sistemas Tutores Inteligentes (STI) (GIRAFFA; VICCARI, 1998). Várias pesquisas com STI foram desenvolvidas, como o Scholar (CARBONELL, 1970; CARBONELL, 1971), que utilizava o método socrático<sup>12</sup> para o ensino de Geografia da América do Sul. O tutor buscava expor o aluno a sucessivas questões a fim de analisar hipóteses, descobrir contradições, realizar inferências e formular novas questões com base no erro do aluno. Além dessa particularidade, uma das características que chamou a atenção dos pesquisadores foi a arquitetura utilizada no desenvolvimento do *software*. Ao criar o Scholar, Carbonell (1970) apresentou uma nova

<sup>10</sup>Pela falta de correspondência na língua portuguesa, optamos deixá-la na linguagem original, no contexto computacional este termo significa comando que pergunta algo, interroga ou busca.

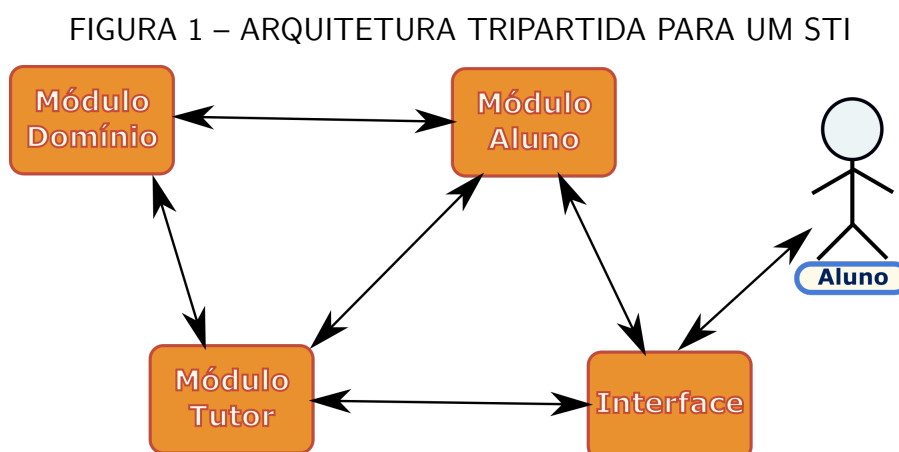
<sup>11</sup>código escrito em determinada linguagem de programação.

<sup>12</sup>Técnica que utiliza o diálogo, no qual o professor, empregando perguntas, conduz o aluno a um processo de reflexão e descoberta.

organização genérica para desenvolvimento de STI, denominada Arquitetura Tripartida, também conhecida nos dias atuais como Arquitetura Clássica ou Tradicional. Esta arquitetura consiste em modularizar o STI em, no mínimo, três módulos:

- **Módulo do Aluno** - Este módulo é o que possibilita o sistema tratar os alunos de forma individual, uma vez que ele contém Representações de Conhecimento (RC) de cada um. Com isso, temos um avaliador contínuo e individual, possibilitando a alteração do fluxo do ensino para cada aluno.
- **Módulo Domínio** - Este deve conter um Sistema Especialista (SE), o qual tem o conhecimento de todos os conteúdos a serem ensinados ao aluno. Sendo assim, todas as questões, respostas corretas e correção das respostas do aluno são delegadas a ele.
- **Módulo Tutor** - É responsável pelas tomadas de decisões no ensino. Para isso, leva em conta informações individuais sobre o aluno (contidas no Módulo do Aluno). De forma genérica, as tomadas de decisões do Tutor consistem em elaborar exercícios personalizados para cada aluno, prover a estratégia de ensino adequada, bem como uma metodologia para o processo de ensino e aprendizagem. Em um STI concebido na teoria sócio-interacionista de aprendizagem, conforme Vygotsky (1978), é este módulo que tem a responsabilidade de prover um ensino que esteja dentro da Zona de Desenvolvimento Proximal (ZDP) dos alunos.

A FIGURA 1 representa a forma de comunicação bidirecional entre os módulos, denotando que cada um contém responsabilidades específicas, porém eles devem comunicar-se.



FONTE: Imagem do autor (2018)

O Módulo do Aluno, por exemplo, deve prover informações sobre os alunos ao Tutor em tempo de execução do sistema. Análogo a isso, o Tutor deve enviar de volta os acertos, erros e deliberações do aluno para que o Módulo do Aluno atualize a representação dos conhecimentos a cada interação. O Módulo Domínio, por sua vez, deve estar à disposição do Tutor, a todo momento, pois é o Especialista que tem o conhecimento de conteúdo a ser ensinado e que faz as correções das respostas que o usuário fornece.

Derivada da Arquitetura Tripartida, Kaplan e Rock (2011) propuseram uma arquitetura que modulariza o sistema, mantendo os três módulos principais, porém o Modelo Instrucional (equivalente ao Módulo Tutor) não mais recebe informações da Interface, apenas é responsável por definir a estratégia de ensino. O único módulo que recebe informações da Interface é o Modelo do Especialista (equivalente ao Módulo de Domínio) que, por sua vez, conversa com o Modelo do Aluno. Nesta proposta, o Modelo Instrucional não se comunica com o Modelo do Especialista, que é questionável, pois o Modelo Instrucional precisa, de alguma forma, ter acesso ao conteúdo para prover o ensino.

Uma outra arquitetura é proposta por Wenger (1988) que preconiza modularizar ainda mais o sistema por meio de cinco módulos: Especialista; Aluno; Perícia Pedagógica; Interface; e um módulo denominado Tomada de Decisões e Conhecimento, o qual centraliza a comunicação entre todos os outros módulos. Este modelo possui um número maior de módulos, porém o fato de centralizar toda a comunicação no módulo de Tomada de Decisões e Conhecimento, pode gerar um ponto único de falhas ou um gargalo e prejudicar a escalabilidade.

### 3.8 TRABALHOS RELACIONADOS

Apesar de a literatura apresentar vários modelos de arquitetura, a arquitetura tradicional continua sendo utilizada por outros STI, como por exemplo o Tutor-Prolog (VICCARI, 1990). Trata-se de um STI cuja arquitetura inspira-se na organização tradicional, pois inclui o Modelo do Domínio, o Modelo do Aluno e as Estratégias de Ensino, além de agregar outros componentes externos aos módulos centrais. Assim como o Scholar de Carbonell (1970), o Tutor-Prolog também se utiliza do método socrático, visando criar um ambiente para resolução voltado ao paradigma de Programação Lógica. O tutor faz uso de um subconjunto da linguagem de programação Prolog e apresenta a capacidade de adequar-se ao estilo de cada aluno tornando a interação mais dinâmica e agradável (VICCARI, 1990). Ele utiliza um modelo híbrido para as questões instrucionais: o método socrático e a exploração direta<sup>13</sup>, com o uso de tentativa e erro (GIRAFFA; VICCARI, 1998). O ambiente apresenta três níveis separados, possibilitando: a apresentação do conteúdo sobre programação lógica; a criação de programas por parte do aluno, passíveis de correções automáticas e *feedback*; e a consulta a uma base de dados contendo exemplos para apresentar ao aluno.

Outro estudo realizado para o ensino de representação do conhecimento por meio da lógica é o *FOL Equivalence System (First-Order Logic Equivalence System)*, que é um sistema *desktop* voltado para a representação de conhecimento e raciocínio por meio da Lógica de Primeira Ordem e Linguagem Natural (NL). Nele, o aluno pode selecionar uma das fórmulas e começar sua conversão desta para uma fórmula equivalente. A cada passo realizado pelo aluno lhe é possível recorrer a um *feedback* baseado em suas ações e seu estado de conhecimento (GRIVOKOSTOPOULOU; PERIKOS; HATZILYGEROUDIS, 2013).

---

<sup>13</sup>Método onde o tutor conduz o aluno no decorrer do curso (VICCARI, 1990).

Se uma regra selecionada não puder ser utilizada, o sistema apresenta uma mensagem explicitando o motivo do presente erro. O estudo de Grivokostopoulou, Perikos e Hatzilygeroudis (2013) consistiu-se na aplicação de pré teste e pós teste utilizando um grupo experimental e um de controle, sendo que apenas o primeiro utilizaria o sistema. Foi observado que houve um menor tempo gasto por parte do primeiro grupo para resolver as questões do pós-teste em comparação com o segundo grupo. Além de que 84% dos alunos tomaram o *feedback* do sistema como muito útil para o aprendizado (GRIVOKOSTOPOULOU; PERIKOS; HATZILYGEROUDIS, 2013).

O IDEAS é outro projeto que fez uso do *feedback* para o ensino de lógica (LODDER; PASSIER; STUURMAN, 2008). Desenvolvido na *Open University Netherlands*, IDEAS é um projeto que consistiu em desenvolver uma ferramenta para ser usada como suporte ao ensino de lógica, reescrevendo fórmulas lógicas na Forma Normal Disjuntiva (FND). A ferramenta possibilita que o aluno aplique regras simples e verifica se a expressão submetida pode ser derivada da expressão previamente fornecida, além de fornecer, a cada passo, um novo *feedback* ao aluno.

Para verificar a eficiência da ferramenta, 23 alunos da *Open University de Netherlands* do período de 2007 a 2008 participaram de um estudo, no qual todos realizaram um pré teste e somente 10 deles realizaram o pós teste. Após a aplicação do teste verificou-se a necessidade de projetar uma ferramenta de *feedback* de qualidade de modo que o conteúdo ficasse mais claro para o aluno. Porém, os resultados contribuíram para iniciar a ampliação do sistema. A partir dessa ferramenta, o grupo criou o LogEx (LODDER; HEEREN; JEURING, 2015).

Por sua vez, o LogEx de Lodder, Heeren e Jeuring (2015), é um Sistema Tutor Inteligente disponível em uma plataforma *Web*<sup>14</sup>, contendo três modalidades de exercícios que o aluno pode realizar: reescrever fórmulas lógicas na Forma Normal Disjuntiva (FND); reescrever fórmulas lógicas na Forma Normal Conjuntiva (FNC); e provar equivalência entre duas fórmulas.

O aluno resolve as atividades obtendo dicas e *feedback* do LogEx, a cada passo. O LogEx também detecta pequenos erros semânticos como, por exemplo, se o aluno possui a expressão  $\neg(p \vee q)$  e digita  $(\neg p \vee \neg q)$ , consequentemente o LogEx detecta que o aluno procurou aplicar a regra De Morgan, e apresenta uma dica face ao erro cometido: “Quando se aplica a regra de DeMorgan, uma disjunção é transformada em uma conjunção”<sup>15</sup> (LODDER; HEEREN; JEURING, 2015). Caso nenhuma regra seja detectada, o LogEx utiliza de um conjunto de erros baseados na regra *Buggy*<sup>16</sup> a fim de oferecer um *feedback* oportuno.

É cabível ao aluno a escolha de exercícios que irá realizar, bem como os diferentes níveis de dificuldade: fácil, normal, difícil e até mesmo a possibilidade de uma entrada manual, no qual o aluno cria seu próprio exercício. Com o uso do LogEx, o aluno aplica as regras de

<sup>14</sup><http://ideas.cs.uu.nl/logex/>

<sup>15</sup>when applying DeMorgan's rule, a disjunction is transformed into a conjunction (Tradução literal).

<sup>16</sup>Conhecido como modelo de perturbação, pois assume que os erros do aluno são decorrentes da concepção errônea de algum conceito ou ausência dele (GAVIDIA; ANDRADE, 2003)



equivalência de uma maneira eficiente, consegue provar equivalências e reescrever as fórmulas na FNC e/ou na FND (LODDER; HEEREN; JEURING, 2015).

No estudo piloto, realizado por meio de um pré teste e pós teste, foi observado que os alunos completaram mais exercícios no pós teste, porém apresentaram mais erros. O nível de dificuldade do pós teste foi ligeiramente maior que o pré teste e uma avaliação em maior escala se mostrou necessária para mensurar melhor os resultados, [...] mas foi observada necessidade de “algumas adaptações, especialmente ao *feedback* no nível estratégico” (LODDER; HEEREN; JEURING, 2015).

O projeto LogEx continuou sendo desenvolvido por Lodder, Heeren e Jeuring (2016), no qual foram realizadas melhorias na interação entre o usuário e o sistema melhorando ainda mais as estratégias de *feedback* e implantando também o *feed forward*, que são como dicas que ajudam o aluno a prosseguir com o exercício. Os experimentos de Lodder, Heeren e Jeuring (2016) consistiram em pré teste e pós teste, em uma amostragem de 12 alunos, onde 6 deles (denominado grupo A) utilizavam o LogEx com todas as suas funcionalidades e os outros 6 (denominado grupo B) não recebiam dicas para os próximos passos. Segundo Lodder, Heeren e Jeuring (2016) ao realizar o pré teste mais alunos do grupo B conseguiram concluir, já no pós teste os resultados foram invertidos, além de que o grupo A cometeu menos erros no pós teste. Outro dado relevante é que os alunos do Grupo A apresentaram mais interesse em utilizar o sistema, de modo que tiveram uma média de tempo de utilização maior que o grupo B (57 minutos contra 41 minutos). no entanto, ainda com os resultados melhorados e com ajuda do *feed forward* a estratégia de *feedback* foi considerada por Lodder, Heeren e Jeuring (2016) um ponto a ser melhorado.

Outro trabalho que se tornou relevante no ensino de Lógica é um STI desenvolvido por Mostafavi e Barnes (2017), denominado Deep Thought. Este sistema utiliza uma modelagem Orientada a Dados<sup>17</sup> (OD) para realizar seus *feedbacks*, além de realizar Mineração de Dados<sup>18</sup> (MD) para reconhecer agrupamento entre os alunos. Dessa forma, além de levar em conta a individualidade dos alunos, ele também é capaz de inferir conteúdos e estratégias educacionais por medidas de similaridade com os demais utilizadores do sistema. O Deep Thought se mostrou muito eficiente no ensino do conteúdo de Resolução de Problemas e Provas Lógicas. Apesar do Deep Thought ter se mostrado promissor, ele não tem como foco o conteúdo de Equivalência Lógica Proposicional, que ao nosso ver, são essenciais para o entendimento dos demais.

Uma característica comum aos estudos anteriores é a necessidade de melhorias no *feedback*, de forma mais estratégica e oportuna, além de que em alguns deles (por exemplo o LogEx) não está clara a consideração quanto a Zona de Desenvolvimento Proximal do Aluno que, segundo Vygotsky (1978), é essencial para a construção do conhecimento. Dessa forma, constatamos que as pesquisas referentes a este tema ainda não se esgotaram. Nossas contribuições referentes a construção de STI serão apresentadas nas próximas seções.

---

<sup>17</sup>Leva em conta os dados dos alunos anteriores, realizando agrupamentos de alunos por característica das respostas.

<sup>18</sup>Técnica utilizada para encontrar anomalias, padrões, realizar agrupamento ou classificação dos dados.

## 4 PROCEDIMENTOS METODOLÓGICOS

### 4.1 METODOLOGIA

O presente trabalho utiliza a metodologia *Development Research*, que, segundo Alan et al. (2004 apud ELLIS; LEVY, 2010) consiste em criar e avaliar artefatos de TI (Tecnologia da Informação), destinados a resolver problemas organizacionais identificados por meio de investigações. Segundo Richey e Klein (2014), este método tem como pressuposto desenvolvimento baseado em dados, os quais são sistematicamente derivados da prática. Ellis e Levy (2010) ressaltam, também, que o desenvolvimento do produto não deve ser confundido com a pesquisa, apesar de estarem diretamente ligados. Uma outra possível definição da metodologia *Development Research* consiste na “investigação disciplinada, conduzida no contexto do desenvolvimento de um produto ou programa, com o objetivo de melhorar o que está sendo desenvolvido ou o próprio desenvolvedor”<sup>1</sup> (HASAN, 2003, p.7).

Dessa forma, nosso trabalho consiste na realização de uma pesquisa sobre os problemas do ensino de Equivalência Lógica Proposicional no âmbito do ensino superior e um estudo sobre os STI existentes na literatura, seguido do desenvolvimento de um STI para auxiliar o ensino do conteúdo de Equivalência Lógica Proposicional bem como avaliação qualitativa e quantitativa do artefato desenvolvido.

O presente estudo é dividido em um conjunto de etapas subsequentes descritas nos parágrafos que seguem.

### 4.2 REVISÃO DA LITERATURA

A primeira etapa consiste na revisão da literatura, no qual: abordamos as principais Teorias de Aprendizagem, bem como alguns conceitos utilizados especificamente no ensino de Lógica e Matemática; analisamos alguns Critérios Ergonômicos para criação de *software* que são utilizados na literatura para avaliar a qualidade da interação com o sistema; consultamos os Princípios Gerais para um bom processo de Engenharia de *software*; analisamos ainda, algumas técnicas de Inteligência Artificial com enfoque em Representação de Conhecimento; realizamos um estudo sobre os processos de análise dos compiladores e suas principais aplicações, pesquisamos sobre os Sistemas Tutores Inteligentes e as principais arquiteturas utilizadas na literatura; e fizemos uma busca pelos trabalhos relacionados ao presente estudo.

---

<sup>1</sup>tradução nossa

### 4.3 VERIFICAÇÃO DOS PROBLEMAS NA APRENDIZAGEM DE LÓGICA E NECESSIDADE DE AUXÍLIO TECNOLÓGICO

Na segunda etapa, verificamos os problemas referentes ao estudo de Lógica. Para isso, foi analisado o cenário educacional do conteúdo de Equivalência Lógica Proposicional no curso de Licenciatura em Computação da UFPR - Setor Palotina. Para averiguar a hipótese de que uma ferramenta educacional seria útil no ensino de Equivalência Lógica Proposicional, aplicamos um questionário a um grupo de alunos que já cursaram a disciplina de Lógica. Este questionário é constituído de duas perguntas de múltipla escolha em que os integrantes da pesquisa podiam responder *online*: A primeira questão é referente ao ensino de Lógica em geral, no contexto da UFPR - Setor Palotina; A segunda é direcionada ao conteúdo de Equivalência Lógica Proposicional, questionando a possível necessidade de ferramentas que proporcionem um melhor entendimento do conteúdo.

### 4.4 DESENVOLVIMENTO DO SOFTWARE

Na terceira etapa, realizamos o desenvolvimento do STI - Lógica livre. A partir dos conceitos adquiridos por meio da revisão da literatura e da conjuntura, bem como da disponibilidade para realização de pesquisas no âmbito do ensino superior na UFPR - Setor Palotina, iniciamos o desenvolvimento do sistema levando em conta as necessidades encontradas no devido contexto.

O desenvolvimento do STI - Lógica Livre foi subdividido em várias partes, as quais iremos explicar nos parágrafos a seguir.

#### 4.4.1 Levantamento de Requisitos

O Levantamento de requisitos foi realizado por meio da análise dos dados obtidos durante a pesquisa, em que foi definido o que o sistema precisa para alcançar os objetivos propostos. Neste sentido foram elencados os seguintes requisitos funcionais:

- Fornecer ao aluno diferentes modalidades de desafios.
- Fornecer explicações elementares sobre lógica proposicional.
- Possibilitar comunicação bidirecional, para que o aluno possa tirar dúvidas com o Tutor.
- Proporcionar *feedbacks* personalizados, baseados na resposta do aluno.
- Prover dicas e orientações para auxiliar o aluno no progresso dos exercícios.
- Dispor de um menu de ajuda para utilização do sistema.
- Fornecer um relatório de desempenho de fácil acesso ao aluno.

Além dos requisitos funcionais, também elencamos os requisitos não funcionais:

- Controle de acesso para usuários.
- Ser Multiplataforma e acessível de qualquer lugar com acesso a *internet*.

- Ter uma interface responsiva, adequando-se aos novos dispositivos *Mobile*.
- Armazenar todas as ações dos usuários durante o uso.
- manter a consistência dos dados armazenados.
- apresentar baixo tempo de resposta aos comandos do usuário.
- Utilizar tecnologias e *frameworks* gratuitas para que o sistema possa ter continuidade na comunidade de *software* livre.

Tais requisitos são necessários para o desenvolvimento de um sistema que possibilite acesso de vários usuários, realizando o controle de dados e garantindo a disponibilidade das funções. Sua interface deve ser agradável e atrativa, a fim de garantir boa experiência de uso, conforme os critérios de ergonomia (*Guidance, Workload, Explicit control, Adaptability, Error management, Consistency, Significance of codes and Compatibility*) (SCAPIN; BASTIEN, 1997).

#### 4.4.2 Ciclo de Vida do Projeto

Para realizar o desenvolvimento, levamos em conta os princípios gerais de Engenharia de *software* citados por Pressman (2005) na seção 3.4 deste trabalho, a fim de minimizar os problemas recorrentes no desenvolvimento. Com o intuito de garantir a segurança e a integridade do *Software*, foram realizados os seguintes testes:

- Testes unitários;
- Testes de usabilidade;
- Testes de funcionalidade;
- Testes de desempenho.

A implantação do sistema ocorreu primeiramente nas dependências da UFPR, sendo que a primeira utilização foi com os alunos do primeiro período do curso de Licenciatura em Computação, para verificar se o STI alcançou os objetivos esperados. Foi verificado se esse STI auxiliou no processo de ensino e aprendizagem de Equivalência Lógica Proposicional. Após a realização dos experimentos o *software* ficou disponibilizado gratuitamente na *Web*<sup>2</sup>.

Pelo fato do STI - Lógica Livre ter sido desenvolvido para ajudar a comunidade, decidiu-se que, após passar pelas fases de teste e implantação seu código deve se tornar aberto, ou seja, disponível para comunidade. Dessa forma, seu ciclo de vida pode se perpetuar e agregar melhorias.

#### 4.4.3 Tecnologias Utilizadas

Sobre as tecnologias que foram utilizadas no desenvolvimento, temos Java como linguagem principal, porém vários frameworks e também outras linguagens foram necessárias para implementação, tais como:

---

<sup>2</sup>[http://lcp.c3sl.ufpr.br/logica\\_livre/](http://lcp.c3sl.ufpr.br/logica_livre/)

- O *Spring Framework*: possibilita uma conexão direta da linguagem JAVA com uma interface *Web*;
- O *JavaCC*: um *framework* para realização de análise léxica e sintática, a qual é utilizada para análise das expressões lógicas de entrada e saída do sistema;
- O *Bootstrap*: um *framework* CSS que possibilita criação de sites com tela responsiva e cores atrativas de forma fácil e rápida;
- O banco de dados *MySQL*: utilizado para armazenar expressões lógicas e as ações realizadas pelos alunos, bem como todos os acertos e erros. Esses dados são utilizados para a tomada de decisão do sistema.
- É utilizada a biblioteca *JFuzzyLogic* para calcular a evolução de cada aluno, levando em conta os níveis de dificuldades em que o mesmo se encontra. É através deste cálculo que o sistema efetua as deliberações sobre os conteúdos e exercícios aplicados.
- A linguagem PROLOG utilizada para implementar a representação de conhecimento do Módulo do Aluno.

#### 4.5 VALIDAÇÃO DO SOFTWARE POR ESPECIALISTAS DAS ÁREAS DE COMPUTAÇÃO, EDUCAÇÃO E MATEMÁTICA

Ao finalizar a primeira versão, realizou-se uma avaliação do *software* por especialistas das áreas de Computação, Educação e Matemática, com o objetivo de validá-lo como ferramenta educacional, bem como agregar melhorias por meio de *feedbacks* construtivos provindos das análises dos especialistas, que serão apresentadas na seção 6.

#### 4.6 APLICAÇÃO DO PRÉ TESTE

A amostra foi constituída por 21 alunos do primeiro período do curso de Licenciatura em Computação da Universidade Federal do Paraná (UFPR) - Setor Palotina.

Com o intuito de mensurar o conhecimento prévio dos alunos perante o conteúdo Equivalência Lógica Proposicional, realizamos um pré teste (apêndice A) com os alunos que estavam cursando a disciplina de Lógica, no período de início do conteúdo proposto. O mesmo pré teste foi aplicado a todos os alunos da turma.

O pré teste foi constituído de vinte e quatro questões agrupadas por Regras de Equivalência Proposicionais. Além disso foi elaborado um exercício, contendo cinco questões, que disponibiliza a expressão ao aluno e solicita que o mesmo aplique qualquer Regra Proposicional que seja possível.

#### 4.7 UTILIZAÇÃO DO SOFTWARE PELO GRUPO DE ESTUDOS

Para mensurar o progresso dos alunos que utilizaram o Tutor em relação aos que permaneceram somente no método tradicional, foi preciso dividir a turma em dois grupos,

grupo de estudo e grupo de apoio, de modo que o grupo de estudo utilizaram o *software* e o grupo de apoio permaneceram nos métodos tradicionais de resolução de exercícios.

Ao final da aplicação do pré teste, foi dividida a turma aleatoriamente, integralizando dez (10) alunos no grupo de estudos e onze (11) alunos no grupo de apoio (também conhecido como grupo de controle). Foi disponibilizado o *link* de acesso do STI - Lógica Livre aos alunos do grupo de estudo para que pudessem realizar o uso do sistema de forma extraclasse. Durante a utilização do STI - Lógica Livre são armazenados todos os dados gerados pelo aluno, os dados armazenados consistem nos erros e acertos, face às questões resolvidas pelos alunos, dados estes que foram estudados posteriormente. Com estes *logs* temos as informações de quem realmente utilizou o sistema e qual foi o seu desempenho individual.

#### 4.8 APLICAÇÃO DO PÓS TESTE

Chegando ao fim da aplicação do conteúdo de Equivalência Lógica Proposicional, foi realizado um pós teste (apêndice B) com os mesmos alunos que participaram do pré teste. A quantidade de exercícios, bem como sua distribuição quanto à abordagem das Regras Proposicionais foram as mesmas do pré teste, porém, as questões aplicadas no pós teste são ligeiramente mais difíceis do que as do pré teste. Após aplicação do pós teste, o sistema foi liberado para uso de toda a classe para que todos os alunos pudessem aproveitar das potencialidades do STI.

#### 4.9 QUESTIONÁRIO SOBRE A EXPERIÊNCIA DE USO DO SISTEMA

Um questionário final, desenvolvido no formato de formulário *online*, o qual foi enviado apenas aos alunos que se cadastraram no sistema, com o intuito de coletar *feedback* sobre sua experiência.

As perguntas foram direcionadas à sua experiência enquanto usuário do *software*, com seis questões, sendo três de múltipla escolha: (i) sobre a impressão geral do sistema (podendo ser ótima, boa, razoável ou ruim); (ii) sobre a qualidade dos conteúdos propostos; e (iii) se os alunos indicam o uso do sistema para turmas posteriores. Além disso foram propostas três questões abertas: (i) solicitando *feedback* e sugestões de melhorias; (ii) questionando sobre a contribuição do sistema no desenvolvimento do conteúdo; e (iii) pontos positivos e negativos sobre o uso do sistema.

#### 4.10 TÉCNICAS DE ANÁLISE DE DADOS

Para analisar os formulários de verificação dos problemas do estudo de Lógica e o questionário sobre a experiência de uso do sistema, aplicados *online* aos alunos que já cursaram a disciplina e aos alunos que utilizaram o *software*, respectivamente, utilizamos técnicas de

análises estatísticas, disponíveis na ferramenta do *Google Forms*<sup>3</sup>. Quanto aos dados de pré teste e pós teste, estes foram tabulados na ferramenta Microsoft *Office Excel*®, gerando tabelas e gráficos para que pudéssemos ter uma visualização geral dos dados.

A análise do pré teste, pós teste e os *logs* de utilização em Banco de Dados, foram feitas de forma categorial, considerando que foi utilizado um instrumento predefinido pelas próprias regras de equivalência, sobre as quais foram analisadas as estatísticas dos erros e acertos. Com essas estatísticas, também foram criadas tabelas e gráficos para melhor visualização.

---

<sup>3</sup><https://docs.google.com/forms/u/0/>

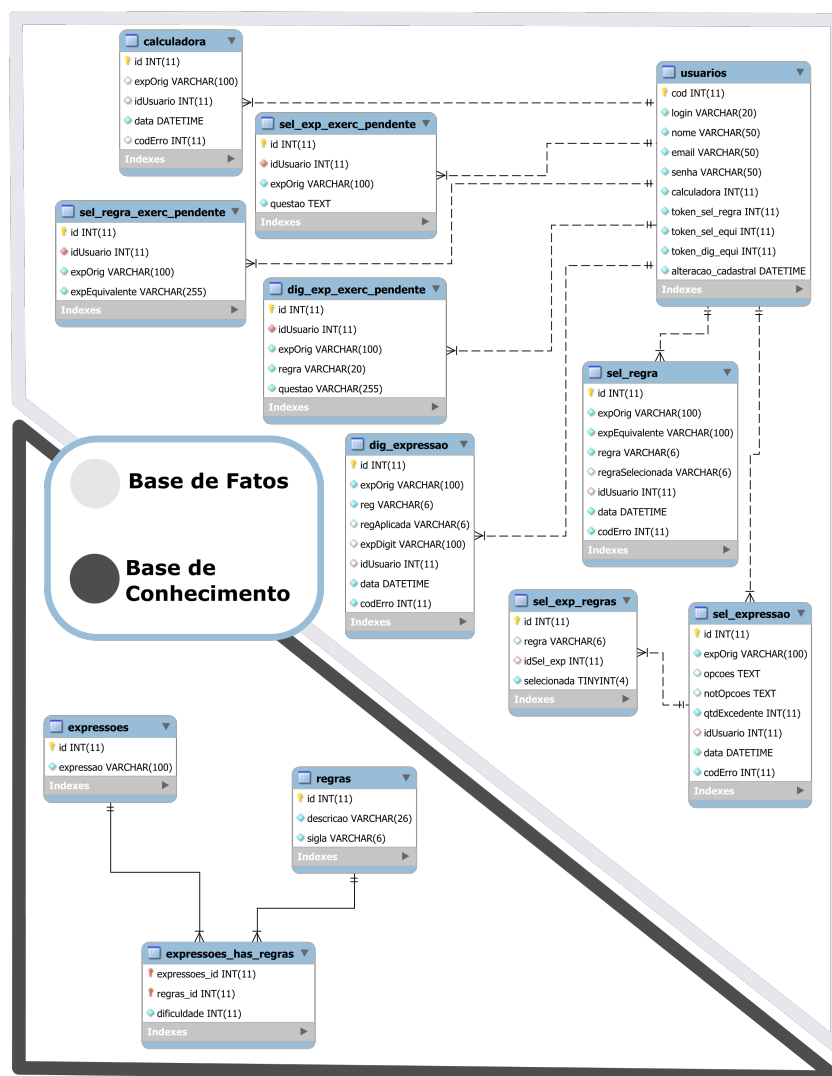
## 5 O SISTEMA STI - LÓGICA LIVRE

Neste capítulo, iremos discutir sobre as características funcionais e não funcionais do STI - Lógica Livre. Iniciaremos descrevendo a Base de Fatos e parte da Base de Conhecimento armazenadas em uma base de dados relacional. Na sequência, abordaremos a Arquitetura utilizada e seus módulos. Por fim, abordaremos as funcionalidades gerais do sistema.

### 5.1 BASE DE FATOS E BASE DE CONHECIMENTO

A Base de Fatos e Base de Conhecimentos do STI - Lógica Livre está armazenada em uma base de dados relacional, representada pelo Diagrama de Entidade e Relacionamento (DER) notação de Crow's Foot (ver na FIGURA 2).

FIGURA 2 – DIAGRAMA DE ENTIDADE E RELACIONAMENTO (DER) DO STI - LÓGICA LIVRE



FONTE: Imagem do autor (2018)



No banco de dados do STI - Logica Livre ficam gravados desde os dados de controle de acesso, até os dados gerados pelo aluno durante o uso do sistema (todos os seus acertos e erros, usos da calculadora, exercícios pendentes etc).

A base de fatos é basicamente a coleta de dados perante a utilização do *software*. Por isso, ela é formada por um conjunto de entidades relacionadas (direta ou indiretamente) ao usuário. Essa base será utilizada pelo módulo do aluno e módulo de domínio que serão explicados nas seções 5.2.1 e 5.2.2 respectivamente.

A entidade *usuario*<sup>1</sup> armazena os dados cadastrais do aluno bem como seus pontos e direito de acesso à ferramenta calculadora. A entidade *calculadora* armazena todas as expressões que o aluno digitou na ferramenta calculadora, por isso existe a relação com cardinalidade muitos-para-um<sup>2</sup> entre as entidades *calculadora* e *usuario* (sendo que um usuário pode ter muitas utilizações da calculadora).

As entidades *sel\_exp\_exerc\_pendente*, *sel\_regra\_exerc\_pendente* e *dig\_exp\_exerc\_pendente* armazenam os dados dos desafios que o aluno iniciou e ainda não terminou. Elas também contém cardinalidade muitos-para-um com *usuario* (um usuário para muitos desafios).

Análogo a isso, nas entidades *sel\_expressao*, *sel\_regra* e *dig\_expressao* são armazenadas as respostas do aluno face aos desafios propostos, ambas têm relações de cardinalidade muitos-para-um com a entidade *usuario* sendo que um aluno pode armazenar muitos desafios resolvidos. Para que a entidade *sel\_expressao* possa armazenar por completo os dados do desafio referente à seleção de expressões equivalentes, foi implementada a entidade *sel\_exp\_regras* com uma relação de cardinalidade muitos-para-um entre elas, uma vez que um desafio dessa modalidade pode conter muitas Regras abordadas, pois cada uma das expressões selecionadas, podem ser equivalentes através de uma Regra distinta.

A segunda parte do DER constitui a base de conhecimento do STI - Lógica Livre, ela é formada por três entidades: a entidade *expressoes* que armazena as Expressões Lógicas Proposicionais que serão utilizadas para elaboração dos desafios; a entidade *regras* que armazena as Regras de Equivalência Proposicional, que são a base para que o sistema tenha conhecimento de domínio; e por último, temos uma entidade denominada *expressoes\_has\_regras*, que tem o papel de relacionar as duas entidades anteriores com cardinalidade muitos-para-muitos, armazenando também, o nível de dificuldade que existe na aplicação da Regra para cada Expressão específica. Essa base será utilizada pelo módulo de domínio que será apresentada na seção 5.2.2.

---

<sup>1</sup>Considerando que o nome das entidades constituem-se linguagem SQL, desconsidera-se o uso de acentuação e caracteres especiais.

<sup>2</sup>É o número de elementos de um conjunto. Cada entidade que se relaciona com outra entidade possui uma relação um-para-um, um-para-muitos ou muitos-para-um (SILBERSCHATZ et al., 1997)

## 5.2 ARQUITETURA

O desenho da arquitetura do STI - Lógica Livre foi inspirado na Arquitetura Tradicional (referenciada na FIGURA 1, página 29) proposta por Carbonell (1970) e denominada Arquitetura Tripartida, na qual o sistema é dividido em, ao menos, três módulos: Aluno, Domínio e Tutor. A escolha dessa arquitetura se deu por ser simples e atender às necessidades da proposta. Ao analisar arquiteturas mais elaboradas, percebemos possíveis problemas tais como, a centralização da comunicação entre os módulos, conforme destacado na seção 3.7, que acontece na arquitetura proposta por Wenger (1988), ou então o isolamento entre os módulos Tutor e Domínio que acontece na arquitetura proposta por Kaplan e Rock (2011), que na nossa concepção, poderá trazer problemas no que se refere à tomada de decisão do Tutor.

Nas próximas seções, iremos discorrer sobre os módulos que compõem a arquitetura tradicional ou tripartida. Na seção 5.2.1 explicaremos sobre o Módulo do Aluno, que tem como função representar o conhecimento do aluno sobre determinado conteúdo. Na seção 5.2.2 iremos explicar sobre o Módulo de Domínio, que utiliza um Sistema Especialista (SE) que contém o conhecimento sobre o conteúdo. Na seção 5.2.3 falaremos sobre o Módulo Tutor, responsável pela tomada de decisão e ensino. E na seção 5.2.4, será explicado sobre o Módulo Interface, o qual realiza a interação com o aluno.

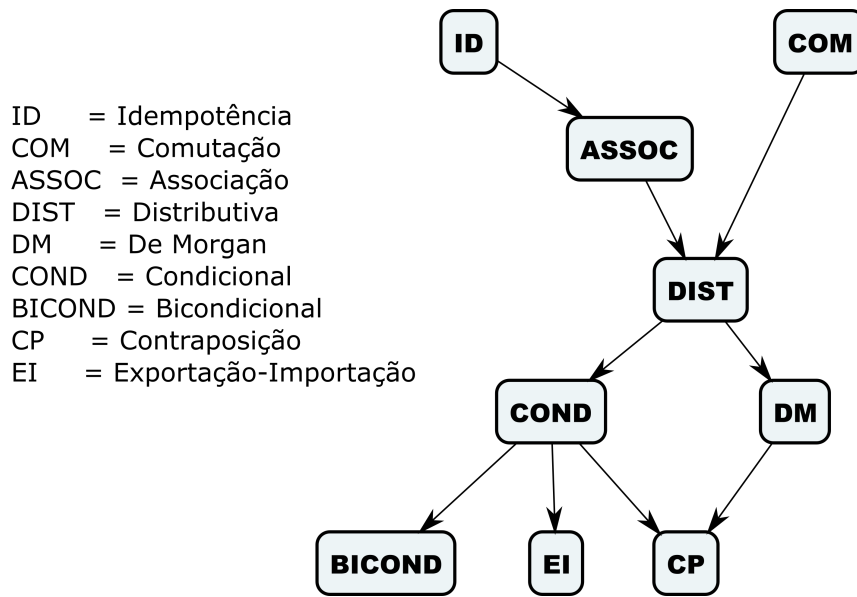
### 5.2.1 Módulo do Aluno

O Módulo do Aluno contém representações do conhecimento do aprendiz referente ao conteúdo (CARBONELL, 1970). Todas as ações do aluno são armazenadas em um sistema de banco de dados relacional MySQL (ver FIGURA 2 na página 39). Estes dados são chamados de Base de Fatos.

Os dados armazenados no banco de dados possibilitam simular a Zona de Desenvolvimento Real (ZDR) do aluno, ou seja, o conhecimento que ele domina (VYGOTSKY, 1978), pois, ao armazenar os dados sobre as respostas do aluno perante os desafios, o sistema realiza um diagnóstico dos mesmos, para avaliar e representar seu conhecimento neste conteúdo.

A FIGURA 3 ilustra a representação do conhecimento do aluno por meio de um grafo direcionado, no qual os nós são Regras de Equivalências Proposicionais e as arestas constituem os pré requisitos entre as Regras. O grafo contém todas as Regras a serem ensinadas pelo Tutor.

FIGURA 3 – GRAFO DE REPRESENTAÇÃO DO CONHECIMENTO



FONTE: Imagem do autor (2018)

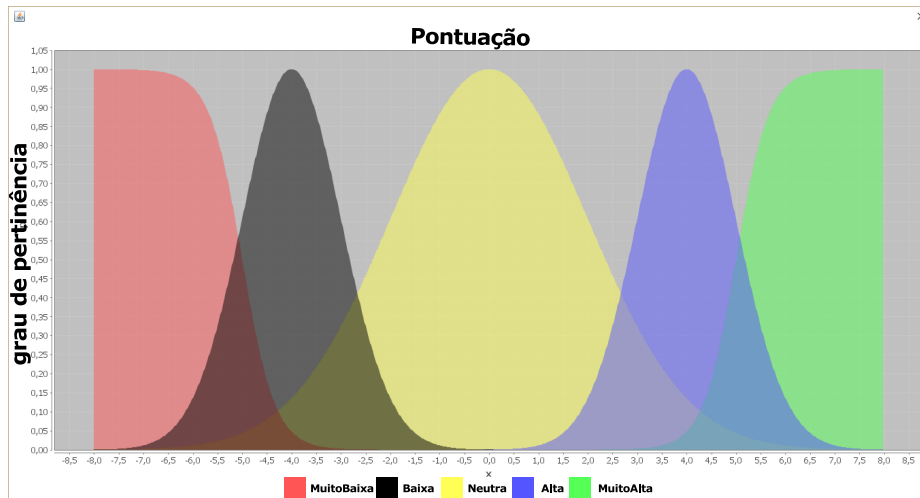
A cada nó do grafo (a cada regra) são atribuídos “pesos” os quais são calculados com base nos erros e acertos dos alunos durante o uso do sistema. Tais pesos são calculados por um SBC baseado em Lógica *Fuzzy*, que por sua vez recebe como entrada a quantidade de exercícios resolvidos e o índice de acerto com a Regra de Equivalência. O índice pode variar de -10 a 10, sendo que cada erro diminui pontos, cada acerto aumenta, podendo também ficar neutro caso o aluno não tenha errado a questão por completo. Quanto maior o número de exercícios que o aluno resolveu mais ênfase o sistema dá ao índice de acertos.

Na FIGURA 4, temos o comportamento do sistema *Fuzzy* frente ao primeiro parâmetro de entrada (o índice de acertos). Esta informação é mapeada por 5 funções de pertinência, 3 funções gaussianas<sup>3</sup> e 2 sigmóides<sup>4</sup> que determinam se é um índice Muito Baixo, Baixo, Neutro, Alto ou então Muito Alto, conforme a localização do número de entrada no eixo X. As funções sigmóides permitem caracterizar os conjuntos Muito Baixo e Muito Alto pois tendem ao infinito. As funções gaussianas permitem caracterizar os conjuntos Baixa, Neutra e Alta. A escolha do índice de pertinência para classificação dos conjuntos foi realizada empiricamente.

<sup>3</sup>Uma função do tipo gaussiana é simétrica em “forma de sino”.

<sup>4</sup>Uma função do tipo sigmoide é simétrica em “forma de S”, também conhecida como função logística.

FIGURA 4 – A PONTUAÇÃO E O ÍNDICE DE ACERTOS EM DETERMINADA REGRA

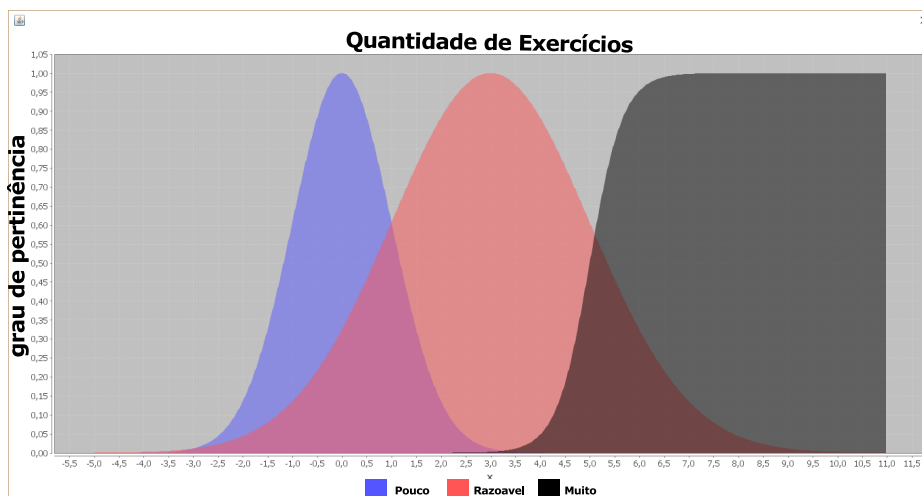


FONTE: Imagem do autor (2018)

Pela FIGURA 4, é possível perceber que a entrada para este cálculo é o índice de acertos, composto de um número decimal que pode variar de -10 a 10 (representado pelo eixo X), e o resultado é o grau de pertinência (localizado no eixo Y) de cada função *Fuzzy*, conforme a localização do índice no eixo X.

O segundo parâmetro de entrada para o sistema *Fuzzy*, é a quantidade de exercícios resolvidos pelo aluno em determinada Regra (ver FIGURA 5). Assim como o parâmetro anterior, este também é mapeado por funções gaussianas perfeitamente simétricas e uma sigmoide, dessa forma permitindo que o sistema infira se o aluno resolveu Pouco, Razoável ou Muitos exercícios.

FIGURA 5 – QUANTIDADE DE DESAFIOS RESOLVIDOS PARA DETERMINADA REGRA



FONTE: Imagem do autor (2018)

A forma como o sistema realiza o cálculo da quantidade de desafios na FIGURA 5 é análogo ao que foi explicado na FIGURA 4. Porém, aqui a entrada constitui-se da “qtd\_exerc” que pode existir entre zero e infinito, definindo o grau de pertinência das funções em Y. Ao mapear estas informações o sistema *Fuzzy* realiza um procedimento que é chamado *Fuzzyficação*<sup>5</sup>. Tal etapa permite ao sistema tomar decisões em ambientes de incerteza.

Após obter um valor *Fuzzy* para cada informação, o sistema inicia a fase de *Defuzzyficação*<sup>6</sup>, onde ele irá tomar a decisão referente ao peso que será retornado para determinada Regra Lógica Proposicional. A Tabela 1 apresenta as Regras de Produção<sup>7</sup> para tomada de decisão perante os resultados das funções anteriores (ver FIGURAS 4 da página 43 e 5 da 43).

TABELA 1 – REGRAS DE PRODUÇÃO PARA TOMADA DE DECISÃO

-	Quantidade de Exercícios	-	Pontuação	-	Peso
Se	Pouco	E	Muito Baixa	Logo	Baixo
Se	Pouco	E	Baixa	Logo	Neutro
Se	Pouco	E	Neutra	Logo	Neutro
Se	Pouco	E	Alta	Logo	Neutro
Se	Pouco	E	Muito Alta	Logo	Médio
Se	Razoável	E	Muito Baixa	Logo	Baixo
Se	Razoável	E	Baixa	Logo	Baixo
Se	Razoável	E	Neutra	Logo	Neutro
Se	Razoável	E	Alta	Logo	Médio
Se	Razoável	E	Muito Alta	Logo	Médio
Se	Muito	E	Muito Baixa	Logo	Muito Baixo
Se	Muito	E	Baixa	Logo	Baixo
Se	Muito	E	Neutra	Logo	Neutro
Se	Muito	E	Alta	Logo	Alto
Se	Muito	E	Muito Alta	Logo	Alto

FONTE: Dados do autor (2018)

É por meio de Regras de Produção que o sistema *Defuzzyfica* as informações para cada opção *Fuzzyficada*. Como exemplo, citamos a Regra de Produção contida na quinta linha da TABELA 1, acima representada: se a “qtd\_exerc” é pouco, mesmo que a pontuação seja Muito Alta, o sistema ainda atribui um peso Médio. Porém conforme a “qtd\_exerc” aumenta o sistema dá maior ênfase na pontuação para atribuir o peso. Como exemplo, temos a última linha da TABELA 1.

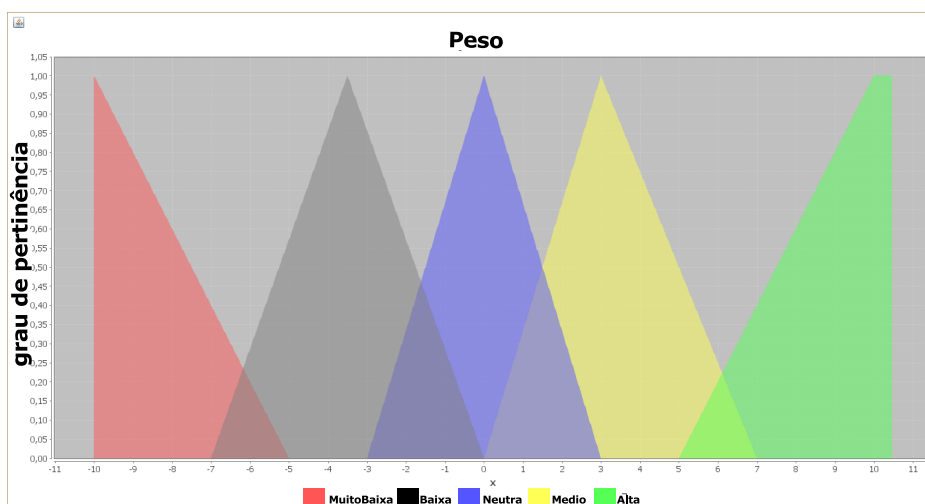
O gráfico de *Defuzzyficação*, representa o cálculo do peso de cada conteúdo no Módulo do Aluno (ver FIGURA 6). Este peso será retornado ao sistema, que o utilizará como processo avaliativo.

<sup>5</sup>Classificar os dados de entrada para conjuntos *Fuzzy*.

<sup>6</sup>Mapear os conjuntos *Fuzzy* para dados que serão retornados

<sup>7</sup>Regras denominadas por Se, Condição e Então.

FIGURA 6 – PESO RETORNADO PELO SISTEMA FUZZY



FONTE: Imagem do autor (2018)

De acordo com a FIGURA 6, assim que o sistema *Fuzzy* calcular todas as regras, teremos um grafo (no mesmo formato do grafo apresentado na FIGURA 3 da página 42) com pesos decimais em cada nó que, por sua vez, constitui a representação do conhecimento do aluno.

O Módulo do Aluno calcula também o relatório de desempenho do aluno, atualizando a interface em tempo de execução, ou seja, a cada exercício resolvido, o relatório é atualizado e disponibilizado ao aluno sempre que ele desejar.

### 5.2.2 Módulo de Domínio

O Módulo de Domínio apresenta o conhecimento do STI acerca dos conhecimentos a serem ensinados ao longo do curso (CARBONELL, 1970), no caso específico deste trabalho, os conhecimentos sobre Equivalência Lógica Proposicional. Este módulo contém uma base de conhecimento a respeito dos conteúdos abordados e um motor para resolução de exercícios, tornando-o assim um Sistema Especialista (SE)<sup>8</sup> em Equivalência Lógica Proposicional.

O SE recebe como entrada um conjunto de caracteres que supostamente é uma Expressão Lógica Proposicional. Após isso, ele faz as validações necessárias para garantir a correteza dessa Expressão e descobrir outras equivalentes à ela. Este processo é dividido em 4 etapas: Análise Léxica; Análise Sintática; Análise Semântica (análises geralmente utilizadas em compiladores) e; a última etapa, o Gerador de Equivalências.

As duas primeiras etapas (Análise Léxica e Sintática) são implementadas em uma *framework* chamada JavaCC<sup>9</sup>, onde criamos uma gramática de autoria própria que valida a correteza de uma Expressão Lógica Proposicional:

<sup>8</sup>Sistema que contém o domínio do conhecimento em um assunto específico.

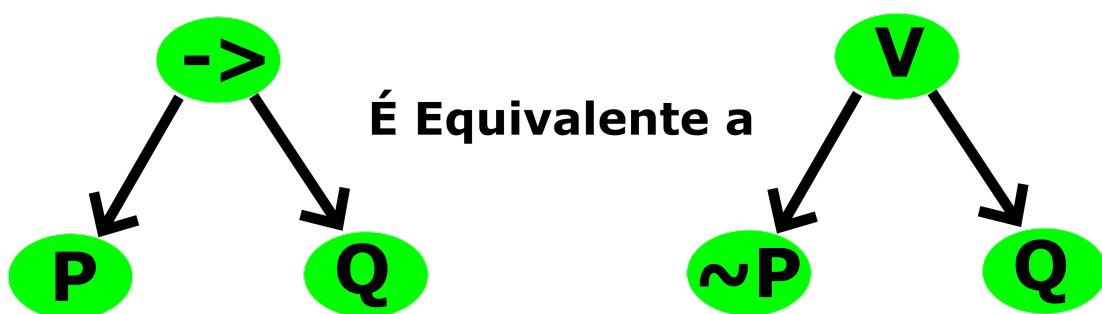
<sup>9</sup>(Java Compiler Compiler) é um gerador de analisador sintático e léxico de código aberto para a linguagem Java.

- Uma “Expressão” é apenas um “Termo”, ou um “Termo” seguido de um “Complemento de Expressão”.
- Um “Complemento de Expressão” é um “Conectivo Binário” seguido de uma “Expressão”.
- Um “Termo” é apenas um “Termo Simples”, ou um “Conectivo Unário” seguido de um “Termo Simples”.
- Um “Termo Simples” é apenas uma “Proposição”, ou então uma “Expressão” entre parênteses.
- Um “Conectivo Binário” pode ser [E (  $\wedge$  ), OU (  $\vee$  ), Bi Implcação (  $\leftrightarrow$  ) ou Implcação (  $\rightarrow$  )].
- Um “Conectivo Unário” pode ser apenas uma Negação (  $\neg$  )
- Uma “Proposição” pode ser qualquer conjunto de caracteres que comece com letras de A a Z, não contenha espaço e nem “Conectivos”.

O sistema retorna então, uma lista de *Tokens*, que consiste na Expressão Lógica quebrada em partes atômicas<sup>10</sup>, ou então retorna os detalhes do erro, caso a expressão não esteja correta. Na terceira e quarta etapa, o SE realiza a Análise Semântica e a Geração das Equivalências. Após ter recebido uma Lista de *Tokens*, o SE insere, um a um, em uma estrutura de dados do tipo árvore binária, respeitando assim os parênteses e precedência dos Conectivos. Na sequência o SE analisa recursivamente a árvore, de modo que gera uma lista de árvores equivalentes a esta, aplicando as Regras Proposicionais cabíveis.

A aplicação das Regras de Equivalência na estrutura de árvore binária é realizada pela alteração do conectivo da raiz e dos seus descendentes imediatos. O resultado é uma árvore com o conteúdo semântico equivalente ao da árvore original, conforme ilustra a FIGURA 7.

FIGURA 7 – EQUIVALÊNCIA LÓGICA PROPOSICIONAL



FONTE: Imagem do autor (2018)

Após aplicar todas as regras possíveis e gerar todas as equivalências, cada equivalência representada por uma estrutura de árvore (como na FIGURA 7), as árvores são inseridas em uma estrutura do tipo lista, obtendo-se, uma lista de equivalências.

O Módulo de Domínio contém também uma base de dados pré carregada com 42 Expressões Lógicas válidas, para que ele possa utilizar-se delas quando o tutor solicitar conteúdos

<sup>10</sup>Partes de uma expressão que já foi dividida o tanto quanto possível

com intuito de formular exercícios. O SE é capaz de calcular quais regras se aplicam em quais expressões, e também o nível de dificuldade de cada aplicação de regra. O nível de dificuldade é um inteiro de 0 a 100, onde 0 é muito fácil e 100 muito difícil.

### 5.2.3 Módulo Tutor

O grafo de representação do conhecimento (ilustrado na FIGURA 3 da página 3), que compõe o Módulo do Aluno, é reutilizado pelo Módulo Tutor, de modo que ele analisa as Regras que o aluno tem domínio (Zona de Desenvolvimento Real - ZDR), para calcular as Regras posteriores do grafo, ou seja, o que o aluno está apto a aprender. Isso possibilita que o Tutor trabalhe em sua Zona de Desenvolvimento Proximal (ZDP), que consiste em conteúdos que ele está apto a aprender de imediato (VICCARI, 1990). Para prover um novo desafio, o tutor busca por expressões nas quais é possível aplicar as Regras encontradas.

Para que o Tutor tome a decisão correta de qual exercício prover ao aluno, além de levar em conta as regras calculadas anteriormente, ele também utiliza-se do nível de dificuldade em que o aluno se encontra em uma determinada Regra. Ele busca o desafio mais difícil que o aluno já respondeu corretamente e incrementa a dificuldade (escolhendo a próxima expressão em nível de dificuldade). Dessa forma, além de o aluno progredir nas Regras ele também pode progredir em nível de dificuldade. Caso ele cometa muitos erros posteriormente e o Tutor tome a decisão de regredir o grafo da Representação do Conhecimento do aluno, fazendo com que retorne nos conteúdos, o aluno será submetido a expressões tão difíceis quanto antes, porém em regras mais fáceis. Esta estratégia foi pensada para que o aluno não precise “reiniciar” o Tutor, e também para que não se repitam muitos exercícios.

Além de construir o desafio em tempo de execução, levando em conta a individualidade do aluno, o Módulo Tutor também é responsável por prover o *feedback* face ao desafio resolvido. Para a realização dessa tarefa, o Módulo Tutor se comunica com o Módulo de Domínio, que é responsável pelas análises e retorno das equivalências. O Módulo Tutor utiliza-se do conhecimento do Módulo Domínio para avaliar a resposta, montando assim um *feedback* contextualizado.

Caso o aluno erre o desafio, o Tutor busca fazer inferência do raciocínio utilizado pelo aluno, por exemplo: o Tutor apresentou uma Expressão e pediu para o aluno aplicar determinada Regra, gerando assim uma Expressão equivalente. Porém, o aluno aplicou outra Regra e gerou uma Expressão Equivalente, ou seja, a Expressão digitada não é uma expressão esperada. Neste caso, o Tutor retorna em seu *feedback* a informação que a resposta não está completamente correta, mostra como seria uma resposta correta e mostra também a Regra que foi aplicada erroneamente. A cada desafio realizado pelo aluno, o Módulo Tutor atualiza as informações do Módulo Aluno com seus resultados. Dessa forma, o Tutor já poderá contar com estes dados para tomar as decisões dos próximos desafios.



#### 5.2.4 Módulo de Interface

O sistema foi desenvolvido com uma interface *Web* responsiva, que permite acesso direto, sem instalação de componentes, podendo ser acessado de qualquer lugar com acesso à *Internet*, sendo assim, compatível com qualquer plataforma de sistema operacional.

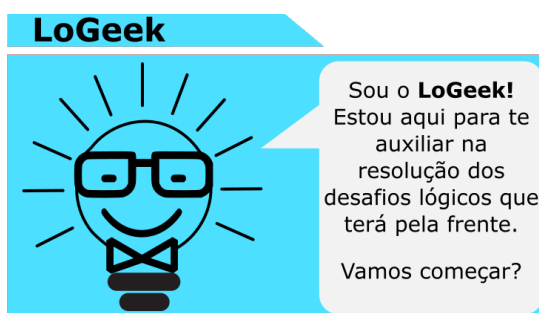
Após fazer a autenticação do usuário no sistema, todas as ferramentas do tutor estão disponíveis em uma única tela por meio de navegação em abas. As abas servem para alterar entre os diferentes tipos de desafios (iremos discorrer sobre eles na seção 5.3). São utilizados modais<sup>11</sup> para: menus de ajuda; acesso à calculadora; e o relatório de desempenho. O motivo de se utilizar essa forma de visualização para estas ferramentas, é o fato do usuário conseguir utilizar determinada funcionalidade sem sair do exercício que está resolvendo, ou seja, ele conta com orientações e dicas, pode fazer uso da calculadora e ainda consultar seu desempenho a qualquer momento.

### 5.3 FUNCIONALIDADES

O STI desenvolvido neste trabalho aborda justamente os conteúdos iniciais e fundamentais do ensino de Lógica, como um aporte tecnológico aos alunos. Conta também com um *feedback* personalizado, fazendo uma análise semântica da resposta do aluno, mostrando outras respostas possíveis, e interpretando a linha de raciocínio que ele(a) desenvolveu. Até em situações em que o aluno não responde corretamente, o STI - Lógica Livre procura um modo de relacionar a resposta com um dos resultados corretos, para verificar qual o equívoco do estudante, provendo *feedback* que são também dicas para exercícios posteriores.

Os exercícios disponibilizados ao aluno são chamados de desafios, e o aluno é incentivado a resolvê-los por meio de artifícios que consistem em: desbloquear recursos do sistema (como por exemplo a Calculadora Lógica, que iremos discorrer ainda nesta seção); incrementar o seu relatório de desempenho; ou até mesmo pela interação com o personagem chamado LoGeek que é mostrado na FIGURA 8.

FIGURA 8 – PERSONAGEM LOGEEK

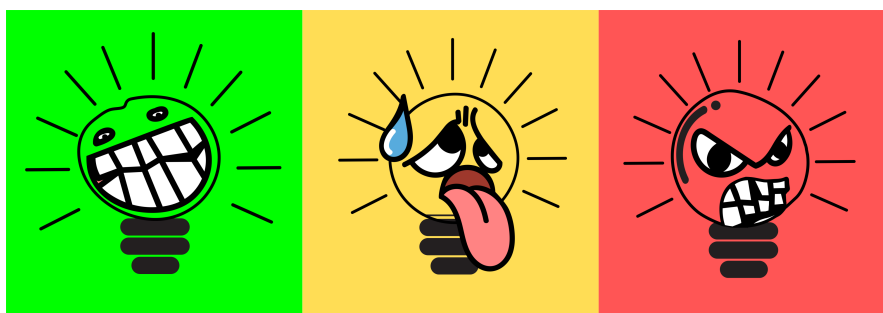


FONTE: Imagem do autor (2018)

<sup>11</sup> Janela secundária, que sobrepõe a janela principal até que se faça determinada ação.

O personagem LoGeek foi desenvolvido para que o Tutor se mostre mais próximo ao aluno, por isso o LoGeek o acompanha durante o uso do sistema. Ele pode se apresentar de várias formas demonstrando emoções face aos erros e acertos do aluno na resolução dos desafios. Quando o aluno acerta ele é apresentado com semblante feliz, análogo a isso, aparenta tristeza com o erro, ou então bravo (conforme a relevância do erro), como ilustra na FIGURA 9.

FIGURA 9 – DIFERENTES FORMAS DE APRESENTAÇÃO DO LOGEEK



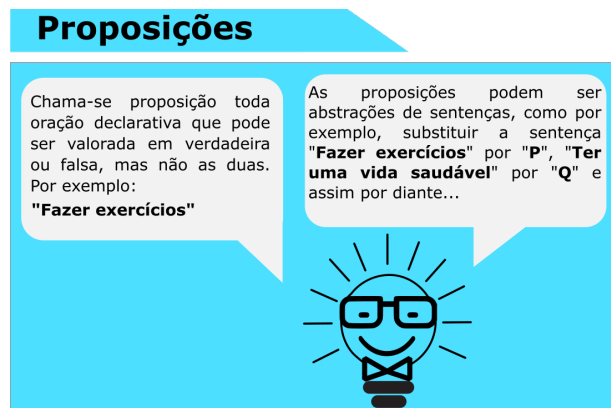
FONTE: Imagem do autor (2018)

O LoGeek está presente em todos os *feedbacks* dados ao aluno. Este personagem foi criado para auxiliar no processo da Transposição Didática, que segundo Chevallard (1991), é a ação de submeter o conteúdo formal a uma série de adaptações tornando-o apto ao processo de ensino e aprendizagem, sem perder seu rigor científico. Outro fator que torna o LoGeek uma característica promissora do sistema é a realidade tecnológica dos alunos da geração atual, pois segundo Veen e Vrakking (2009), os jovens apresentam cada vez mais dificuldade em manter a atenção em uma só atividade por períodos prolongados, pensamos que o LoGeek possa ser um recurso visual para atrair a atenção do aluno ao sistema, sendo também um reforço positivo (behaviorismo).

Para que não se perca a formalidade de termos técnicos do conteúdo de lógica, O STI - Lógica Livre utiliza, inclusive, orientações elementares sobre os Termos e as Regras essenciais ao conteúdo, exemplificando conteúdos abstratos de maneira visual. Essas orientações são de uso ilimitado e ficam acessíveis ao aluno durante toda sua interação.

Algumas dessas orientações são disponibilizadas na aba inicial, por se tratarem de termos técnicos essenciais para o entendimento do conteúdo. Na FIGURA 10 temos a explicação sobre o termo proposição, que está presente (explícita ou implicitamente) em todos os desafios do sistema.

FIGURA 10 – EXPLICAÇÃO INICIAL SOBRE PROPOSIÇÕES



FONTE: Imagem do autor (2018)

A explicação do Tutor sobre Proposições é simples e direta, mostrando que proposições podem ser abstrações de sentenças, sendo possível substituí-las por símbolos proposicionais.

Os conectivos lógicos que vemos na FIGURA 11, também são apresentados ao aluno na aba inicial do sistema, uma vez que são essenciais para prosseguir com o aprendizado do conteúdo.

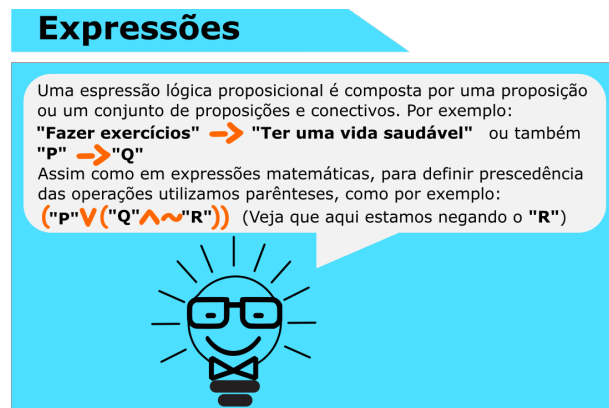
FIGURA 11 – EXPLICAÇÃO INICIAL SOBRE CONECTIVOS



FONTE: Imagem do autor (2018)

O Tutor explica como são formadas as expressões e contextualiza fazendo substituições das proposições por símbolos proposicionais facilmente conhecidas no cotidiano, como observado na FIGURA 12.

FIGURA 12 – EXPLICAÇÃO INICIAL SOBRE EXPRESSÕES

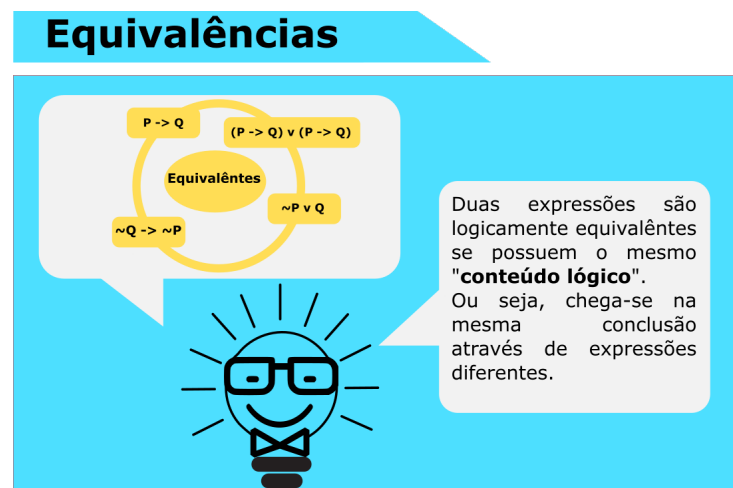


FONTE: Imagem do autor (2018)

De acordo com a FIGURA 12, por meio dessa prática, o Tutor está facilitando o processo de aprendizagem significativa ao mesmo tempo que realiza uma transformação adaptativa, para contribuir com o processo de transposição didática.

Ao explicar sobre Equivalência Lógica Proposicional, o tutor faz uso de todos os conceitos vistos anteriormente. Apesar de serem conteúdos básicos, são elementares para prosseguir com o aprendizado que o Tutor irá mediar posteriormente (ver FIGURA 13).

FIGURA 13 – EXPLICAÇÃO INICIAL SOBRE EQUIVALÊNCIA



FONTE: Imagem do autor (2018)

Após ver os conteúdos iniciais, o aluno estará apto a resolver os desafios propostos pelo Tutor. Esta primeira versão do sistema possibilita que o aluno escolha dentre três modalidades de desafios lógicos, os quais são: Selecionar Regra, Selecionar Equivalência e Digitar Equivalências.

A FIGURA 14 ilustra a modalidade Selecionar Regra. Nesta modalidade o Tutor apresenta duas expressões equivalentes (a original e a equivalente) e dez Regras de Equivalência Lógica Proposicional.

FIGURA 14 – DESAFIO DE SELECIONAR REGRA

No desafio "**Selecionar Regras**" são dadas duas expressões equivalentes e devemos responder qual **Regra** as tornam equivalentes

FONTE: Imagem do autor (2018)

Ao apresentar a expressão, o tutor solicita ao aluno que selecione a regra que as tornam equivalentes. Depois de selecionado, o Tutor retorna o *feedback*. Caso tenha acertado o aluno é parabenizado, caso não, o Tutor mostra a resposta correta e onde ele errou. Mas caso o aluno tenha selecionado uma Regra que é aplicável à expressão original, mas não seja a correta, o Tutor apresenta a Regra correta, mostra a(s) expressão(ões) equivalente(s) através da Regra selecionada e diz que a resposta não está completamente correta. Esta prática utilizada pelo Tutor é uma maneira de levar em conta o raciocínio que o aluno desenvolveu ao resolver o desafio, mesmo que o resultado esteja errado.

A modalidade de desafio Selecionar Equivalência é ilustrada na FIGURA 15.

FIGURA 15 – DESAFIO DE SELECIONAR EQUIVALÊNCIAS

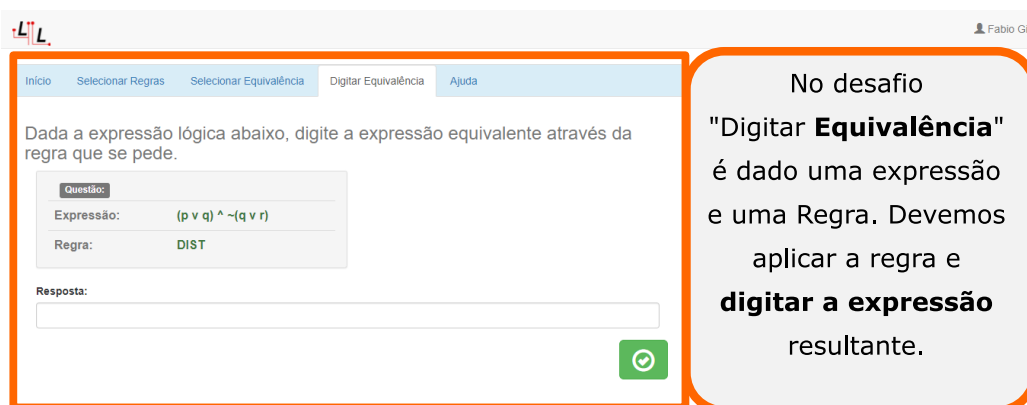
No desafio "**Selecionar Equivalência**" é dado uma expressão principal e várias outras equivalentes ou não a ela. Devemos selecionar apenas **expressões equivalentes**.

FONTE: Imagem do autor (2018)

Nesta modalidade, o Tutor apresenta uma expressão original ao aluno e outras expressões (algumas equivalentes e outras não), solicitando que ele escolha todas as expressões que são equivalentes à original. Após o aluno ter selecionado e enviado a resposta, o Tutor retorna o *feedback* mostrando as respostas corretas, os erros e acertos do aluno. Percebe-se que aqui a resposta pode estar correta, pode não estar completamente correta, ou pode estar errada, pois o aluno pode selecionar expressões indevidas como também deixar de selecionar expressões equivalentes. Em qualquer um dos cenários, o Tutor analisa e retorna todos os erros e acertos, ou seja, o *feedback* mostra passo a passo a resolução do exercício, contextualizando com a resposta do aluno.

Na modalidade Digitar Equivalência, o tutor apresenta uma expressão e uma regra, para que o aluno aplique a regra sobre a expressão e digite uma outra expressão, que seja equivalente à primeira (ver FIGURA 16).

FIGURA 16 – DESAFIO DE DIGITAR EQUIVALÊNCIA

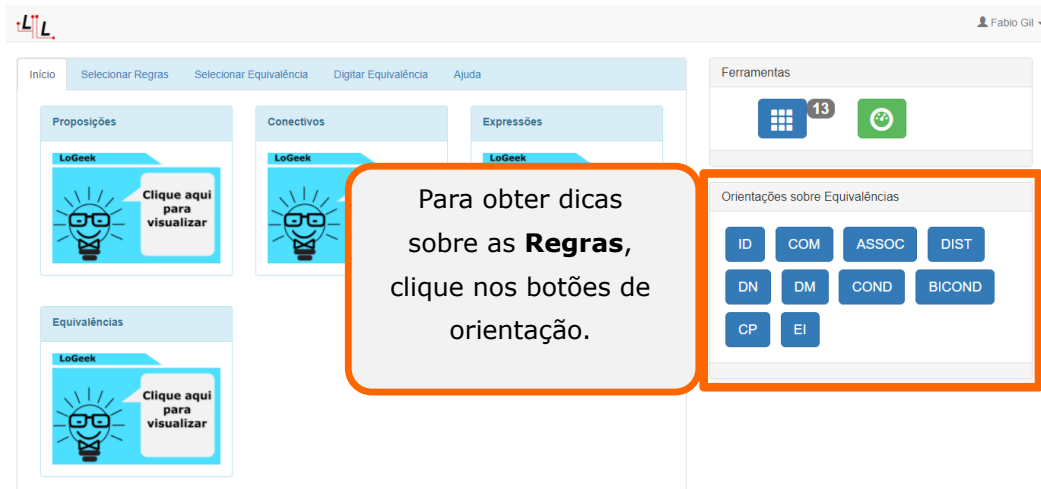


FONTE: Imagem do autor (2018)

Vale lembrar que podem existir várias respostas corretas para o desafio Digitar Equivalência, ao avaliar a resposta do aluno o tutor: (i) pode parabenizá-lo - caso esta seja uma das corretas; (ii) pode informar que está errada - caso não seja equivalente a expressão inicialmente apresentada; e (iii) pode informar que a resposta está parcialmente correta - caso a expressão digitada seja equivalente através de outra regra, diferente da solicitada. Neste último caso, o Tutor informa qual Regra o aluno utilizou no lugar da que seria correta. Por fim, independente da avaliação, o tutor retorna ao aluno todas as expressões que seriam corretas. Este é um outro exemplo de *feedback* personalizado, pois o Tutor, novamente, leva em conta o fato de que o aluno utilizou-se de um raciocínio para esta resolução.

Além de prover *feedback* personalizado, o STI - Lógica Livre também dispõe de dicas para auxiliar o aluno antes mesmo da resolução dos desafios. Em um painel localizado na região direita da tela, ficam disponíveis orientações sobre todas as Regras que o Tutor utiliza durante os desafios. Estas dicas são de uso livre e podem ser utilizadas a qualquer momento, pois sempre estão acessíveis na tela do usuário, como podemos ver na FIGURA 17.

FIGURA 17 – DICAS SOBRE AS REGRAS



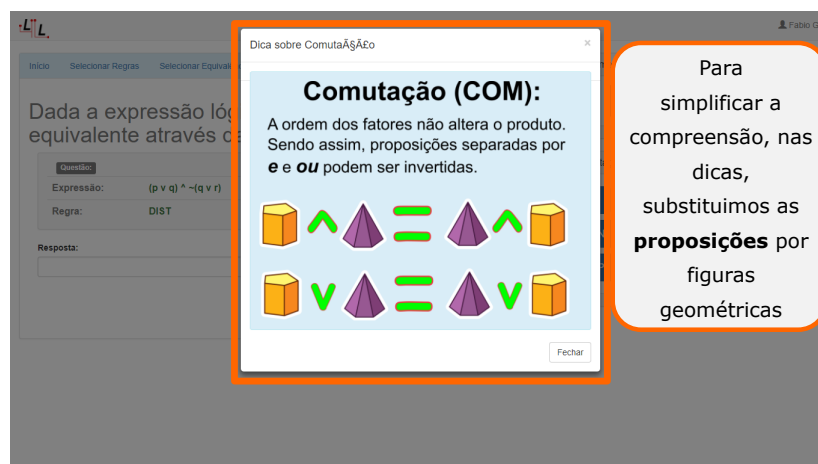
FONTE: Imagem do autor (2018)

Ao clicar no botão, a dica é disponibilizada em um modal, permitindo que o aluno utilize durante a resolução dos desafios, sem mesmo precisar sair da aba.

O motivo que nos levou a colocar o painel de dicas em uma região exclusiva da tela, tirando a necessidade de sair do desafio para realizar a consulta, é a perda de atenção ao realizar tal procedimento. Segundo Veen e Vrakking (2009) os jovens da geração atual tendem a *zapear*<sup>12</sup> durante o uso de computadores e outros dispositivos ligados a *Internet*. Desta forma, para evitarmos que os alunos percam o foco no conteúdo, deixamos à sua disposição tudo o que é necessário para resolução do desafio.

A FIGURA 18 ilustra a forma como o tutor apresenta as dicas sobre Regras Lógicas Proposicionais.

FIGURA 18 – DICA COM REGISTRO SEMIÓTICO



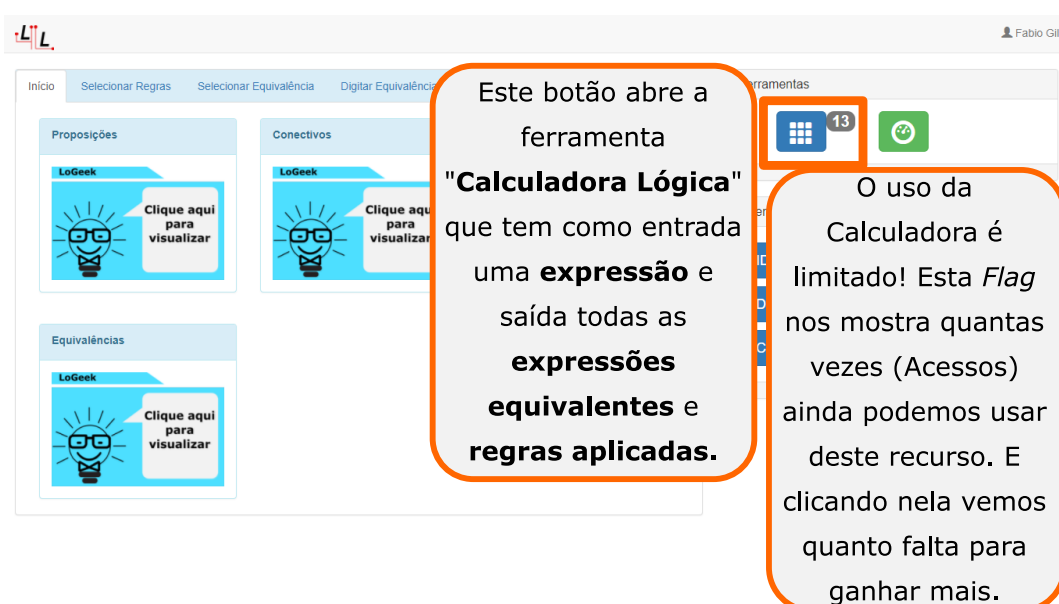
FONTE: Imagem do autor (2018)

<sup>12</sup> *Zapear* remete a um ato de indecisão, onde a pessoa muda rápida e repetidamente para encontrar algo que agrade, como alterar canais muitas vezes em um aparelho de televisão.

Para que o STI - Lógica Livre pudesse contemplar esta característica, substituímos as proposições por figuras geométricas coloridas (com cores contrastantes) como está ilustrado na FIGURA 18, para que o aluno tenha uma forma visual, que o possibilite compreender o conteúdo abstrato sem perder o rigor científico, contribuindo também com a prática da Transposição Didática (CHEVALLARD, 1991).

Outro diferencial do STI - Lógica Livre em relação a outros sistemas do mesmo segmento, é a existência de uma ferramenta de auxílio denominada Calculadora Lógica, que pode resolver qualquer desafio proposto pelo Tutor (ver FIGURA 19).

FIGURA 19 – CALCULADORA LÓGICA



FONTE: Imagem do autor (2018)

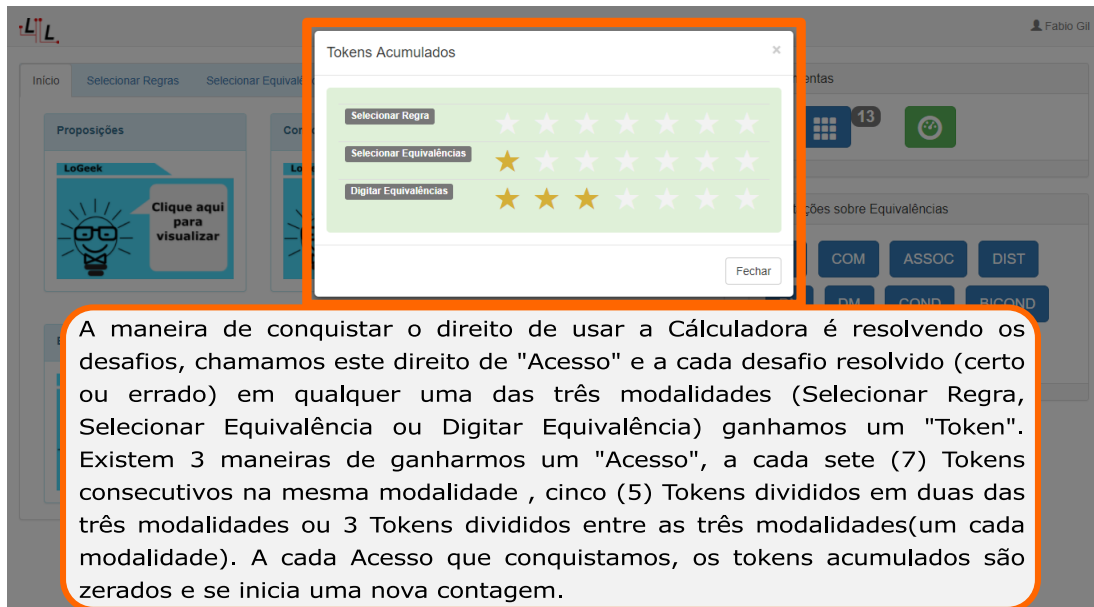
A Calculadora Lógica fica disponível ao aluno, porém constitui-se de um recurso limitado, de modo que sua utilização é liberada na medida em que o aluno resolve os desafios. O motivo principal pelo qual mantivemos o uso do recurso limitado ao aluno é o cuidado para que o mesmo não se torne dependente, uma vez que ela retorna as Equivalências e as Regras para qualquer Expressão Lógica Proposicional.

Assim como as dicas, a Calculadora Lógica é um recurso para ser utilizado durante a resolução dos desafios, por este motivo o botão de acesso à ela encontra-se na região superior direita, acima do painel de orientações sobre as Regras, como mostra a FIGURA 19. Sua utilização também é feita por meio de uma janela modal para que o aluno não necessite sair da aba do desafio.

Outro motivo que nos levou a restringir o uso da Calculadora é a necessidade de criar um fator motivacional para resolução de desafios. Sendo assim, utilizamos disso como um artifício, de modo que o aluno aumente seu limite de acessos conforme resolve os próprios desafios. Criamos, então, um protocolo específico para liberação desse recurso, ilustrado na FIGURA 20.



FIGURA 20 – TOKENS - LIBERAR USO DA CALCULADORA

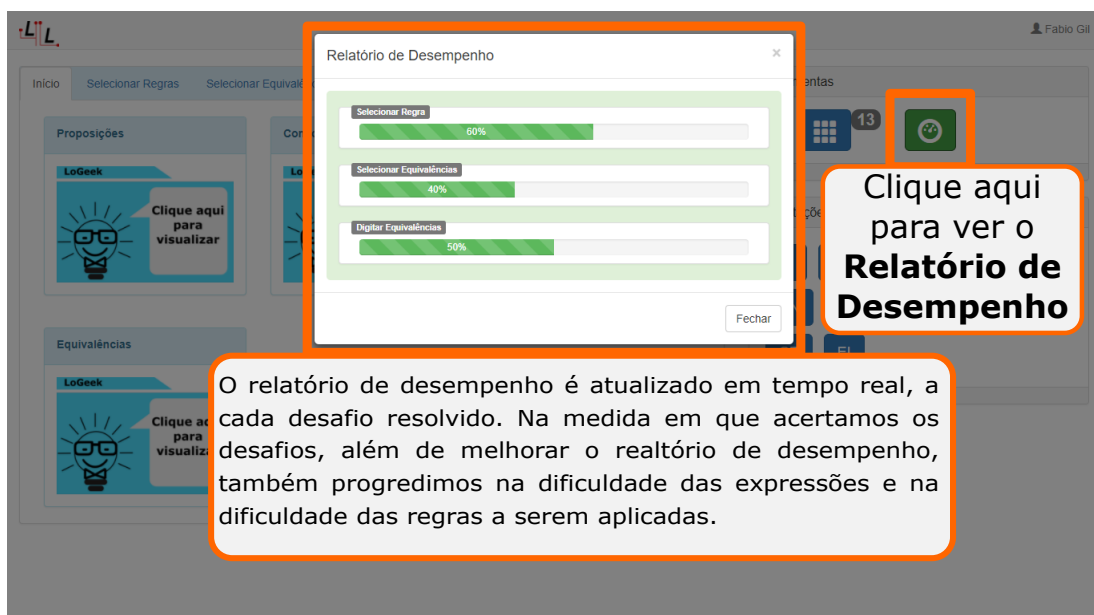


A maneira de conquistar o direito de usar a Calculadora é resolvendo os desafios, chamamos este direito de "Acesso" e a cada desafio resolvido (certo ou errado) em qualquer uma das três modalidades (Selecionar Regra, Selecionar Equivalência ou Digitar Equivalência) ganhamos um "Token". Existem 3 maneiras de ganharmos um "Acesso", a cada sete (7) Tokens consecutivos na mesma modalidade, cinco (5) Tokens divididos em duas das três modalidades ou 3 Tokens divididos entre as três modalidades (um cada modalidade). A cada Acesso que conquistamos, os tokens acumulados são zerados e se inicia uma nova contagem.

FONTE: Imagem do autor (2018)

Conforme observado na FIGURA 20, na região superior direita, ao lado do botão da Calculadora, se encontra o botão de acesso para o Relatório de Desempenho, onde o aluno pode acompanhar seu progresso em cada uma das modalidades. Esta informação é ilustrada na FIGURA 21 que segue.

FIGURA 21 – RELATÓRIO DE DESEMPENHO DO ALUNO



O relatório de desempenho é atualizado em tempo real, a cada desafio resolvido. Na medida em que acertamos os desafios, além de melhorar o relatório de desempenho, também progredimos na dificuldade das expressões e na dificuldade das regras a serem aplicadas.

FONTE: Imagem do autor (2018)

O Relatório de desempenho é outro recurso que possui relevância devido ao fato de o aluno poder consultar com facilidade, em que pontos ele pode melhorar. Além disso, o fato

de que é possível visualizar uma melhora no índice do aluno à medida que ele(a) resolver os desafios, pode ser outro fator motivacional para que se dedique na resolução. Por esse motivo, pensamos ser necessário deixá-lo em uma região exclusiva da tela, possibilitando sua visualização a qualquer momento.

## 6 ANÁLISE E DISCUSSÕES

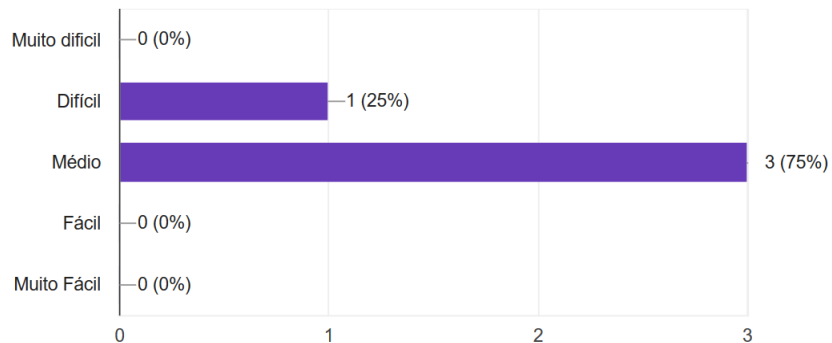
A presente seção contém os resultados obtidos pelos experimentos realizados durante o trabalho: o primeiro experimento consistiu-se na verificação dos problemas de ensino e aprendizagem de Lógica, mediante a aplicação de um questionário; no segundo experimento aplicamos um questionário direcionado à especialistas das áreas de Computação, Educação e Matemática, com o intuito de validar o *software* como ferramenta educacional e obter sugestões de melhoria; o terceiro experimento foi realizado com alunos de Graduação que estão cursando a disciplina de Lógica e consistiu na aplicação de um pré teste, que teve como objetivo avaliar os conhecimentos prévios referente ao conteúdo do estudo; o quarto experimento foi a utilização do STI - Lógica Livre por um grupo de alunos escolhidos aleatoriamente dentre a turma em que foi aplicado o pré teste; no quarto experimento aplicamos outro teste (denominado pós teste), direcionado aos mesmos alunos do pré teste, para uma nova avaliação de conhecimento após a utilização do sistema; e por último, para avaliar a experiência de utilização do sistema, aplicamos um questionário aos alunos que utilizaram o *software*. Nos próximos parágrafos, iremos apresentar e discutir sobre os dados obtidos com os instrumentos citados acima.

Com intuito de analisar os problemas do estudo de Lógica do curso de Licenciatura em Computação da UFPR - Setor Palotina, aplicamos um questionário *online* com duas questões de múltipla escolha, direcionado aos alunos que já cursaram a disciplina de Lógica. O questionário foi respondido por 4 alunos e os resultados serão apresentados a seguir. A primeira questão, ilustrada pela FIGURA 22, é referente ao ensino de Lógica em geral, questionando o nível de dificuldade que a disciplina representa para os alunos.

FIGURA 22 – NÍVEL DE DIFICULDADE DA DISCIPLINA DE LÓGICA

Na sua percepção, qual nível de dificuldade da disciplina de Lógica?

4 responses



FONTE: Imagem do autor (2018)

Dentre os quatro alunos que responderam o questionário, 3 (75%) deles consideram o nível de dificuldade da disciplina de Lógica como médio, enquanto 1 (25%) a considera difícil (ver na FIGURA 22).

A segunda questão foi direcionada ao ensino do conteúdo de Equivalência Lógica Proposicional, sobre a possível necessidade do auxílio de ferramentas para um melhor entendimento. Todos os alunos questionados pensam ser importante e que ajudaria muito, a existência de uma ferramenta para auxiliar no entendimento do conteúdo.

Os dados apresentados sustentam a necessidade da criação de um *software* é, sim, relevante para nosso contexto, como também imaginamos que seja relevante à outras instituições que contém cursos na mesma área. Esses resultados não só nos motivam a prosseguir com a pesquisa, como também, valida o primeiro princípio geral para Engenharia de *Software* proposto por Hooker (1996 apud PRESSMAN, 2005), o qual afirma que todo sistema precisa ter uma razão de existir.

Com isso, o *software* STI - Lógica Livre foi desenvolvido e avaliado por especialistas das áreas de Computação, Educação e Matemática, com intuito não só de validar o mesmo como ferramenta educacional, como também de obter *feedbacks* para melhorias contínuas no projeto. Os especialistas escolhidos foram professores da UFPR - Setor Palotina, de modo que pudessem avaliar conceitos técnicos do desenvolvimento, aplicabilidade e adaptabilidade para o ensino, além da coerência sobre os conteúdos abordados.

Os questionários continham questões que podiam ser respondidas atribuindo relevâncias segundo uma escala que varia de 1 (um) à 4 (quatro), sendo 1 (um) ruim, 2 (dois) suficiente, 3 (três) bom e 4 (quatro) excelente, além da opção NA (não se aplica) para cada questão.

As questões elaboradas eram específicas para a área de cada avaliador. Para facilitar a observação, as questões receberam rótulos que foram constituídos por um prefixo seguido de um número. Os prefixos são relacionados ao nome da área sendo: C para Computação; E para Educação e M para Matemática. O número que sucede o prefixo é atribuído sequencialmente para diferenciação entre as questões. Os resultados do questionário aplicado ao especialista em Computação podem ser observados na TABELA 2.

TABELA 2 – RESULTADOS DO QUESTIONÁRIO DO ESPECIALISTA EM COMPUTAÇÃO

Questionário para Especialista em Computação	Relevância
C1: Facilidade de acesso	4
C2: Facilidade de uso e apresentação de uma Interface bem planejada e de fácil manipulação	3
C3: Multiplataforma (Grau de compatibilidade)	4
C4: Ausência de erros de programação (falhas e <i>bugs</i> )	4
C5: Design de cores e <i>layout</i> amigável, proporcionando boa experiência de uso.	3
C6: Considera a individualidade do aluno (ritmo e forma de aprendizagem)	4
C7: Ajuda ou orientação ao(à) utilizador(a) durante o uso	4
C8: Eficiência e tempo de resposta dos comandos.	3
C9: Qualidade e correção científica dos conteúdos	4
C10: Clareza no uso de termos e conceitos	4
C11: Qualidade dos <i>feedbacks</i>	4
C: Avaliação Global (Domínio Técnico):	4

FONTE: Dados do autor (2018)

Na TABELA 2 tivemos algumas questões classificadas com relevância 3 (bom), como a questão C5 sobre o *design* das cores, a questão C8 no que se refere ao tempo de resposta aos comandos, e a questão C2 que tratava sobre a manipulação da interface. Os outros itens do questionário, tiveram relevância 4 (excelente), como por exemplo a questão C11 que se refere à qualidade dos *feedbacks*, e a questão C3 que estava relacionada ao grau de compatibilidade com diferentes plataformas.

Considerando que as questões aplicadas aos especialistas são, em sua maioria, voltadas à ergonomia do *software*, os resultados nos dão subsídio para afirmar que o STI - Lógica Livre cobre suficientemente os critérios elencados por Scapin e Bastien (1997) na seção 3.3 deste trabalho, critérios estes que são utilizados para a avaliação da qualidade ergonômica da interação com os sistemas computacionais. Estar de acordo com estes critérios valoriza a potencialidade pedagógica do trabalho, pois, segundo Catapan et al. (1999), esta potencialidade está diretamente ligada à sua qualidade ergonômica.

A TABELA 3 apresenta os resultados do questionário aplicado ao especialista em Educação, sobre o qual iremos explicar a seguir.

TABELA 3 – RESULTADOS DO QUESTIONÁRIO DO ESPECIALISTA EM EDUCAÇÃO

Questionário para Especialista em Educação	Relevância
E1: Relevância para o desenvolvimento de competências	1
E2: Possibilidade de articulação/integração curricular	4
E3: Adequação da linguagem ao público-destinatário	3
E4: Correção linguística	2
E5: Considera a individualidade do aluno (ritmo e forma de aprendizagem)	3
E6: Clareza da Linguagem	3
E7: Comunicação bidirecional (Tutor - Aluno)	2
E8: Facilidade de uso e apresentação de uma Interface bem planejada e de fácil manipulação	3
E9: Design de cores e layout amigável, proporcionando boa experiência de uso	3
E10: Ajuda ou orientação ao(à) utilizador(a) durante o uso	3
E11: Qualidade dos <i>feedbacks</i>	1
E: Avaliação Global (Domínio Pedagógico e Linguístico):	1

FONTE: Dados do autor (2018)

Os dados da TABELA 3 mostram que o especialista em Educação atribuiu relevância 1 nas questões E1 e E11 referentes ao desenvolvimento de competências e à qualidade dos *feedbacks* respectivamente e relevância 2 para as questões E4 e E7 que tratam da correção linguística e comunicação bidirecional entre o tutor e o aluno respectivamente. Porém, classificou como relevância 3 (Bom) a maioria das questões, como por exemplo E5, que diz considerar a individualidade do aluno e seu ritmo de aprendizagem e a questão E8 que se refere à facilidade de uso e qualidade da interface com o usuário. E ainda classificou com relevância 4 (excelente) a questão E2 sobre a articulação/integração curricular do conteúdo abordado pelo tutor. A explicação sobre a atribuição das relevâncias nos itens do questionário foram esclarecidas na questão aberta do questionário - fator qualitativo (ver APÊNDICE C).

A avaliação realizada pelo especialista em Educação apresentou carência ao *software* no que se refere ao desenvolvimento de competências e qualidade dos *feedbacks*. Ao fazer suas considerações qualitativas o especialista em Educação explica que o motivo de ter classificado a questão sobre o desenvolvimento de competências com peso 1 (ruim) foi o fato de necessitar uma melhor formulação da questão. Ao explicar sobre a classificação da questão relacionada aos *feedbacks*, o especialista sugere que o sistema possa prover um *feedback* construtivo, o que de fato, nos parece uma sugestão pertinente. Vale ressaltar que não encontramos algo parecido nos trabalhos relacionados ou em qualquer STI que tenhamos estudado. Dessa forma, consideramos esta sugestão como um desafio a ser cumprido pela equipe de pesquisa. O especialista finaliza suas considerações com uma pergunta sobre a funcionalidade de ajuda ao utilizador que é oferecida pelo STI - Lógica Livre. Em sua pergunta, ele diz que “a ajuda ao utilizador só se refere a apresentação de conceitos básicos, será que somente isso possibilitará uma ajuda consistente para sanar a dúvida?”. Como resposta, pensamos que o propósito na construção da ajuda é justamente mostrar conceitos elementares do conteúdo, de forma que o aluno

irá progredir mediante a resolução dos desafios. Posteriormente (ao avançar nos conteúdos) ele recebe outro tipo de ajuda, que é o *feedback* personalizado, o qual é capaz de mostrar conceitos mais avançados conforme a necessidade do aluno. Ademais, as outras respostas do especialista em educação classificaram o *software* como bom ou razoável, e até mesmo com uma classificação excelente no que se refere à articulação/integração curricular.

Na TABELA 4, temos os resultados referentes ao questionário aplicado ao especialista em Matemática.

TABELA 4 – RESULTADOS DO QUESTIONÁRIO DO ESPECIALISTA EM MATEMÁTICA

Questionário para Especialista em Matemática	Relevância
M1: Qualidade e correção científica dos conteúdos	4
M2: Adequação dos conteúdos ao público-destinatário	4
M3: Exemplos de exercícios abordados são adequados e capazes de mediar o conhecimento aos alunos	3
M4: Avalia a aquisição do conhecimento do aluno de forma suficiente para avançar no conteúdo	4
M5: Considera a individualidade do aluno (ritmo e forma de aprendizagem)	4
M6: Pertinência dos conteúdos face à natureza da temática e aos objetivos curriculares	4
M7: Clareza no uso de termos e conceitos	3
M8: Qualidade das ferramentas para sanar as dúvidas do aluno durante o uso do software	4
M9: Facilidade de acesso	4
M10: Facilidade de uso e apresentação de uma Interface bem planejada e de fácil manipulação	4
M11: Qualidade dos <i>feedbacks</i>	4
M: Avaliação Global (Domínio Científico):	4

FONTE: Dados do autor (2018)

O especialista no campo da Matemática considerou como classificação 3 (bom) as questões M3 e M7, as quais se referem ao exemplo de exercícios abordados e a clareza no uso de termos e conceitos. Os demais pontos avaliados receberam classificação 4 (Excelente) abordando assuntos como: qualidade de correção científicas dos conteúdos (M1); a forma com que o *software* avalia a aquisição dos conhecimentos do aluno para avançar no conteúdo (M4); a pertinência dos conteúdos face à natureza da temática e aos objetivos curriculares (M6); e por fim, sobre a qualidade dos *feedbacks* (M11).

Ter obtido classificação excelente nos itens M1 e M2 são resultados importantes para a pesquisa, pois dão indícios que o STI - Lógica Livre implementa o conceito de Transposição Didática de Chevallard (1991), que consiste em submeter um conteúdo formal a um conjunto de transformações adaptativas que irão torná-lo apto a ser ensinado e aprendido, sem perda de rigor científico. Outra característica em que o STI - Lógica Livre se mostrou excelente foi a qualidade da ferramenta para sanar dúvidas do aluno durante o uso do *software*. A classificação obtida por essa questão pelo especialista em Matemática responde o questionamento feito pelo

especialista em Educação, sobre a ajuda ao utilizador (ver APÊNDICE C), sendo esta adequada para sanar as dúvidas do aluno. Outras características foram classificadas como excelentes pelo especialista em Matemática, por exemplo: considerar a individualidade do aluno perante o ritmo de aprendizagem; a avaliação do conhecimento do aluno de forma suficiente para avançar no conteúdo; e a qualidade dos *feedbacks*. Essas três características, mostram que o tutor pode ser capaz de prover um ensino um-para-um, como propõe Elsom-Cook (1987) ao descrever as vantagens preliminares de um STI.

Nas observações qualitativas do especialista em Matemática (ver APÊNDICE D), recebemos apenas correções simples, por exemplo, grafia de palavras e sugestões como a não necessidade de explicar sobre um mesmo conectivo duas vezes, ou então, parênteses desnecessários nas expressões. Consideramos tais sugestões como pertinentes de modo que as incluamos nas alterações pendentes.

Em geral, consideramos que as avaliações dos especialistas foram pertinentes, tanto para validação do *software* como ferramenta educacional, bem como nos ofereceram oportunidades de melhorias. Outra contribuição que as avaliações nos proporcionaram foi nos fornecer subsídio para que pudéssemos realizar experimentos com alunos de graduação, garantindo maior segurança na abordagem a ser aplicada. Iremos discorrer sobre esse experimento nos parágrafos a seguir.

O experimento de pré e pós teste, bem como o uso do sistema STI - Lógica Livre pelos alunos do curso de Licenciatura em Computação, trouxeram um apanhado de dados que nos permitiram fazer algumas análises relevantes para o trabalho. Estes experimentos foram aplicados com os alunos do curso de Licenciatura em Computação da UFPR - Setor Palotina, matriculados na disciplina de Introdução à Lógica. O experimento foi constituído da análise de desempenho dos alunos mediante um pré teste (APÊNDICE A), e um pós teste (APÊNDICE B), além da utilização do sistema por alunos matriculados na Disciplina de Lógica. Para verificar se realmente houve progresso na aprendizagem, optou-se por dividir os alunos em dois grupos: o grupo de estudo e o de apoio.

A divisão em dois grupos ocorreu de maneira aleatória, de modo que o grupo de estudo consistiu nos alunos que utilizaram o STI - Lógica Livre, enquanto o grupo de apoio continuou estudando segundo os métodos tradicionais de resolução de exercícios. É importante frisar que o grupo de apoio foi fundamental para verificar se um possível progresso viria a ser consequência da utilização do *software* ou não.

O experimento constituiu-se na seguinte ordem de atividades: (1) a realização do pré teste; (2) utilização do STI - Lógica Livre pelo grupo de estudo; e (3) aplicação do pós teste. Todos os alunos realizaram tanto o pré quanto o pós teste. Vale ressaltar que o nível de complexidade das questões é maior no pós teste do que no pré teste.

Durante a segunda etapa (utilização do STI - Lógica Livre pelo grupo de estudo), foram coletado dados durante a utilização do sistema, armazenando os *logs* de utilização no banco de dados, que consistem nos erros e acertos face aos desafios resolvidos pelos alunos.



Deste modo, foi possível verificar quem realmente utilizou o sistema.

As médias de acerto das respostas ao pré teste dos alunos do grupo de apoio, foram agrupadas por Regras de Equivalência Proposicional, conforme a regra abordada pela questão (ver FIGURA 23<sup>1</sup>).

FIGURA 23 – RESULTADOS DO PRÉ TESTE DO GRUPO DE APOIO

Pré Teste com os alunos da equipe de Apoio											
ALUNOS	Aplicar Regras	ID	COM	ASSOC	DIST	DM	COND	BICOND	CP	EI	Total
Aluno_1	0,0	0,0	0,0	0,0	10,0	0,0	0,0	0,0	0,0	0,0	10%
Aluno_2	0,0	0,0	5,0	0,0	10,0	6,7	0,0	0,0	0,0	0,0	22%
Aluno_3	0,0	0,0	5,0	5,0	0,0	0,0	0,0	5,0	0,0	0,0	15%
Aluno_4	6,0	10,0	10,0	5,0	10,0	0,0	6,7	0,0	0,0	0,0	48%
Aluno_5	0,0	5,0	10,0	5,0	0,0	0,0	6,7	10,0	0,0	10,0	47%
Aluno_6	0,0	10,0	10,0	5,0	0,0	0,0	0,0	5,0	0,0	10,0	40%
Aluno_7	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0%
Aluno_8	8,0	10,0	10,0	5,0	0,0	6,7	10,0	10,0	10,0	10,0	80%
Aluno_9	0,0	5,0	0,0	5,0	10,0	0,0	0,0	0,0	0,0	10,0	30%
Aluno_10	0,0	10,0	10,0	5,0	10,0	10,0	10,0	10,0	10,0	10,0	85%
Aluno_11	2,0	5,0	10,0	5,0	10,0	6,7	10,0	0,0	0,0	0,0	49%

Média Total
39%

FONTE: Imagem do autor (2018)

De acordo com a figura, houve uma diversidade de ocorrências. Tivemos dois alunos com médias iguais e acima de 80% (Aluno\_8 e o Aluno\_10), assim como tivemos alunos que ficaram abaixo de 20% de acerto (Alunos 1, 3 e 7). Já a média de acerto geral do grupo foi de 39%.

No que diz respeito aos resultados do pós teste com o grupo de apoio, na FIGURA 24, são apresentadas as médias de acerto por aluno.

FIGURA 24 – RESULTADOS DO PÓS TESTE DO GRUPO DE APOIO

Pós Teste com os alunos da equipe de Apoio											
ALUNOS	Aplicar Regras	ID	COM	ASSOC	DIST	DM	COND	BICOND	CP	EI	Total
Aluno_1	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0%
Aluno_2	0,0	5,0	5,0	5,0	5,0	10,0	0,0	0,0	0,0	0,0	30%
Aluno_3	0,0	5,0	0,0	0,0	0,0	0,0	0,0	10,0	0,0	0,0	15%
Aluno_4	8,0	5,0	0,0	10,0	10,0	10,0	5,0	10,0	0,0	5,0	63%
Aluno_5	0,0	5,0	10,0	5,0	5,0	10,0	5,0	10,0	5,0	5,0	60%
Aluno_6	0,0	10,0	10,0	5,0	0,0	0,0	0,0	0,0	0,0	0,0	25%
Aluno_7	0,0	5,0	5,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	10%
Aluno_8	4,0	5,0	10,0	5,0	0,0	0,0	0,0	10,0	5,0	0,0	39%
Aluno_9	0,0	5,0	0,0	5,0	0,0	10,0	0,0	0,0	0,0	0,0	20%
Aluno_10	2,0	0,0	10,0	5,0	10,0	10,0	10,0	10,0	10,0	10,0	77%
Aluno_11	0,0	5,0	5,0	5,0	5,0	5,0	5,0	5,0	0,0	0,0	35%

Média Total
34%

FONTE: Imagem do autor (2018)

<sup>1</sup>Para garantir o anonimato dos alunos atribuímos rótulos com numeração de Aluno\_1 até Aluno\_11 para o grupo de apoio.

Assim como no pré teste, houve também uma grande diversidade entre os resultados, tivemos alunos com médias de acerto de até 77% (Aluno\_10), assim como houveram casos abaixo de 20% (Alunos 1, 3 e 7). A média de acerto geral ficou em 34%.

Assim como no grupo de apoio, também se fez necessário ocultar as identidades dos alunos do grupo de estudo. Porém, como forma de diferenciar os integrantes, utilizamos letras de A até J como pós fixo.

Análogo à apresentação dos dados obtidos no pré e pós teste do grupo de apoio, serão apresentados os dados do grupo de estudo. Abaixo temos a FIGURA 25<sup>2</sup>, que nos apresenta os resultados do pré teste realizado pelo grupo de estudo (alunos que utilizaram o STI - Lógica Livre).

FIGURA 25 – RESULTADOS DO PRÉ TESTE DO GRUPO DE ESTUDO

Pré Teste com os alunos da equipe de Teste											
ALUNOS	Aplicar Regras	ID	COM	ASSOC	DIST	DM	COND	BICOND	CP	EI	Total
Aluno_A	0,0	5,0	5,0	0,0	0,0	6,7	0,0	5,0	0,0	0,0	22%
Aluno_B	0,0	0,0	10,0	5,0	10,0	10,0	3,3	5,0	5,0	10,0	58%
Aluno_C	2,0	5,0	10,0	10,0	10,0	6,7	6,7	5,0	0,0	10,0	65%
Aluno_D	10,0	10,0	10,0	10,0	10,0	10,0	10,0	10,0	0,0	10,0	90%
Aluno_E	0,0	0,0	10,0	5,0	10,0	3,3	6,7	5,0	5,0	10,0	55%
Aluno_F	0,0	5,0	0,0	5,0	0,0	6,7	6,7	5,0	0,0	10,0	38%
Aluno_G	2,0	5,0	10,0	5,0	10,0	0,0	10,0	10,0	0,0	10,0	62%
Aluno_H	4,0	0,0	0,0	10,0	10,0	6,7	0,0	0,0	0,0	10,0	41%
Aluno_I	10,0	10,0	10,0	10,0	10,0	10,0	6,7	5,0	10,0	10,0	92%
Aluno_J	2,0	5,0	10,0	5,0	10,0	10,0	6,7	0,0	0,0	0,0	49%

Média Total
57%

FONTE: Imagem do autor (2018)

No grupo de estudos, houve alunos que apresentaram médias de acertos de 90% e 92% (Alunos D e I). A média dos outros alunos se manteve acima de 50%, com exceção dos Alunos A, F, H e J, que obtiveram 22%, 38%, 41% e 49% respectivamente. Dessa forma, o grupo de estudo alcançou uma média de acerto geral de 57% no pré teste. Os dados obtidos no pós teste do grupo de estudos são apresentados na FIGURA 26.

<sup>2</sup>Para garantir o anonimato dos aluno, atribuímos rótulos com letras de A a J para o grupo de estudo.

FIGURA 26 – RESULTADOS DO PÓS TESTE DO GRUPO DE ESTUDO

Pós Teste com os alunos da equipe de Teste

ALUNOS	Aplicar Regras	ID	COM	ASSOC	DIST	DM	COND	BICOND	CP	EI	Total
Aluno_A	10,0	5,0	10,0	5,0	0,0	0,0	0,0	0,0	0,0	0,0	30%
Aluno_B	2,0	5,0	10,0	10,0	5,0	10,0	5,0	0,0	0,0	0,0	47%
Aluno_C	10,0	10,0	10,0	10,0	10,0	10,0	10,0	10,0	10,0	10,0	100%
Aluno_D	10,0	10,0	10,0	10,0	10,0	10,0	10,0	10,0	5,0	10,0	95%
Aluno_E	4,0	5,0	10,0	5,0	5,0	10,0	5,0	10,0	10,0	5,0	69%
Aluno_F	0,0	5,0	0,0	10,0	5,0	10,0	5,0	10,0	5,0	5,0	55%
Aluno_G	8,0	10,0	10,0	10,0	5,0	10,0	10,0	10,0	5,0	10,0	88%
Aluno_H	8,0	10,0	10,0	10,0	10,0	5,0	10,0	10,0	10,0	5,0	88%
Aluno_I	8,0	10,0	10,0	10,0	10,0	10,0	10,0	10,0	5,0	10,0	93%
Aluno_J	0,0	10,0	10,0	5,0	10,0	5,0	5,0	0,0	0,0	0,0	45%

Média Total
71%

FONTE: Imagem do autor (2018)

Os dados da FIGURA 26 mostram que 5 alunos (50% do grupo) conseguiram médias de acerto acima de 87% (Alunos C, D, G, H e I), sendo que o Aluno\_C atingiu 100% de acertos. Ademais, 3 dos alunos ficaram abaixo de 50% (Alunos A, B e J) e outros 2 se mantiveram acima, com médias razoáveis (Alunos E e F). Contudo, a média de acerto geral que o grupo de estudo atingiu no pós teste foi de 71%.

Para uma análise mais assertiva dos dados acima, confrontamos as informações de pré e pós teste de cada grupo em números relativos (ver FIGURAS 27 e 28), possibilitando assim, mensurar o nível de progresso dos alunos de forma geral e individual.

FIGURA 27 – RELAÇÃO ENTRE ACERTOS DO GRUPO DE APOIO

Grupo Apoio	Total
Aluno_1	-10%
Aluno_2	8%
Aluno_3	0%
Aluno_4	15%
Aluno_5	13%
Aluno_6	-15%
Aluno_7	10%
Aluno_8	-41%
Aluno_9	-10%
Aluno_10	-8%
Aluno_11	-14%
<b>Total</b>	<b>-5%</b>

FONTE: Imagem do autor (2018)

FIGURA 28 – RELAÇÃO ENTRE ACERTOS DO GRUPO DE ESTUDO

Grupo Estudo	Total
Aluno_A	8%
Aluno_B	-11%
Aluno_C	35%
Aluno_D	5%
Aluno_E	14%
Aluno_F	17%
Aluno_G	26%
Aluno_H	47%
Aluno_I	1%
Aluno_J	-4%
<b>Total</b>	<b>14%</b>

FONTE: Imagem do autor (2018)

Mesmo levando em conta que o nível de dificuldade do pós teste é ligeiramente maior que do pré teste, é importante analisarmos a FIGURA 27, no que se refere ao grupo de apoio: seis dos onze alunos desse grupo apresentaram redução na média de acertos, como por exemplo, o Aluno\_8, o qual obteve 80% de acertos no pré teste decaindo para 39% no pós teste (redução de 41%), assim como o aluno\_10 que obteve 85% no pré teste decaindo para 77% no pós teste (redução de 8%). Os outros alunos que tiveram resultados inferiores foram os Alunos 1, 6, 9 e

11 (redução de 10%, 15%, 10% e 14% respectivamente). Porém, houve melhoras de até 15%, como é o caso do Aluno\_4. A média geral de acertos do grupo de apoio no pré teste foi de 39%, decaindo para 34% no pós teste, ou seja, o desempenho geral do grupo de apoio caiu 5%.

Análogo a isso, no grupo de estudo ( ver FIGURA 28), temos um cenário bem diferente: apenas dois dos dez alunos desse grupo tiveram um desempenho reduzido no pós teste (Aluno\_B decaiu 11% e o Aluno\_J decaiu 4%). Os outros oito alunos obtiveram aumento de desempenho, inclusive alguns apresentaram um progresso expressivo, como por exemplo o Aluno\_H que alcançou 47% de aumento no seu desempenho seguido do Aluno\_C com aumento de 35%.

No geral, podemos afirmar que o grupo de estudos obteve uma melhora significativa no desempenho em relação ao grupo de apoio, ou seja o auxílio do STI - Lógica Livre pode ter contribuído à aqueles que o utilizaram.

Após a análise entre os dois grupos, investigamos também os dados individuais de uso do sistema entre os alunos do grupo de estudo, durante o período em que utilizavam o STI para aprenderem Equivalência Lógica Proposicional como conteúdo curricular obrigatório. Para isso, foram recolhidos *logs* de utilização do sistema (ver na TABELA 5), onde encontra-se a informação da quantidade de desafios realizados pelos alunos, individualmente.

TABELA 5 – DADOS DE UTILIZAÇÃO DO STI - LÓGICA LIVRE PELOS ALUNOS

<b>Nome</b>	<b>Desafios Resolvidos</b>	<b>Acertos</b>	<b>Erros Justificáveis</b>	<b>Erros</b>
Aluno_A	0	0	0	0
Aluno_B	0	0	0	0
Aluno_C	207	189	6	12
Aluno_D	118	97	7	14
Aluno_E	76	52	12	12
Aluno_F	9	3	3	3
Aluno_G	158	142	9	7
Aluno_H	393	362	11	20
Aluno_I	178	141	23	14
Aluno_J	45	35	7	3
Média Absoluta	54,9	40,9	6,4	7,6

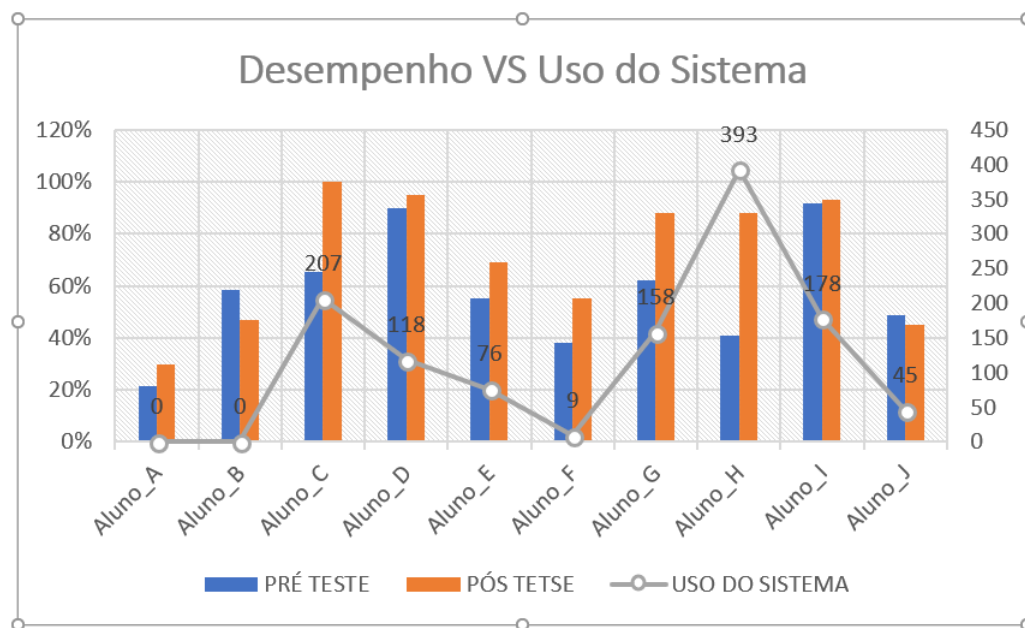
FONTE: Dados do autor (2018)

Os dados da TABELA 5 trazem o total de desafios que os alunos resolveram no sistema, a quantidade de acertos, a quantidade de erros que são justificáveis<sup>3</sup> e por último, a quantidade de erros. Os dados evidenciam que existiram alunos do grupo de estudos que sequer utilizaram o STI - Lógica Livre (Alunos A e B), ou resolveu uma quantidade relativamente pequena de desafios (Aluno\_F resolveu apenas 9 desafios) o qual obteve, uma baixa quantidade de acertos em relação aos demais (apenas 33%). Por outro lado, existiram alunos que resolveram centenas de desafios, como é o caso do Aluno\_H que resolveu 393 desafios mantendo uma média de

<sup>3</sup>Quando o Tutor encontrou um raciocínio (possivelmente) seguido pelo aluno perante a resposta, por exemplo, se o aluno aplica uma Regra corretamente em determinada expressão, mas não utiliza a Regra solicitada

acertos de 92%, assim como o Aluno\_C que resolveu 207 desafios e dentre eles 91% estavam completamente corretos. Mesmo com os casos de não uso do sistema, os alunos resolveram, em média, 54,9 desafios entre eles e mantiveram uma média de 74,5% de acertos, 11,6% de erros justificáveis e apenas 13,9% de erros. Para melhor visualização desses dados, fizemos uma correlação com o desempenho individual dos alunos no pré e pós teste, como vemos na FIGURA 29.

FIGURA 29 – RELAÇÃO ENTRE DESEMPENHO NOS TESTES E QUANTIDADE DE UTILIZAÇÃO DO STI - LÓGICA LIVRE



FONTE: Imagem do autor (2018)

Na FIGURA 29 temos um gráfico que relaciona o desempenho dos alunos do grupo de estudos (no pré e pós teste) com a quantidade de desafios resolvidos no STI - Lógica Livre. Para representar essa relação utilizamos uma estrutura de gráfico que contém três eixos, um eixo em X, que representa os alunos, e dois eixos em Y: o eixo Y da esquerda (com escala de porcentagem) representa o desempenho no pré e pós teste; já o eixo Y da direita (com escala absoluta) representa a quantidade de desafios que foram resolvidos pelo aluno no STI - Lógica Livre.

Ao analisar esse gráfico, percebemos uma forte relação entre o desempenho dos alunos e a utilização do sistema. Pois, os alunos que mais resolveram desafios, são os que tiveram um ganho maior em relação ao pré teste (por exemplo, o aluno\_C e o aluno\_H). Já os alunos que menos resolveram desafios, foram os que tiveram desempenho inferior (como por exemplo o Aluno\_A, Aluno\_B, Aluno\_F e Aluno\_J). Vale ressaltar ainda que os dois únicos alunos que apresentaram desempenho inferior no pós teste em relação ao pré teste, foram os alunos B e J, sendo que o Aluno\_B não utilizou o STI - Lógica Livre e o Aluno\_J resolveu apenas 45 desafios. Dessa forma, os dados nos mostram que, ao utilizar o STI - Lógica Livre, na medida

em que o aluno resolve mais desafios, ele progride no conteúdo.

Após a utilização do sistema e realização do pós teste, foi aplicado um questionário apenas aos alunos que haviam se cadastrado no sistema, com intuito de verificar a experiência de uso dos mesmos. Ao todo, foram aplicadas nove questões, sendo três delas para solicitar permissão de uso dos dados, identificação do semestre atual e gênero dos alunos. As outras seis questões eram referentes ao uso do sistema: três delas eram de múltipla escolha e três perguntas abertas não obrigatórias, nas quais o participante poderia discorrer sobre suas percepções. O questionário foi respondido por oito alunos, e dentre estes, dois não responderam as questões abertas. A seguir, apresentaremos às questões juntamente com as respostas obtidas.

Ao questionar sobre a impressão que os alunos tiveram do sistema (questão de múltipla escolha: ruim, razoável, boa ou ótima), todos afirmaram ter uma boa impressão do STI - Lógica Livre.

Sobre os desafios elaborados pelo Tutor, 100% dos alunos também entendem que foram satisfatórios, atendendo suas expectativa. Este dado se mostra interessante pelo fato de que o Tutor elabora questões personalizadas em tempo de execução do sistema. Sendo assim, ao afirmarem que tais exercícios foram satisfatórios, as respostas dos alunos nos conduzem a crer que o STI - Lógica Livre está sendo assertivo na proposta de levar em consideração a individualidade do aluno.

Uma das questões abertas do questionário solicitava que os alunos se posicionassem acerca da contribuição que o STI proporcionou em seu desenvolvimento na disciplina Lógica (ver FIGURA 30).

#### FIGURA 30 – SEXTA QUESTÃO DO FORMULÁRIO SOBRE A EXPERIÊNCIA DE USO

6. De forma sucinta, diga qual foi a contribuição que o Sistema teve para o seu desenvolvimento na disciplina de lógica.

7 responses

Facilitou muito o entendimento das Equivalências Lógicas.
Foi praticamente 100% do meu entendimento da matéria
Fixação de conteúdo
Sim, um lógica de sistema
Ajudou na diferenciação das equivalencias
Ajudou a memorizar as regras de equivalencia
Bem mais facil que nos livros

FONTE: Imagem do autor (2018)

As respostas em geral foram positivas, contudo, obtivemos somente sete respostas, das quais destacam-se: “Bem mais fácil que nos livros” e “Foi praticamente 100% do meu

entendimento da matéria". Essas respostas são indícios, que o método de Transposição Didática, proposto por Chevallard (1991), tem sua devida relevância dentro do nosso trabalho, pois o STI - Lógica Livre conseguiu apresentar o mesmo conteúdo dos livros, mas que no entendimento do aluno, se mostrou menos complicado.

Outra questão aberta no questionário inquiriu da amostra os pontos positivos e negativos do STI - Lógica Livre. No tocante a esta pergunta obtivemos somente sete respostas (ver FIGURA 31).

FIGURA 31 – SÉTIMA QUESTÃO DO FORMULÁRIO SOBRE A EXPERIÊNCIA DE USO

### 7. Diga o que mais gostou e o que menos gostou no Sistema.

7 responses

O que mais Gostei foi a calculadora das equivalências. O que não agradou muito, foi as inumeras repetições de questões .
Gostei que ele tem diferentes meios de avaliar, não gostei repetir varias vezes proposições com respostas iguais as anteriores
As formas geometricas para maior entendimento da matéria
Aham, sim gostei sistema
Organização,Bug da porcentagem de conclusão
o que achei mais interessante foi a forma com que as regras são mostradas por meio de formas geométricas, e o que não gostei foi o bug das porcentagens diminuir
é bem didatico mas repete muito os exercicios

FONTE: Imagem do autor (2018)

Conforme a FIGURA 31, recebemos dos alunos alguns prós e contras do sistema. Como ponto negativo, foi indicado que o sistema repetia muitas vezes as mesmas questões. Isso pode acontecer caso o aluno tenha errado várias vezes, porém, esta informação nos serve como dado de estudo, com ela continuaremos os testes do sistema, para que possamos talvez, recalibrar alguns parâmetros na tomada de decisão do STI - Lógica Livre, se assim se fizer necessário. Outra informação que levaremos para o laboratório de testes é a afirmação da existência de *bugs* no relatório de desempenho.

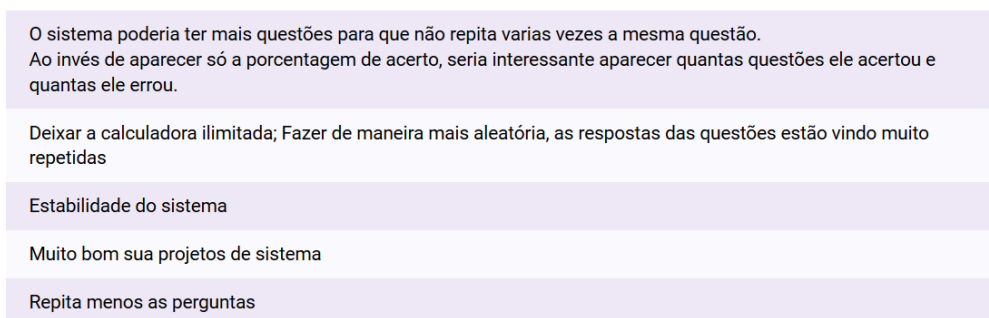
Contudo, apesar de cinco dos sete alunos que responderam esta questão terem contribuído com os pontos negativos, todos os sete contribuíram com pontos positivos no sistema, dentre eles, tivemos a consideração sobre a Calculadora Lógica, que possibilita comunicação bidirecional entre o Aluno e o Tutor e as diferentes formas de avaliar. Dois dos alunos elogiaram as figuras geométricas que substituem as proposições. Ao questionar os alunos sobre a possível utilização do STI - Lógica Livre em turmas posteriores, todos os alunos afirmaram ser importante que o sistema seja ofertado às próximas turmas. Este é mais um fator que nos motiva a dar continuidade ao projeto.

Ao solicitar sugestões aos alunos, apenas 5 deles redigiram pontos de melhorias. Tais respostas apresentadas na FIGURA 32, serão importante para ajustes futuros nas regras de tomada de decisão do tutor.

FIGURA 32 – NONA QUESTÃO DO FORMULÁRIO SOBRE A EXPERIÊNCIA DE USO

### 9. Aponte sugestões de melhoria para nosso sistema.

5 responses



FONTE: Imagem do autor (2018)

A FIGURA 32 nos mostra que os cinco alunos reafirmaram as repetições dos exercícios, por isso, voltaremos a testar esta característica do sistema. Porém, uma sugestão que se mostrou pertinente é sobre o relatório de desempenho, em que o aluno sugere mostrar, além da porcentagem de desempenho, o valor absoluto das questões.

Os experimentos junto aos alunos do Curso de Licenciatura em Computação da UFPR - Setor Palotina e com os especialistas contribuíram com muitos resultados para o presente estudo, de forma que obtivemos dados suficientes para cumprir nosso objetivo geral, que consistiu em desenvolver e aplicar um STI para auxiliar os alunos do ensino superior no processo de aprendizagem de Equivalência Lógica Proposicional, avaliando-o enquanto ferramenta educacional.



## 7 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, apresentamos o desenvolvimento e aplicação de um STI que auxiliou os alunos do ensino superior no processo de aprendizagem de Equivalência Lógica Proposicional. Tal *software* foi denominado STI - Lógica Livre. A metodologia *Development Research* foi utilizada para o seu desenvolvimento. Sendo assim, desenvolvemos e avaliamos a ferramenta educacional, bem como sua eficiência e eficácia, por meio de experimentos e investigações.

Para isso, realizamos uma revisão da literatura referente ao cenário educacional do ensino de Lógica, sobre estudos já realizados dentro da temática e também das tecnologias utilizadas no desenvolvimento. Aplicamos ainda um questionário sobre o ensino de Lógica aos alunos que já cursaram a disciplina, fundamentando o desenvolvimento do STI.

Construímos, então, o *software* STI - Lógica Livre, um sistema voltado para o ensino de Equivalência Lógica Proposicional. Para sua construção, fizemos uso de tecnologias na área da Inteligência Artificial, levando em conta a individualidade dos alunos perante seu ritmo de aprendizagem. Sua implementação foi alicerçada pelos Princípios Gerais para Engenharia de *Software*, além de seguir um padrão de usabilidade definido pelos Critérios Ergonômicos para o Desenvolvimento de *Software*, critérios estes que estão ligados diretamente com a potencialidade pedagógica do sistema.

Na modelagem pedagógica do STI, implementamos aspectos de algumas teorias da literatura educacional como: (i) teoria sócio-interacionista, com o objetivo de trabalhar dentro da Zona de Desenvolvimento Proximal do aluno; (ii) aprendizagem significativa, pela qual ligamos o conteúdo com objetos conhecidos pela realidade do aluno; (iii) alguns traços da teoria de condicionamento operante, oferecendo motivações para o aluno resolver mais desafios; (iv) teoria da construção do conhecimento, propondo desafios para desencadear o processo de assimilação, acomodação e adaptação do aluno perante o conteúdo; e (v) métodos de Transposição Didática submetendo o conteúdo à transformações adaptativas, sem perda de rigor científico, com objetivo de ser mais facilmente aprendido.

Após o desenvolvimento, avaliamos o STI, validando suas potencialidades enquanto ferramenta educacional. Isso foi feito por meio de: *feedbacks* de especialistas das áreas de Matemática, Computação e Educação; e dados recolhidos junto aos alunos matriculados na disciplina de Lógica na UFPR - Setor Palotina, na qual aplicamos experimentos com o uso do STI.

Este trabalho contribuiu, de modo geral, para que refletíssemos sobre as dificuldades no ensino e pudéssemos pensá-las nas possíveis alternativas tecnológicas, aliadas às pedagógicas, para auxiliar na aprendizagem. Como resultados, obtivemos índices experimentais relevantes. Podemos considerar, entre eles, o aumento significativo da média de acertos no pós teste em relação ao pré teste da equipe de estudo. Outro dado expressivo é a relação encontrada entre a porcentagem de desempenho e a utilização do sistema. Tal análise nos mostrou que os alunos que mais utilizaram o STI - Lógica Livre tiveram melhor progresso. Para complementar

o experimento, aplicamos um questionário aos alunos que utilizaram o sistema, com o intuito de confirmar os dados obtidos e angariar sugestões de melhorias.

Sendo assim, após receber sugestões dos alunos e dos especialistas, elencamos melhorias a serem realizadas e apresentadas em trabalhos futuros, dentre elas: (i) implementar dicas personalizadas, que também levam em conta o desempenho do aluno; (ii) novas modalidades de desafios para que tenhamos ainda mais formas de avaliação; (iii) desenvolver um relatório analítico de desempenho, no qual o aluno possa visualizar os números absolutos de erros e acertos, ou até mesmo rever os desafios e consultar novamente seus *feedbacks*; (iv) realizar novos testes com o STI, a fim de identificar possibilidades de melhorias na tomada de decisão do Módulo Tutor, por exemplo, identificar e corrigir o módulo para que haja menos repetição de exercícios; e (v) a implementar *feedbacks* construtivos, como proposto na avaliação do especialista em Educação.

De modo geral, os objetivos da atual pesquisa foram atingidos, deixando possibilidades de reaplicar o sistema em próximas turmas, além de tê-lo tornado de acesso livre, para que possa ser utilizado por outros pesquisadores.

## REFERÊNCIAS

- AHO, A. V.; SETHI, R.; ULLMAN, J. D. *Compilers, principles, techniques*. **Addison Wesley**, v. 7, n. 8, p. 9, 1986. Citado na página 28.
- ALAN, R. H. V.; MARCH, S. T.; PARK, J.; RAM, S. Design science in information systems research. **MIS quarterly**, Springer, v. 28, n. 1, p. 75–105, 2004. Disponível em: <[http://wise.vub.ac.be/sites/default/files/thesis\\_info/design\\_science.pdf](http://wise.vub.ac.be/sites/default/files/thesis_info/design_science.pdf)>. Acesso em: 31 de Outubro de 2017. Citado na página 33.
- AUSUBEL, D. P. **Aquisição e retenção de conhecimentos: uma perspectiva cognitiva**. Lisboa: Plátano, 2003. v. 1. Citado 4 vezes nas páginas 16, 19, 20 e 21.
- BACKUS, J. W.; BEEBER, R. J.; BEST, S.; GOLDBERG, R.; HAIBT, L. M.; HERRICK, H. L.; NELSON, R. A.; SAYRE, D.; SHERIDAN, P. B.; STERN, H. et al. **The FORTRAN automatic coding system**. Los Angeles, California, 1957. 188–198 p. Citado na página 28.
- CARBONELL, J. R. Ai in cai: An artificial-intelligence approach to computer-assisted instruction. **IEEE transactions on man-machine systems**, IEEE, v. 11, n. 4, p. 190–202, 1970. Disponível em: <<http://ieeexplore.ieee.org/abstract/document/4081977/?reload=true>>. Acesso em: 31 de Outubro de 2017. Citado 5 vezes nas páginas 15, 28, 30, 41 e 45.
- CARBONELL, J. R. Artificial intelligence and large interactive man computer systems. **BOLT BERANEK AND NEWMAN INC CAMBRIDGE MASS**, 1971. Disponível em: <<http://www.dtic.mil/docs/citations/AD0726441>>. Acesso em: 31 de Outubro de 2017. Citado na página 28.
- CASTRO, C. A aula tradicional: a iguana do ensino. **Fórum Educação e Conjuntura**, v. 10, p. 2005, 2005. Citado na página 26.
- CATAPAN, A. H.; CORNÉLIO, F.; SOUZA, A. d.; THOMÉ, Z.; CYBYS, W. Ergonomia em software educacional: a possível integração entre usabilidade e aprendizagem. **Atas Workshop sobre fatores humanos em sistemas computacionais: rompendo barreiras entre pessoas e computadores.**, Campinas, São Paulo: UNICAMP/SEEC, 1999. Citado 2 vezes nas páginas 21 e 60.
- CHEVALLARD, Y. La transposición didáctica. **Del saber sabio al saber enseñado**, v. 3, 1991. Citado 5 vezes nas páginas 21, 49, 55, 62 e 70.
- COPPIN, B. Inteligência artificial/ben coppin; tradução e revisão técnica jorge duarte pires valério. **Rio de Janeiro: LTC**, 2010. Citado na página 26.
- DARWIN, C. . **On the Origin of Species**, 1859. Citado na página 25.
- DUVAL, R. Registres de représentation sémiotique et fonctionnement cognitif de la pensée. **Annales de didactique et de sciences cognitives**, v. 5, n. 1, p. 37–65, 1993. Disponível em: <[https://mathinfo.unistra.fr/fileadmin/upload/IREM/Publications/Annales\\_didactique/vol\\_05/adsc5\\_1993-003.pdf](https://mathinfo.unistra.fr/fileadmin/upload/IREM/Publications/Annales_didactique/vol_05/adsc5_1993-003.pdf)>. Acesso em: 31 de Outubro de 2017. Citado na página 21.
- ELLIS, T. J.; LEVY, Y. A guide for novice researchers: Design and development research methods. **Proceedings of Informing Science & IT Education Conference (InSITE)**, p. 107–118, 2010. Citado na página 33.

ELSOM-COOK, M. **Intelligent Computer-Aided Instruction Research at the Open University. CITE Report No. 10.** EUA :Open University: ERIC, 1987. Citado 3 vezes nas páginas 16, 26 e 63.

FERNANDES, A. M. d. R. et al. . **Sistema especialista difuso aplicado ao processo de análise química qualitativa de amostras de minerais**, 1996. Citado na página 27.

GAVIDIA, J. J. Z.; ANDRADE, L. C. V. d. **Sistemas tutores inteligentes**. 24 p. Monografia — Universidade Federal do Rio de Janeiro, Rio de Janeiro–RJ, 2003. Citado na página 31.

GIRAFFA, L. M. M. **Uma arquitetura de tutor utilizando estados mentais**. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, 1999. Citado na página 16.

GIRAFFA, L. M. M.; VICCARI, R. M. Fundamentos de sistemas tutores inteligentes. **Porto Alegre: CPGCC-UFRGS**, 1998. Citado 2 vezes nas páginas 28 e 30.

GIUSTI, R. **Descoberta de regras de conhecimento utilizando computação evolutiva multiobjetivo**. Tese (Doutorado) — Universidade de São Paulo, 2010. Citado na página 25.

GRIVOKOSTOPOULOU, F.; PERIKOS, I.; HATZILYGEROUDIS, I. An intelligent tutoring system for teaching fol equivalence. **The First Workshop on AI-supported Education for Computer Science (AIEDCS 2013)**, p. 20, 2013. Citado 5 vezes nas páginas 15, 16, 20, 30 e 31.

HASAN, H. Information systems development as a research method. **Australasian Journal of Information Systems**, Rio de Janeiro, v. 1, n. 11, p. 4–13, 2003. Disponível em: <<http://ro.uow.edu.au/buspapers/458/>>. Acesso em: 31 de Outubro de 2017. Citado na página 33.

HOOKER, D. Seven principles of software development. **Wayback Machine Internet Archive**, 1996. Citado 2 vezes nas páginas 24 e 59.

KAPLAN, R.; ROCK, D. **New directions for intelligent tutoring systems**. EUA American: Springer verlag ny, 2011. v. 1. 30–40 p. Citado 2 vezes nas páginas 30 e 41.

KLIR, G.; YUAN, B. **Fuzzy sets and fuzzy logic**. New Jersey, USA: Prentice hall, 1995. v. 4. Citado na página 27.

LODDER, J.; HEEREN, B.; JEURING, J. A pilot study of the use of logex, lessons learned. **arXiv preprint arXiv:1507.03671**, 2015. Disponível em: <<https://arxiv.org/abs/1507.03671>>. Acesso em: 31 de Outubro de 2017. Citado 2 vezes nas páginas 31 e 32.

LODDER, J.; HEEREN, B.; JEURING, J. A domain reasoner for propositional logic. **J. UCS**, v. 22, n. 8, p. 1097–1122, 2016. Disponível em: <[http://www.jucs.org/jucs\\_22\\_8/a\\_domain\\_reasoner\\_for/jucs\\_22\\_08\\_1097\\_1122\\_lodder.pdf](http://www.jucs.org/jucs_22_8/a_domain_reasoner_for/jucs_22_08_1097_1122_lodder.pdf)>. Acesso em: 31 de Outubro de 2017. Citado na página 32.

LODDER, J.; PASSIER, H.; STUURMAN, S. Using ideas in teaching logic, lessons learned. In: IEEE. **Computer Science and Software Engineering, 2008 International Conference on**. [S.l.], 2008. v. 5, p. 553–556. Citado 4 vezes nas páginas 15, 16, 20 e 31.

MOSTAFAVI, B.; BARNES, T. Evolution of an intelligent deductive logic tutor using data-driven elements. **International Journal of Artificial Intelligence in Education**, Springer, v. 27, n. 1, p. 5–36, 2017. Disponível em: <<https://link.springer.com/article/10.1007/s40593-016-0112-1>>. Acesso em: 31 de Outubro de 2017. Citado 2 vezes nas páginas 16 e 32.

NILSSON, N. J. **Artificial intelligence: a new synthesis**. San Francisco, USA: Elsevier, 1998. Citado na página 24.

NWANA, H. S. Intelligent tutoring systems: an overview. **Artificial Intelligence Review**, Springer, v. 4, n. 4, p. 251–277, 1990. Disponível em: <<https://link.springer.com/article/10.1007/BF00168958>>. Acesso em: 31 de Outubro de 2017. Citado na página 15.

PAVLOV, I. P.; GANTT, W. H. **Conditioned reflexes and psychiatry**. [S.l.]: International publishers New York, 1941. v. 2. Citado na página 19.

PIAGET, J. **A equilibração das Estruturas cognitivas**. Rio de Janeiro: Zahar, 1975. Citado na página 19.

PIMENTEL, E. P.; FRANÇA, V. F. de; OMAR, N. A identificação de grupos de aprendizes no ensino presencial utilizando técnicas de clusterização. **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**, v. 1, n. 1, p. 495–504, 2003. Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/280>>. Acesso em: 31 de Outubro de 2017. Citado na página 15.

PRESSMAN, R. S. **Software engineering: a practitioner's approach**. New York, NY, EUA: Palgrave Macmillan, 2005. Citado 4 vezes nas páginas 23, 24, 35 e 59.

PY, M. X. Sistemas especialistas: uma introdução. **Universidade Federal do Rio Grande do Sul**. URL: [HTTP://www.inf.ufrgs.br/gppd/disc/cmp135/trabs/mpy/sistemasespecialistas.pdf](http://www.inf.ufrgs.br/gppd/disc/cmp135/trabs/mpy/sistemasespecialistas.pdf). Acessado em, v. 10, n. 11, 2009. Citado na página 26.

RICHEY, R. C.; KLEIN, J. D. **Design and development research: Methods, strategies, and issues**. Springer, New York, NY: Routledge, 2014. Citado na página 33.

RISSOLI, V. R. V.; GIRAFFA, L. M. M.; MARTINS, J. de P. Sistema tutor inteligente baseado na teoria da aprendizagem significativa com acompanhamento fuzzy. **Informática na educação: teoria & prática**, v. 9, n. 2, 2006. Citado 2 vezes nas páginas 26 e 27.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A modern approach**. Malaysia; Pearson Education Limited: Prentice-Hall, Englewood Cliffs, 1995. v. 25. 27 p. Citado na página 25.

SACRISTÁN, G.; GÓMEZ, A. P. **Compreender e transformar o ensino-4**. [S.l.]: Artmed Editora, 2009. Citado na página 20.

SCAPIN, D. L.; BASTIEN, J. C. Ergonomic criteria for evaluating the ergonomic quality of interactive systems. **Behaviour & information technology**, Taylor & Francis, v. 16, n. 4-5, p. 220–231, 1997. Citado 4 vezes nas páginas 21, 23, 35 e 60.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. et al. **Database system concepts**. McGraw-Hill New York: [s.n.], 1997. v. 4. Citado na página 40.

SILVA, A. P. Aplicações de sistemas tutores inteligentes na educação a distância: possibilidades e limites. **Anais do Seminário Nacional ABED de Educação a Distância**. Brasília: ABED, 2006. Disponível em: <<http://www.abed.org.br/seminario2006/pdf/tc056.pdf>>. Acesso em: 31 de Outubro de 2017. Citado na página 16.

SKINNER, B. F. Operant behavior. **American Psychologist**, American Psychological Association, v. 18, n. 8, p. 503, 1963. Citado na página 19.

SOBRAL, O. J. Inteligência humana: Concepções e possibilidades. **Revista Científica Fac-Mais, Inhumas, GO**, v. 3, n. 1, p. 31–46, 2013. Citado na página 24.

SUN, R. Artificial intelligence:.. **Connectionist and symbolic approaches**, Citeseer, 1999. Citado na página 25.

VEEN, W.; VRAKING, B. **Homo Zappiens: educando na era digital**. Porto Alegre : Artimed: Artmed Editora, 2009. Citado 3 vezes nas páginas 15, 49 e 54.

VICCARI, R. Um tutor inteligente para a programação em lógica–idealização. **Projeto e Desenvolvimento. Coimbra: Universidade de Coimbra**, 1990. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/17971>>. Acesso em: 31 de Outubro de 2017. Citado 3 vezes nas páginas 16, 30 e 47.

VYGOTSKY, L. S. **Mind in society: The development of higher psychological processes**. EUA: Library of congress cataloging in publication data: Harvard university press, 1978. Citado 6 vezes nas páginas 17, 19, 20, 29, 32 e 41.

WATSON, J. B. Psychology as the behaviorist views it. **Psychological review**, Psychological Review Company, v. 20, n. 2, p. 158, 1913. Citado na página 19.

WENGER, E. **Artificial intelligence and tutoring systems: computational and cognitive approaches to the communication of knowledge**. California: Morgan Kaufman Inc: Elsevier Science Publishers BV, 1988. v. 1. 311–311 p. Citado 2 vezes nas páginas 30 e 41.

## APÊNDICE A – PRÉ TESTE

### Teste de conhecimento sobre equivalências

Fábio Henrique Gil, Prof. Marcos Antonio Schreiner

<sup>1</sup> Curso de Licenciatura em Computação  
Universidade Federal do Paraná - UFPR  
Setor Palotina

Nome:..... Assinatura:.....

GRR..... Data:.....

*Resumo.* Este teste de conhecimento faz parte da avaliação do Sistema Tutor Inteligente (STI) Lógica Livre. Este STI foi desenvolvido para ser utilizado no ensino de equivalência no conteúdo de lógica proposicional. Ele é um dos resultados do projeto de TCC do aluno Fábio Henrique Gil no curso de Licenciatura em Computação da UFPR. Destacamos que o resultado deste teste será utilizado para avaliar o ganho na aprendizagem com o uso do STI-Lógica Livre na disciplina de Introdução à Lógica e os resultados poderão ser publicados em artigos científicos.

#### 1. Aplique as regras de equivalência nas seguintes fórmulas.

- a)  $((p \wedge q) \rightarrow r)$
- b)  $(p \vee r) \wedge (p \vee q)$
- c)  $(p \vee q)$
- d)  $(p \wedge q)$
- e)  $(r \rightarrow (p \rightarrow q))$

#### 2. Aplique a regra de equivalência INDICADA para as seguintes fórmulas.

##### 2.1. Idempotência (ID)

- a)  $s$
- b)  $\neg p \vee q$

##### 2.2. Comutação (COM)

- a)  $p \vee q$
- b)  $(\neg a \vee b) \wedge (b \rightarrow a)$

##### 2.3. Associação (ASSOC)

- a)  $(p \vee q) \wedge ((q \rightarrow r) \wedge (r \leftrightarrow p))$
- b)  $p \wedge (q \wedge r)$

##### 2.4. Distributiva (DIST)

- a)  $p \wedge (q \vee r)$
- b)  $(q \rightarrow r) \wedge ((p \rightarrow q) \wedge (q \rightarrow r))$

##### 2.5. De Morgan (DM)

- a)  $\neg(a \wedge b)$
- b)  $\neg p \wedge \neg q$
- c)  $\neg((q \vee r) \wedge (\neg p \vee q))$

##### 2.6. Condicional (COND)

- a)  $\neg p \rightarrow q$
- b)  $\neg r \vee s$
- c)  $\neg(q \vee r) \vee (\neg p \vee q)$

##### 2.7. Bicondicional (BICOND)

- a)  $\neg p \leftrightarrow q$
- b)  $p \vee q \leftrightarrow q \wedge r$

##### 2.8. Contraposição (CP)

- a)  $p \rightarrow q$
- b)  $\neg(p \vee q) \rightarrow \neg(q \wedge r)$

##### 2.9. Exportação-Importação (EI)

- a)  $p \wedge q \rightarrow r$
- b)  $p \rightarrow (q \rightarrow r)$

## APÊNDICE B – PÓS TESTE

### Teste de conhecimento sobre equivalências

Fábio Henrique Gil, Prof. Marcos Antonio Schreiner

Curso de licenciatura em Computação

Universidade Federal do Paraná – UFPR

Setor Palotina

Nome: \_\_\_\_\_ Assinatura: \_\_\_\_\_

GRR: \_\_\_\_\_ DATA: \_\_\_\_\_

**Resumo:** Este teste de conhecimento faz parte da avaliação do Sistema Tutor Inteligente (STI) Lógica Livre. Este STI foi desenvolvido para ser utilizado no ensino de equivalência no conteúdo de lógica proposicional. Ele é parte dos resultados do projeto de TCC do aluno Fábio Henrique Gil no curso de Licenciatura em Computação da UFPR. Destacamos que o resultado deste teste será utilizado para avaliar o ganho na aprendizagem com o uso do STI-Lógica Livre na disciplina de Introdução à Lógica e os resultados poderão ser publicados em artigos científicos.

**1. Aplique as Regras de equivalência nas seguintes Expressões.**

- a)  $(p \vee q) \wedge (r \vee s \vee t)$
- b)  $((p \wedge q) \rightarrow (r \vee s))$
- c)  $(p \rightarrow q) \vee (q \rightarrow p)$
- d)  $(r \rightarrow (p \rightarrow q))$
- e)  $(p \wedge q) \vee (r \wedge s \vee t)$

**2. Aplique a Regra de equivalência INDICADA para as seguintes Expressões**

**2.1. Idempotência (ID)**

- a)  $((a \wedge b) \rightarrow r)$
- b)  $t \wedge t$

**2.2. Comutação (COM)**

- a)  $(p \wedge q) \vee (\sim p \wedge \sim q)$
- b)  $a \vee (b \vee c)$

**2.3. Associação (ASSOC)**

- a)  $(a \vee q) \wedge (r \wedge s \wedge t)$
- b)  $(a \wedge b) \wedge c$

**2.4. Distributiva (DIST)**

- a)  $(p \vee q) \vee (r \wedge s \wedge t)$
- b)  $s \wedge (q \vee r)$

**2.5. De Morgan (DM)**

- a)  $\sim(m \wedge n)$
- b)  $\sim(\sim r \wedge \sim s)$

**2.6. Condicional (COND)**

- a)  $(r \rightarrow (p \rightarrow q))$
- b)  $a \vee (b \vee c)$

**2.7. Bicondicional (BICOND)**

- a)  $(a \rightarrow b) \wedge (b \rightarrow a)$
- b)  $(a \wedge b) \vee (\sim b \wedge \sim a)$

**2.8. Contraposição (CP)**

- a)  $(m \rightarrow n) \vee (n \rightarrow m)$
- b)  $c \rightarrow (b \rightarrow a)$

**2.9. Exportação-Importação (EI)**

- a)  $((m \wedge n) \rightarrow (o \vee p))$
- b)  $((a \wedge q) \rightarrow r)$



## APÊNDICE C – OBSERVAÇÕES QUALITATIVAS DO ESPECIALISTA EM EDUCAÇÃO

### Ponderações

E1: quais competências? somente da lógica? deveria ser especificadas na questão. E: A de se refletir sobre esse item, do ponto de vista pedagógico há algumas questões a repensar. O *feedback* por exemplo (E11), me parece mais informativo no qual informa quais questões o acadêmico(a) errou e quais acertou, não apresentando o porquê do erro. Há necessidade de se pensar em um *feedback* construtivo ou em uma forma de exercício que contribua com a construção do conhecimento. No item E10, a ajuda ao utilizador só se refere a apresentação de conceitos básicos, será que somente isso possibilitará uma ajuda consistente para sanar a dúvida?

## APÊNDICE D – OBSERVAÇÕES QUALITATIVAS DO ESPECIALISTA EM MATEMÁTICA

### **Ponderações**

Primeira tela: (Início)

Conectivos: quando aparece os símbolos, não precisa escrever de novo que a negação é um conectivo unário.

Corrigir a palavra “expressão”, que aparece como “expressão”. E a palavra “precedência” que está grafada como “prescedência”.

Na tela Orientações sobre Equivalência:

ASSOC: Corrigir parênteses. Sugestão: usar preposições ao invés de figuras geométricas.

DN: Corrigir a palavra “vezes” que aparece grafada incorretamente.

Talvez pudesse ter uma seta para passar de uma orientação para outra.

Nas respostas dos desafios, penso que há excesso de parênteses, como no caso:

$(p \wedge p) \wedge (p \wedge p)$ , que aparece  $((p \wedge p) \wedge (p \wedge p))$ . Como cada resposta está em uma linha, penso que dispensa os parênteses externo.

Na tela Ajuda:

Correções: acentuar a palavra “várias”, que encontra sem acento. Pontuar todas as telas.

Corrigir a palavra “calculadora” que aparece acentuada, e também a palavra “relatório”.

Quando pedi para gerar o meu relatório, todos os itens estavam em branco. Usei o Google Chrome, como navegador. Está bem lento para abrir os *pop-ups*.