

LOGOS - Caderno de Estudos e Exercícios de Lógica do ambiente de ensino Heráclito

Fabiane F. P. Galafassi¹, Cristiano Galafassi¹, João Carlos Gluz², Rosa Maria Vicari¹, Rafael Bueno²

¹ Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul (UFRGS), CEP: 91501-970, Porto Alegre, RS – Brasil

² Pós-Graduação em Computação Aplicada (PPGCA) – ISELab - Laboratório de Engenharia de Softwares Inteligentes – Universidade do Vale do Rio dos Sinos (UNISINOS) – Caixa Postal 275, CEP: 93.022-000, São Leopoldo, RS – Brasil

{fabiane.galafassi, cristianogalafassi¹}@gmail.com, jcgluz@unisinos.br, rosa@inf.ufrgs.br, rafaelbueno@outlook.com

Abstract. O presente artigo tem como objetivo apresentar o Caderno de Estudos e Exercícios de Lógica (LOGOS) do ambiente de ensino Heráclito. O ambiente Heráclito² auxilia estudantes a resolver vários tipos de exercícios de Lógica, começando com exercícios para calcular o valor lógico de uma fórmula, passando por exercícios de construção de tabelas-verdade e chegando até exercícios de elaboração de provas de argumentos por meio das regras da Dedução Natural. Desta forma, o ambiente disponibiliza o Caderno Eletrônico LOGOS que permite criar e editar fórmulas, tabelas-verdade e provas da Lógica Proposicional. O caderno LOGOS é compatível com *tablets*, *smartphones* e PCs e atualmente vem sendo utilizado com alunos de graduação dos 1º e 2º semestres em currículos das áreas científicas e tecnológicas.

1 Introdução

A disciplina de Lógica Formal é de suma importância para o currículo das áreas científicas e tecnológicas (genericamente agrupadas sob a sigla inglesa STEM – *Science, Technology, Engineering, and Mathematics*) sendo normalmente ofertada entre o 1º e o 2º semestre da graduação. Considerada fundamental na formação dos alunos, ela possibilita o desenvolvimento das habilidades de análise lógica, formalização e resolução de problemas. Habilidades essas, que por sua vez são necessárias para a compreensão dos diversos conteúdos e atividades encontradas neste currículo.

Estudos empíricos indicam sérias dificuldades encontradas por alunos em assimilar os conceitos dessa disciplina. Estes estudos apontam para índices altos em termos de reprovações e principalmente desistências, ocasionando uma retenção de alunos acima do esperado. Essas desistências, em particular, tendem a ocorrer no início da disciplina, principalmente quando os conteúdos de Dedução Natural começaram a ser abordados no contexto da Lógica Proposicional [Galafassi, 2012]. Na prática, as dificuldades começam quando conceitos como fórmula, regra de dedução e prova formal começam a ser apresentados. Objetivando contribuir na melhoria destes índices, a metodologia dialética de ensino (abordagem sócio histórica utilizada em sala de aula) foi associada a um modelo de mediação por computador, modelado em um STI, ao qual foi denominado de Ambiente Heráclito.

Nesse sentido o Caderno Eletrônico LOGOS disponibilizado pelo ambiente Heráclito é composto por dois editores: o Editor de fórmulas e tabelas-verdade e o Editor de Provas da Lógica Proposicional. Os editores contam ainda com o apoio de um tutor (*on-line*) neste processo de resolução de exercícios. O ambiente se destaca dos demais ambientes de ensino de Lógica em três aspectos fundamentais: a) uma completa aderência aos princípios Dialéticos e Sócio Históricos na implementação dos processos de mediação e tutoria, b) um modelo de aluno adaptativo e flexível e; c) um suporte completo ao ensino dos conceitos formais da Lógica Proposicional.

¹ Doutorandos no Programa de Pós-Graduação em Informática na Educação da UFRGS e professores na Universidade Federal do Pampa atuando no *campus* Itaqui/RS.

² Caderno Eletrônico LOGOS - <http://labsim.unipampa.edu.br:8080/heraclito/>.

2 Desenvolvimento

O ambiente Heráclito foi desenvolvido através do uso da tecnologia de agentes com fins pedagógicos. Estes agentes são responsáveis pela interação do aluno/tutor/ambiente auxiliando no desenvolvimento da resolução dos exercícios indicando caminhos corretos, incorretos e não recomendáveis durante o andamento da prova. O serviço de tutoria visa ajudar o aluno no passo a passo da resolução dos exercícios, fazendo o papel do professor. Toda interação do aluno/tutor/ambiente é monitorado e as ações registradas servem para determinar as ações do serviço de tutoria. Esse serviço usa um conjunto de estratégias de aprendizagem, elaboradas especificamente para Lógica, com base nas experiências em sala de aula. Se o aluno não conseguir avançar na realização de uma prova, por exemplo, ele tem a opção de utilizar o botão ajuda para pedir dicas, exemplos ou qual o próximo passo a ser dado. Esta opção Ajuda pode ser acionada a qualquer momento do exercício.

Todos os agentes do ambiente Heráclito foram desenvolvidos na linguagem Prolog e possuem uma interface em Java com suporte da plataforma JADE para comunicação entre eles. Estas linguagens e plataforma foram escolhidas por sua versatilidade, eficiência, portabilidade e segurança. Todo o processo de comunicação entre os agentes se dá por troca de mensagens na linguagem FIPA-ACL [FIPA, 2018], suportadas pela plataforma de agentes JADE [BELLIFEMINE; CAIRE; GREENWOOD, 2007], que é compatível com os padrões da FIPA [FIPA, 2018].

As especificações FIPA³ definem um modelo de referência para uma plataforma de agentes e também um conjunto de serviços fornecidos ao se conceber sistemas multiagentes interoperáveis, e o JADE (*Java Agent Development Framework*) é uma plataforma (em conformidade com as especificações da FIPA) para o desenvolvimento e execução de sistemas baseados em agentes de software que segue as recomendações da FIPA. Está escrito em Java, por questões de portabilidade, e distribuído sob a licença de software livre LGPL (*Lesser General Public License*).

A estrutura das mensagens trocadas na comunicação entre agentes é baseada na linguagem ACL (*Agent Communication Language*) definida pela FIPA e contém campos tais como, variáveis que indicam o contexto ao qual a mensagem se refere e, o tempo limite que pode ser aguardado até a resposta ser recebida (timeout), visando suporte a interações complexas e conversas paralelas múltiplas [BORDINI; HÜBNER; WOOLDRIDGE, 2007].

A comunicação entre agentes no JADE é a baseada em mensagens assíncronas, ou seja, um agente que deseja se comunicar deve apenas transmitir uma mensagem para um destino identificado (ou conjunto de destinos), não existindo nenhum tipo de dependência temporal entre o transmissor e o receptor.

A figura 1 apresenta a arquitetura do ambiente:

³ A FIPA (*Foundation for Intelligent Physical Agents*) é uma associação internacional composta por companhias e organizações que têm como objetivo compartilhar esforços com o intuito de produzir especificações para tecnologias de agentes genéricas. Foi criada em 1996 sem fins lucrativos para desenvolver uma coleção de normas relativas à tecnologia de agentes de software. Disponível em: < <http://www.fipa.org/> >.

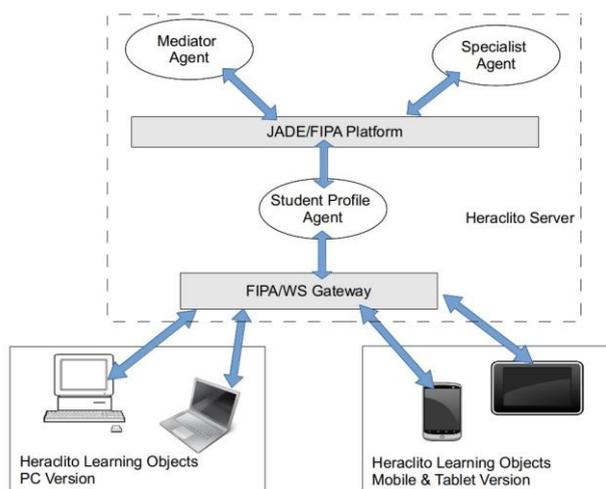


Figura 1 – Arquitetura do ambiente Heráclito.

Todas as mensagens trocadas entre esses três agentes passam obrigatoriamente pelo agente mediador, que é quem media esta comunicação entre o agente perfil do aluno e agente especialista. Nesta arquitetura todas as estratégias foram desenvolvidas com base no perfil do aluno e estão centradas no agente Mediador, o qual toma uma decisão baseado no contexto em que o aluno está inserido, fazendo uso das estratégias de aprendizagem e assim gerenciando as trocas de mensagens entre os agentes Perfil do Aluno e Especialista.

Ao desenvolver a interface *Web* do ambiente Heráclito optou-se pela arquitetura cliente-servidor. Essa escolha se deu em virtude da segurança e desempenho que essa metodologia, associada a linguagens de programação adequadas. Cabe ressaltar que, essa interface foi desenvolvida também com o objetivo de portabilidade, ou seja, ser portátil para diferentes dispositivos com diferentes capacidades, tamanhos de telas bem como desempenhos distintos. Com a utilização da arquitetura cliente-servidor, é possível enviar para o usuário apenas a resposta da requisição em *HTML 5*, *CSS* e *Javascript*, deixando toda a parte lógica do provedor e toda a parte de agentes sendo processado inteiramente dentro do servidor, o que torna o sistema extremamente leve e portátil para qualquer dispositivo que possuir um navegador *Web*. A figura 2 apresenta essa arquitetura.

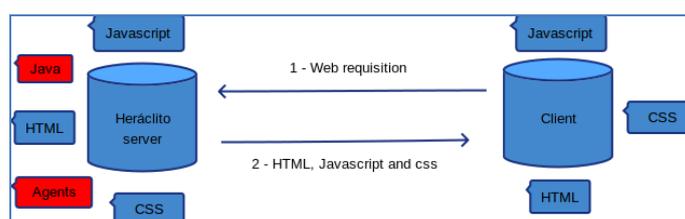


Figura 2 – Representação da arquitetura cliente-servidor do ambiente Heráclito.

Visando o aumento da segurança, robustez, compatibilidade com diversos sistemas e tamanhos de tela, foi utilizada para desenvolvimento o framework *Bootstrap*. Este framework fornece uma gama de elementos prontos em *HTML 5*, *CSS 3* e *JavaScript* que auxiliam na adaptação do site a diversos tamanhos de tela e diversos sistemas tornando-os responsivos, o que faz com que o objeto de aprendizagem possa, com apenas uma versão, ser distribuído para diversas plataformas diferentes alcançando um número maior de usuários.

3 Apresentação

O caderno Eletrônico LOGOS do ambiente Heráclito pode ser usado em modo *on-line*, com o apoio do serviço de tutoria, ou em modo *off-line*, como usuário visitante, mas sem a tutoria. Para o modo *on-line* é necessário fazer *login* ao ambiente. Na página inicial é possível realizar o cadastro e efetuar o *login* (autenticação) em poucos passos que são “nome”, “sobrenome”, “*e-mail*” e “senha”. Após o cadastro é necessário entrar com *e-mail* e senha. A interface inicial é apresentada na figura 3, bem como o cadastro e acesso para uso do ambiente e tutoria.

Figura 3 – Interface inicial do ambiente Heráclito

Após o *login* ou acesso como visitante ao ambiente, o caderno Eletrônico LOGOS e seus editores são apresentados. O caderno de estudos e exercícios de Lógica oferece ao usuário o Editor de fórmulas e tabelas-verdade e o Editor de Provas da Lógica Proposicional. Ver figura 4:

Figura 4 – LOGOS

O ambiente também disponibiliza o capítulo 1 e 2 em formato *E-book* de Introdução a Lógica Proposicional dos professores João Carlos Gluz⁴ e Monica Xavier Py⁵. Ver figura 5:

Figura 5 – Livro Eletrônico

⁴ Professor da Universidade do Vale do Rio do Sinos – UNISINOS. Computação Aplicada.

⁵ Professora da Universidade do Vale do Rio do Sinos – UNISINOS. Computação Aplicada.

O editor de fórmulas e tabelas-verdade objetiva compreender o conteúdo de lógica booleana e trabalhar com decomposição de equações booleanas em áreas e tabelas-verdade. Este editor oferece dois tipos de opções para resolução do exercício: Exercício Personalizado e Exemplos de Exercícios (figura 6):



Figura 6 – Interface Editor de Fórmulas e Tabelas-Verdade

No Exercício Personalizado o aluno pode inserir a fórmula desejada e escolher entre as três formas de resolução: Decomposição de Fórmula, Avaliação de Fórmula e Construção de Tabela-Verdade. Em Exemplos de Exercícios o aluno escolhe um exemplo de exercício já pré-definido da lista e em seguida a forma em como vai resolvê-lo (utilizando as três formas de resolução mencionadas no Exercício Personalizado).

O Editor de Fórmulas e Tabelas-Verdade ainda possui a opção Modo supervisionado. Nesta opção o próprio editor de tabela verdade ou o editor de decomposição de fórmulas, verifica se o aluno está executando corretamente os passos dos exercícios. Assim, por exemplo, se o supervisor está desabilitado um aluno pode preencher um valor-verdade errado em uma tabela-verdade, deveria ser verdadeiro e ele colocou falso. Com o supervisor desabilitado nada é indicado, mas se o supervisor está habilitado o valor verdade fica em vermelho, indicando que não é o valor correto. No guia de utilização do caderno LOGOS as três formas de resolução são detalhadas passo a passo (com esta opção habilitada) em seus exemplos de utilização.

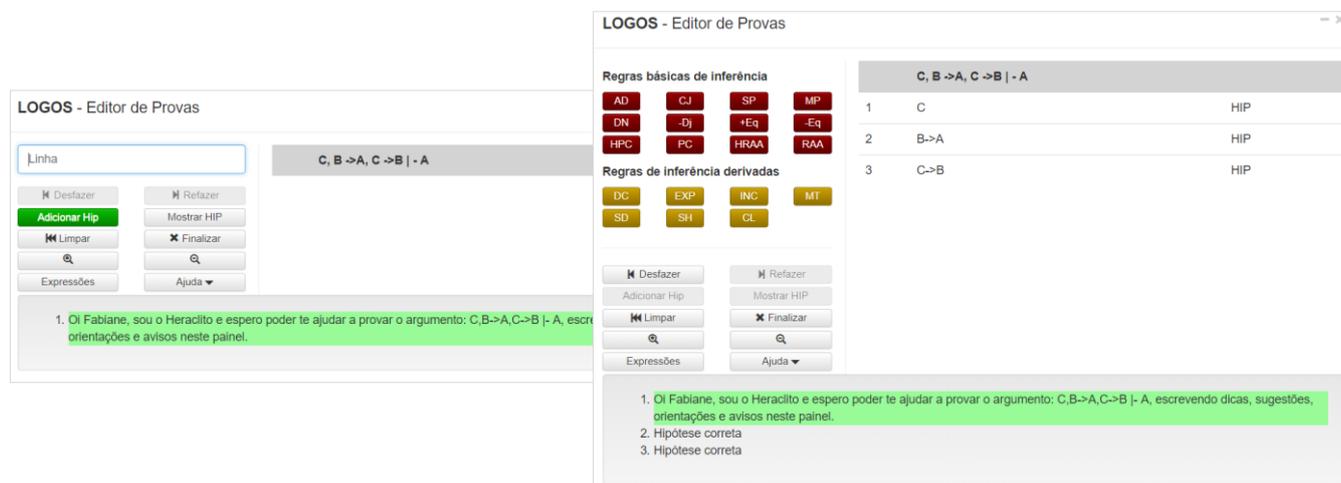
O Editor de Provas de Lógica Proposicional tem como principal funcionalidade auxiliar na elaboração de provas de argumentos formais por meio das regras da Dedução Natural (DNLP). A edição das provas oferece dois tipos de opções para resolução do exercício (ver figura 7).



Figura 7 – Interface Editor de Provas de Lógica Proposicional

Em Prova Personalizada o aluno pode inserir suas hipóteses e conclusão e iniciar o exercício.

Na interface do editor de provas o argumento deve ser digitado como mostra o exemplo acima HIPÓTESE1, HIPÓTESE 2, HIPÓTESE 3, ... HIPÓTESE N | - CONCLUSÃO no campo de entrada de dados para a nova prova e clique Começar. A seguir têm-se as figuras 8 e 9 que apresentam as inserções das hipóteses e da edição das regras de dedução.



Figuras 8 e 9 – Interfaces de inserção de hipóteses e edição de regras de dedução na Lógica Proposicional

A segunda opção é escolher um exemplo de exercício da lista com provas já pré-selecionadas que estão ordenadas por níveis de dificuldade: Provas Básicas, Provas Intermediárias e Provas Avançadas. As formas de aplicação e funcionamento de cada regra de inferência são melhores descritas e detalhadas no guia de utilização do caderno LOGOS. O guia também pode ser acessado através do ambiente quando logado.

3.1 Tutoria baseada na teoria Sócio Histórica

Desde a metodologia utilizada em sala de aula até a concepção do ambiente Heráclito, a aderência pelos princípios dialéticos e sócio históricos faz parte da pressuposição que cada exercício de Lógica se constitui em uma Cena de Atenção Conjunta (JAS, do inglês *Joint Attention Scene*) (TOMASELLO, 2001), onde o processo dialético de mediação entre o aluno e o ambiente, permite que este aluno evolua sua Zona de Desenvolvimento Proximal (ZDP) (VYGOTSKY, 1986). No caso do Heráclito há uma ampla liberdade de escolha do objeto a ser trabalhado na JAS: não há uma restrição de escolhas de exercícios particulares. Dentro das formas de exercícios suportadas pelo ambiente, alunos e professores podem escolher livremente quais exercícios serão trabalhados.

O ambiente assume que a mediação ocorre sob um processo de regulação proposto por Diaz et al. (1992) e adaptado por Passerino e Santarosa (2008) que começa a partir da categoria de controle e passa pela categoria de autocontrole com o objetivo de alcançar a categoria de autorregulação. A categoria de controle é identificada pela necessidade de ações mediadoras geradas pelo sujeito mais experiente e direcionadas ao sujeito menos experiente e tem duas dimensões: direta ou indireta. O controle direto é verificado através de pedidos, diretivas e questionamentos. O controle indireto pode ser identificado através de questões perceptuais, conceituais ou processuais. Isso culmina com o estabelecimento de uma distância física (o sujeito mais experiente deixa o menos experiente sozinho, apenas observando o processo) que se enquadra na categoria de autocontrole. Diaz et al. (1992) definem o autocontrole como a realização pelo sujeito de uma ação obedecendo a expectativa de um tutor internalizado. A autorregulação não pode ser observada diretamente porque é interna ao sujeito. No entanto, considera-se que o sujeito está nesta categoria quando organiza, planeja e executa ações sem a intervenção do mediador externo. Assim, a autorregulação é alcançada quando o sujeito pode conceber um plano de ação e segui-lo como se fosse seu próprio tutor.

Aplicando este modelo de regulação ao processo de tutoria, o ambiente Heráclito assume que o aluno, como sujeito menos experiente, começa a partir de um estado onde há necessidade de intervenções pedagógicas correspondentes ao controle externo direto, passando a um estado onde as intervenções visando o controle indireto se tornam mais eficazes e gradualmente passando a ter autonomia para ser capaz de fazer ações autocontroladas, onde as ações pedagógicas se tornam mais reflexivas. O ambiente Heráclito suporta vários tipos de intervenção pedagógica, incluindo ações de supervisão, orientação, confirmação, suporte, contestação e reflexão. As intervenções de supervisão, orientação e confirmação, são classificadas como controle direto, enquanto confirmação, suporte e contestação são formas de controle indireto. As ações de reflexão visam ajudar na evolução cognitiva do aluno do autocontrole para a categoria de autorregulação.

Para cada exercício (objeto da JAS), o ambiente oferece um serviço de supervisão elementar de erros que detecta e avisa sobre erros básicos na resolução do exercício (por exemplo, erros na decomposição da fórmula, no preenchimento de um valor da tabela-verdade, ou na aplicação de uma regra de inferência). Em concomitante também é fornecido um serviço de tutoria inteligente capaz de diagnosticar e agir sobre situações críticas nas categorias de controle direto e indireto. Assim são detectadas situações que requerem intervenções de orientação (ajuda), confirmação (incentivo), suporte (dicas e indicações), contestação (avisos e alertas de possíveis problemas) e chegando a possibilidade do ambiente sugerir processos de reflexão, através de exemplos e referências de leitura, que ajudam o aluno a avançar para a autorregulação.

4 Potencial de aplicação/Resultados esperados e ou alcançados

O ambiente Heráclito apresentado neste artigo visa ajudar alunos a aprender conceitos da Lógica Proposicional. Experimentos empíricos conduzidos com este ambiente mostram que a possibilidade de ajuda é real e que, além disso, o ambiente está sendo bem aceito pelos alunos. É claro que isso não sinaliza o fim da pesquisa, muito antes pelo contrário, abre todo um novo conjunto de possibilidades de pesquisa que se espera poder conduzir com este ambiente. Mas antes disso se espera poder contribuir com o avanço no ensino da Lógica pela disponibilização do ambiente Heráclito à comunidade. Assim, após os resultados dos experimentos, o ambiente entrou em uso experimental e pode ser acessado e usado por qualquer estudante no endereço <http://labsim.unipampa.edu.br:8080/heraclito/>.

- Os primeiros experimentos realizados com o ambiente podem ser encontrados em: <http://www.repositorio.jesuita.org.br/handle/UNISINOS/4257>.
- Resultados mais recentes podem ser encontrados em <http://www.br-ie.org/pub/index.php/sbie/article/view/7639/5435>.
- Dados obtidos em 2017 ainda estão em processo de análise e publicação;

5 Considerações Finais e Trabalhos Futuros

Ao desenvolver e adaptar o ambiente Heráclito esperou-se poder contribuir, não só em um melhor entendimento dos conteúdos abordados na disciplina de Lógica, mas também diminuir o número de desistências e reprovações existentes neste contexto.

Em termos de novas pesquisas várias possibilidades já estão sendo estudadas, sendo uma das mais importantes à evolução do seu modelo de aluno. Atualmente o modelo de aluno do Heráclito está centrado nas várias formas possíveis de exercícios da Lógica Proposicional, faltando uma visão geral do processo de aprendizagem do aluno em relação aos conceitos por trás de cada tipo de exercícios e da posição destes conceitos na estrutura curricular da disciplina.

Desta forma, um modelo bayesiano de inferência está em fase de implementação e testes, o qual se espera então ser capaz de representar o conhecimento do aluno acerca dos conceitos trabalhados em DNLP. Este modelo busca compreender melhor os processos que ocorrem durante as interações entre aluno/tutor/ambiente. Desta forma, estima-se que este novo mecanismo de inferência possa auxiliar na identificação de indícios que apontem o nível de conhecimento real do aluno (ou seja, informações a acerca do que o aluno conhece) e que as relações topológicas da rede possam expressar a ZDP deste aluno. Ainda de posse destas informações, futuramente, será desenvolvido um mecanismo de estratégias de ensino-aprendizagem específicas para cada perfil de aluno encontrado.

Referencias

AMBIENTE HERÁCLITO. Disponível em: <<http://labsim.unipampa.edu.br:8080/heraclito/>>. Acesso em 04 junho de 2018.

BELLIFEMINE, Fábio; CAIRE, Giovanni; GREENWOOD, Dominic. Developing Multi-Agent Systems with JADE. Wiley Series in Agent Technology, Series Editor: Michael Wooldridge, Liverpool University, UK. 2007.

BORDINI, R. H.; HÜBNER, J. F.; WOOLDRIDGE, M. Programming multi-agent systems in AgentSpeak using Jason. Volume 8. Editora John Wiley & Sons, 2007.

DIAZ, R., NEAL, C., AMAYA-WILLIAMS, M. (1992) The Social Origin of Self-regulation. In Moll, L.C (Comp.), Vygotsky and Education: Instructional Implications and Applications of Sociocultural Psychology. Cambridge University Press.

FIPA-ACL. FIPA ACL Message Structure Specification. FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS. Disponível em: <<http://www.fipa.org/specs/fipa00061/SC00061G.html>>. Acesso em 04 junho de 2018.

GALAFASSI, F.F.P.: Agente Pedagógico para Mediação do Processo de Ensino-Aprendizagem da Dedução Natural na Lógica Proposicional. Dissertação de Mestrado. UNISINOS. 2012. Disponível em: <<http://biblioteca.asav.org.br/vinculos/000003/00000335.pdf>>. Acesso em 04 junho de 2018.

GALAFASSI, P. F. F.; SANTOS, A. V.; PERES, R. K.; VICARI, R. M.; GLUZ, J. C. Multi-plataform Interface to an ITS of Proposicional Logic Teaching. In: 13 ° International Conference on Practical Applications of Agents and Multi-Agent Systems, 2015, Salamanca. Highlights of Practical Applications of Agents, Multi-Agent Systems, and Sustainability - The PAAMS Collection, 2015. v. 524. p. 309-319.

GLUZ, J. C; BUENO, R.; PERES, R. K.; GALAFASSI, P. F. F. Tutoria Inteligente Completa para os Conceitos Formais da Lógica Proposicional: Experimentos e Resultados. VI Congresso Brasileiro de Informática na Educação (CBIE 2017). Anais do XXVIII Simpósio Brasileiro de Informática na Educação (SBIE 2017). Disponível em: <<http://www.br-ie.org/pub/index.php/sbie/article/view/7639/5435>>. Acesso em 04 junho de 2018.

GLUZ, J.C.; PY, M. (2010) Lógica para Computação. Coleção EAD. Editora Unisinos.

ISELab - Laboratório de Engenharia de Softwares Inteligentes. Projeto Heráclito: Ambiente Inteligente de Ensino de Lógica. Disponível em: <<http://obaa.unisinos.br/drupal7/?q=node/43>>. Acesso em 04 junho de 2018.

PASSERINO, L. M., SANTAROSA, L. (2008) Autism and Digital Learning Environments: processes of interaction and mediation. Computers and Education, v.51, pg. 385-402.

TOMASELLO, M. (2001). The Cultural Origins of Human Cognition. Harvard Univ. Press.

VYGOTSKY, L. S. (1986). Thought and Language. The M.I.T. Press, Cambridge, MA.