

## **LpT: ferramenta educacional para auxiliar o ensino/aprendizagem de traduções de diferentes níveis de linguagens de programação**

**Costa, D.B.B.<sup>1</sup>, Teixeira, D. L. S.<sup>1</sup>, Grisotto, R, S.<sup>1</sup>, Inocêncio, A.C.G.<sup>1</sup>, Rocha, B. M.<sup>1</sup>**

<sup>1</sup>Instituto de Ciência Exatas e Tecnológicas (ICET)

Universidade Federal de Goiás (UFG)

Caixa Postal 03-CEP:75801-615-91.501-970 Jataí-GO-Brazil

{diegocostacmp,ufgdaniel,grisotto18,anacarolina.inocencio,rocha3runo}@gmail.com

**Abstract.** *This article discusses the development of an educational tool to aid in the content abstraction on the translation of different levels of programming languages of the Computer Architecture discipline in the Computer Science, computer engineering, and other courses. The developed software aims to help the teaching and learning of the concepts of a high-level programming language translated into other levels (medium and low). To demonstrate some criteria inherent to the software efficiency it was carried out an experiment with Computer Architecture discipline students from the Computer Science course to clarify the level of usability and usefulness of the educational tool. Subsequently a questionnaire was applied to check the quality of student learning and the tool effectiveness. The tool qualitative evaluation results showed that the construct “Use Utility” obtained greater acceptance when compared with the “Ease of Use”, thus highlighting the ease of learning and the students acceptance of the tool. And thus, it can be a great tool to help teachers in teaching and student learning process.*

**Resumo.** *Este artigo aborda o desenvolvimento de uma ferramenta educacional para o auxílio na abstração do conteúdo acerca da tradução de diferentes níveis de linguagens de programação da disciplina de Arquitetura de Computadores do curso de Ciência da Computação, engenharia da computação e afins. O software desenvolvido tem como objetivo auxiliar o ensino e aprendizado dos conceitos de uma linguagem de alto nível traduzida em outros níveis (médio e baixo). Para demonstrar alguns critérios inerentes à eficiência do software foi realizado um experimento com os alunos da disciplina de Arquitetura de Computadores do curso de Ciência da Computação, que visava elucidar o nível de usabilidade e utilidade da ferramenta educacional. Posteriormente foi aplicado um questionário para verificar a qualidade do aprendizado dos alunos e a eficácia da ferramenta. Os resultados da avaliação qualitativa da ferramenta mostraram que o constructo “Utilidade de Uso” obteve maior aceitação quando comparado com a “Facilidade de Uso”, ressaltando assim a facilidade de aprendizado e a aceitação da ferramenta por parte dos alunos. Podendo assim ser uma ferramenta de grande ajuda para o professor no processo de ensino e aprendizagem dos alunos.*

## 1. Introdução

A disciplina de Arquitetura de Computadores (ACs) tem por objetivo apresentar ao aluno o computador como um componente eletrônico, mostrando o seu funcionamento no nível de linguagem de máquina e linguagem *Assembly*. Permitindo ao aluno compreender todos os níveis de abstração que se fazem presentes em um ambiente computacional. A ACs trata principalmente do comportamento funcional de um sistema computacional, onde o programador se comunica com o computador por meio de uma linguagem de programação. Desta forma é necessário um mecanismo de tradução das linguagens de alto nível (de fácil entendimento para o usuário), para as linguagens de mais baixo nível, sendo a linguagem binária o menor nível, que representa o código de máquina que uma arquitetura específica utiliza. Estas disciplinas possuem um conteúdo altamente complexo e de grande embasamento teórico, sendo que tais disciplinas podem ser realizadas em estágios sequenciais e evolutivos (PENNA; FREITAS, 2013). Para Torres e Brito (2012), Ullmann et al. (2014), Zeferino et al. (2012) o alto grau de complexidade da disciplina de ACs dificulta o nível de compreensão dos alunos, levando a uma deficiência de aprendizagem que se propaga durante todo o curso.

Independente do método de ensino utilizado pelo docente, o processo de compreensão nem sempre é fácil, pois são apresentados ao discente uma grande gama de detalhes sobre o conteúdo, que nem sempre são intuitivos. O modelo tradicional de aulas baseado em aulas práticas, teóricas e exercícios, nem sempre é interessante quando se trata de abstração e consolidação do conhecimento, o que pode tornar o processo de aprendizado muito mais doloroso, desmotivando o aluno e o distanciando do objeto de estudo (MARCONDES et al., 2016; SOBREIRA et al., 2007; MORANDI; RAABE; ZEFERINO, 2006). Em Felix, Pousa e Carvalho (2006) afirma-se que existe a necessidade de utilizar outros mecanismos que facilitem e evidenciem a importância da ACs, afim de tornar a disciplina mais atrativa e menos complexa para os alunos.

Neste aspecto podemos contar com a ajuda das ferramentas educacionais (FEs) para auxiliar o aluno na compreensão do conteúdo. Para Ullmann et al. (2014) as FEs digitais têm sido desenvolvidas para auxiliar o ensino em várias áreas da computação e outras ciências. Tais ferramentas tornaram-se cada vez mais sofisticadas e intuitivas, visando apoiar o ensino em disciplinas cujo conteúdo possui natureza abstrata e dinâmica. Desta forma, segundo Rodrigues e Martins (2008), no contexto da disciplina de ACs, o processo de ensino e aprendizagem pode ser facilitado com a utilização desse tipo de ferramenta.

Portanto, o objetivo geral deste trabalho é fornecer ao professor uma ferramenta de auxílio para o ensino da disciplina de ACs, no que diz respeito ao conteúdo inerente as linguagens de programação em diferentes níveis, facilitando assim o entendimento e o aprendizado do aluno. Por meio da ferramenta o aluno poderá alcançar maior êxito na disciplina cursada, além disso, sobretudo beneficiará a comunidade acadêmica que esteja interessada em aprender sobre o conteúdo em questão.

A realização de pesquisas sobre o ensino de ACs é de grande relevância. No Brasil, inclusive, há um evento específico sobre esse tema de pesquisa, o *Workshop* sobre Educação em Arquitetura de Computadores (WEAC),<sup>1</sup> que ocorre em parceria com o

<sup>1</sup><http://www2.sbc.org.br/sbac/2015/weac.html>

*International Symposium on Computer Architecture and High Performance Computing*. Além disso, na literatura, contamos com diversas ferramentas educacionais cujo objetivo é permitir ao aluno visualizar e entender melhor os conceitos relacionados a essa matéria. Apesar disso, ainda existe muita carência de ferramentas para o auxílio no entendimento de conceitos específicos da disciplina. Como exemplo, Torres e Brito (2012) cria uma extensão chamada *Ptolemy*, desenvolvida numa ferramenta de modelagem e simulação de sistemas concorrentes para contribuir no ensino e aprendizagem de Arquitetura. Já Machado e Maia (2004) utiliza o *SOSim* que é um *framework* de apoio educacional que permite facilitar o processo de ensino e aprendizagem dos conceitos e técnicas envolvidos em um sistema operacional moderno, tornando o processo mais eficiente, simples e agradável.

Neste sentido, para auxiliar no processo de aprendizagem da disciplina elucidada foi desenvolvido uma FE chamada LpT (tradutor de linguagem de programação) que auxiliará o aprendiz na compreensão do conteúdo referente a conversão de linguagens de alto nível (que se aproxima da comunicação humana) para linguagens de baixo nível (linguagem de máquina), tornando o aprendizado mais interessante. Além disso, permite um respaldo maior para a comunidade acadêmica, pois apesar de existirem várias ferramentas para simulação (REIS; COSTA, 2009; ISOTANI et al., 2009), não foi encontrada nenhuma que aborde a transformação da linguagem de programação de um nível (mais alto) para outro (nível mais baixo).

Por fim, este artigo está organizado da seguinte forma: na Seção 2 está o referencial teórico, que contém alguns conceitos sobre este trabalho. Na seção 3 é mostrado os trabalhos relacionados que expõe as ferramentas educacionais de apoio ao ensino de ACs e Sistemas Operacionais (SOs). Na Seção 4 é descrito os aspectos inerentes à interface gráfica da ferramenta; já na subseção 4.1 está disposto a arquitetura, caso de uso e principais funções. Já na Seção 5 é apresentada a avaliação do LpT e, por fim, na Seção 6 estão as considerações finais e trabalhos futuros.

## **2. Referencial Teórico**

No sentido de esclarecer o que compõe a disciplina de Arquitetura de Computadores, Tanenbaum (2007) atesta que o objetivo da matéria é apresentar ao aluno o computador como um componente eletrônico, mostrando o seu funcionamento no nível de linguagem de máquina e linguagem *Assembly*. Isto permite ao aluno compreender melhor todos os níveis de abstração, que é representado pelas diferentes linguagens de programação. Para Torres e Brito (2012) o conhecimento sobre as linguagens de programação permite que o indivíduo consiga desenvolver uma série de conjunto de competências necessárias para conceber programas e sistemas computacionais capazes de resolver problemas reais, ressaltando ainda a necessidade de compreender e aplicar certos conceitos abstratos de programação.

Sendo assim, as linguagens de programação podem ser classificadas quanto a sua proximidade com a linguagem de máquina. Quanto maior a semelhança com a linguagem de máquina, mais baixo é o nível da linguagem. As linguagens de programação mais semelhantes à linguagem de máquina são conhecidas como linguagens de baixo nível. Analogamente, linguagens de programação “distantes” da linguagem de máquina são conhecidas como linguagens de programação de alto nível (NOSCHANG; PELZ; RAABE, 2014; FERRANDIN; STEPHANI, 2012).

Dentre as linguagens que podem representar instruções primitivas de baixo nível, podemos ressaltar as linguagens *Assembly*, que compreendem as linguagens simbólicas de montagem, projetadas para minimizar as dificuldades da programação em notação binária. Nas linguagens de alto nível, a linguagem Java ganha destaque por ser amplamente utilizada, por ter a característica de ser multiplataforma e entre outros (SCHILDT; SKRIEN, 2013). Sendo assim, é possível desenvolver uma série de aplicações para diversos segmentos, dentre elas as ferramentas educacionais. Estas FE podem ser desenvolvidos tanto para o contexto de ensino e aprendizado quanto para outras finalidades, sendo que estes têm como objetivo favorecer os processos de ensino e aprendizagem; são desenvolvidos especialmente para construir o conhecimento relativo a um conteúdo didático. A característica principal de uma FE é o seu caráter didático, que possibilita a construção do conhecimento em uma determinada área, com ou sem a mediação de um professor (JUCÁ, 2011).

### 3. Trabalhos Relacionados

As FEs são instrumentos tecnológicos que podem auxiliar no entendimento de muitas áreas. No contexto utilizado neste trabalho as FEs auxiliam os professores na abordagem do tema estudado, garantindo melhor interação do aluno e compreensão do tema abordado. As ferramentas educacionais constituam uma das maiores fontes de mudança do ensino e do processo de manipulação da informação, o que torna o processo educacional muito mais eficiente e interessante no aspecto pedagógico (FRÓES; JUNIOR, 2016). Foram encontradas muitas ferramentas na Internet para simulação de conceitos e técnicas em Sistemas Operacionais e Arquitetura de Computadores, visando maior eficácia no ensino. Entretanto estas ferramentas apenas simulam de forma gráfica aspectos inerentes aos componentes eletrônicos, projeto de circuitos e características básicas da arquitetura.

Podemos citar algumas ferramentas voltadas para o campo da informática educativa, como por exemplo, o uso de simuladores para ensino de Arquitetura e Sistemas Operacionais, que são: NeanderSIM, IOsimulator e ferramenta de apoio a SOs, respectivamente:

1. *NeanderSIM* (ULLMANN et al., 2014): um simulador gráfico para apoiar o ensino de Arquitetura de Computadores, que apresenta, de forma interativa e animada, o processo de execução de instruções em um computador.
2. *IOsimulator* (MEDEIROS et al., 2011): Este trabalho apresenta o *IO Simulator*, um ambiente de simulação que mostra de maneira prática os conceitos teóricos introduzidos pelo professor em sala de aula, exibindo detalhadamente os processos e peculiaridades do comportamento do sistema de gerenciamento de dispositivos de entrada e saída.
3. Ferramenta de apoio a SOs (ISOTANI et al., 2009): relata uma ferramenta para interação virtual por meio do gerenciamento de processos, gerando simulações com algoritmos específicos de escalonamento e apresenta os resultados através de interfaces gráficas amigáveis para os usuários.

O trabalho LpT é um tradutor de linguagens diferente dos demais simuladores e programas encontrado durante a pesquisa, isso se deve a alguns atributos do sistema como usabilidade, que foi avaliada e testada durante as interações com aluno e professor. O LpT possui características singulares, tais como recursos visuais, facilidade de uso e a visualização das entradas no programa em diferentes níveis de linguagens, de forma simultânea.

## 4. Interface Gráfica

O tradutor de linguagens LpT foi desenvolvido com a utilização da linguagem de programação *Java*, utilizando as bibliotecas AWT e Swing (CASTRO; ALVES, 2013). A tela inicial do LpT é mostrada na figura 1, onde o usuário pode visualizar a tela principal e as opções de “Funcionamento”, “Lista de Comandos” (Figura 1 - direita), “Sair” e um ícone para a avaliação da ferramenta.

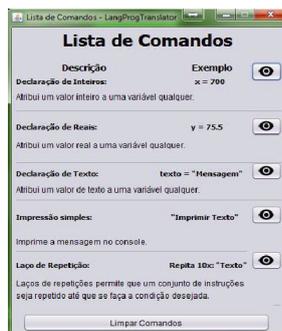


Figura 1. Lista exemplificando como deve ser a descrição de entrada em cada nível.



Figura 2. Exemplo de conversão de Linguagem Natural para linguagens de mais baixo nível.

A opção de “Funcionamento” disposto na parte superior da interface do programa (Figura 2) é responsável por informar: a finalidade do programa, funções básicas do programa e como é representado sintaticamente o formato de cada linguagem de programação, em cada nível de abstração durante o processo de tradução da linguagem. Para demonstrar a finalidade da ferramenta é disponibilizado um pequeno tutorial que contém conceitos essenciais sobre as linguagens de alto e baixo nível, visando esclarecer o entendimento sobre abstração de linguagens. O campo “Linguagem natural” (Figura 2) permite que o usuário digite um entrada em linguagem natural (exemplo, laço de repetição, condicional, atribuição, etc) que será traduzida simultaneamente nos demais campos (Código em JAVA, código em Fortran e Linguagem de Máquina), respeitando a sintaxe apropriada de cada linguagem.

Na parte de representação sintática da ferramenta (Figura 1) contamos com a opção “Lista de comandos” que define como deve ser a sintaxe das entradas no programa e um exemplo de cada operação, que pode ser apagado a qualquer momento da execução através do botão “Limpar Comandos”. As operações são: declaração de inteiros, declaração de reais, declaração de texto, impressão simples e laço de repetição. Após o usuário digitar a entrada desejada, de acordo com a lista de comandos, ele deverá pressionar a tecla ENTER, assim ele poderá verificar o resultado da tradução em outras linguagens. Caso o aprendiz não digite a entrada da forma correta não será mostrado a tradução para os outros níveis de abstração, sendo necessário reformular o texto da entrada na Linguagem Natural.

### 4.1. Arquitetura do LpT

A implementação foi feita na linguagem Java, onde foi possível fazer a manipulação dos dados e resolução dos problemas. O ambiente de desenvolvimento NetBeans IDE 8.0.2 (*Integrated Development Environment*) foi escolhido para auxiliar na criação da FE. A arquitetura da ferramenta LpT foi projetada em basicamente dois módulos. Os módulos

criados são (Figura 2): (1) módulo de interface gráfica, (2) módulo de tradução e caso de uso da ferramenta.

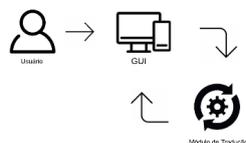


Figura 3. Arquitetura do LpT.

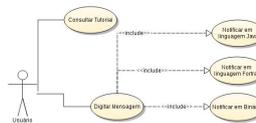


Figura 4. Caso de uso do LpT.

No módulo de interface (GUI) o usuário poderá fazer a sua interação com o sistema (Figura 3), construir suas próprias mensagens e traduzi-las no sistema. A tradução (transformar linguagem de alto nível para baixo nível) do LpT foi feita na linguagem Java. Neste módulo (módulo de tradução) contamos com a ajuda do padrão *observer*<sup>2</sup>, que notifica todas as demais classes de tradução quando ele detecta uma mudança na mensagem de entrada, ou seja, o comportamento de um determinado objeto depende de um estado. O módulo de tradução transforma a linguagem de alto nível para outras três linguagens, que são: linguagem Java, Fortran e binária. Estas linguagens traduzidas - Java, Fortran e binária - representam alto, médio e baixo nível de linguagem de programação, respectivamente.

O diagrama de caso de uso (Figura 4) documenta as funcionalidades do sistema do ponto de vista do usuário (PINTO, 2013), podemos notar que o ator fica limitado basicamente a: fazer uma consulta no tutorial, ou a digitar uma mensagem. Repare que quando o ator interage com o caso de uso “Digitar Mensagem” e pressiona a tecla ENTER, posteriormente ocorre três eventos, estes eventos são notificados pelo padrão *observer*, sendo eles: notificar mensagem em Java, Fortran e em binário, que são fundamentais para o comportamento e a interação do usuário, pois eles é quem irão fazer a tradução para as suas respectivas linguagens.

## 5. Experimento e Avaliação

Para a avaliação da ferramenta LpT foi utilizado o Modelo de Aceitação de Tecnologia (TAM), desenvolvido por Davis, Bagozzi e Warshaw (1989), que é uma adaptação do modelo da teoria da ação racionalizada (Fishbein & Ajzen, 1975), especificamente aplicado para os usuários de computadores ou de sistemas de informação. O objetivo do modelo é fornecer explicação das causas determinantes da aceitação dos computadores, de forma geral, também visa descrever o comportamento dos usuários em face das diversas tecnologias ligadas à informática. Além disso, explica aos interessados por que um *software*, em especial, pode ser aceito ou rejeitado, fornecendo instruções para correções e melhorias. Desta forma, autores como Jr, Cavazotte e Alves (2011) relatam sobre a utilização do modelo TAM para verificar a relação de aceitação e satisfação quanto ao aprendizado através de plataformas tecnológicas que visam promover benefícios ao processo educacional no ensino à distância.

O TAM funciona sob duas perspectivas:

<sup>2</sup>O *Observer* é um padrão de projeto de *software* que define uma dependência um-para-muitos entre objetos de modo que quando um objeto muda o estado, todos seus dependentes são notificados e atualizados automaticamente (GAMMA, 2009).

1. **Utilidade de uso:** trata da verificação de desempenho sobre o objeto de uso, ou seja, desempenho proporcionado ao indivíduo.
2. **Facilidade de uso:** refere-se a isenção de esforço físico ou mental para o uso do *software* ou tecnologia, isto é, a facilidade de uso proporcionada pela ferramenta.

Para isso, o modelo TAM sugere a criação de questionários, aos quais, deve ser adicionados as questões inerentes a facilidade de uso e desempenho proporcionado. Para cada afirmação, o questionado deverá optar por uma das seguintes opções: “Discordo Totalmente”, “Discordo Fortemente”, “Discordo Parcialmente”, “Neutro”, “Concordo Parcialmente”, “Concordo Fortemente” e “Concordo Totalmente”, conforme sua opinião. O questionário (Figura 5) que foi desenvolvido contém dezoito afirmações (nove relacionadas a facilidade de uso e nove relacionadas com utilidade da ferramenta).

Afirmações relacionadas à facilidade de uso		Afirmações relacionadas à utilidade percebida	
1	Os recursos de navegação do Lpt estão todos claros e fáceis de achar.	10	Utilizar a LpT é importante e adiciona valor ao meu trabalho/estudo.
2	Eu gosto de trabalhar com a LpT.	11	A LpT é útil no processo de ensino/aprendizado dos conceitos de Linguagem de alto nível e linguagem de baixo nível.
3	O acesso à LpT é simples.	12	Usar a LpT pode aumentar meu desempenho durante os estudos ou ensino da disciplina Arquitetura de Computadores.
4	Usar a LpT é uma boa ideia.	13	A LpT pode facilitar a realização do meu estudo/trabalho.
5	Na LpT eu sempre sei onde estou e como chegar aonde quero chegar.	14	A LpT produz os resultados que espero de uma ferramenta de suporte ao ensino Arquitetura de Computadores.
6	Minha interação com a LpT é clara e compreensível.	15	Eu tenho a intenção de utilizar a LpT ao longo dos próximos semestres.
7	Na LpT é fácil encontrar a informação que desejo.	16	Eu pretendo integrar a LpT à minha rotina de trabalho/estudo.
8	A LpT possui visual/interface atraente.	17	Eu recomendaria o uso da LpT.
9	Mesmo antes de clicar em um botão na LpT eu já sei a ação dele.	18	Todos os conceitos das linguagens na LpT foram abordados de forma correta.

Figura 5. Quadro de questões relacionadas a facilidade e utilidade de uso.

A avaliação do LpT foi realizada com 33 graduandos do curso de Ciência da Computação, que cursavam a disciplina de Arquitetura de Computadores, onde cerca de 90% dos alunos estavam fazendo a disciplina pela primeira vez. No primeiro momento foi feita uma explicação sobre como funcionava a ferramenta LpT. Após a explicação os alunos testaram as funcionalidades da FE, caso algum aluno ainda estivesse com dúvida eles podiam recorrer a Lista de Comandos do LpT, que disponibilizava um exemplo de entrada (Linguagem natural, Java, Fortran e Binário) em cada nível de linguagem, bem como sua tradução simultânea.

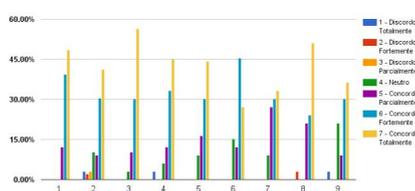


Figura 6. Facilidade de uso.

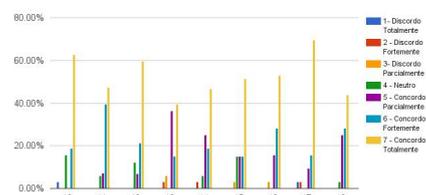


Figura 7. Utilidade de uso.

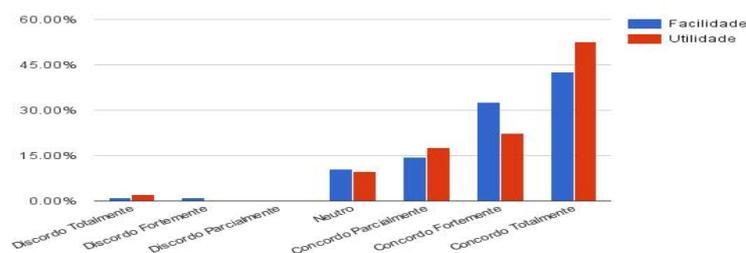
Na avaliação de facilidade de uso (Figura 6) pode-se notar que para as questões de 1 a 9, a maioria dos usuários escolheram entre as opções 6 e 7 (de acordo com a escala *Likert*<sup>3</sup>), que significa respectivamente: “Concordo Fortemente” e “Concordo Totalmente”

<sup>3</sup>Escala que permite mensurar as atitudes e o grau de conformidade do entrevistado (JÚNIOR; COSTA, 2014).

(o que girou em torno de 35% e 45%). As demais opções, foram escolhidas em média de 12% dos resultados. A afirmação com menor pontuação foi a 7 (“Na LpT é fácil encontrar a informação que desejo”). Este resultado se deve ao fato de que os alunos não estavam familiarizados com a ferramenta educacional LpT, entretanto, ainda houve uma boa aceitação por parte dos usuários, totalizando cerca de 62% (referente a “Concordo Totalmente” e “Concordo Fortemente”). Por fim, a questão que recebeu melhor pontuação foi a 3 (“O acesso à LpT é simples”) com um total de 87%, subdividido entre “Concordo Totalmente” e “Concordo Fortemente”, evidenciado assim a simplicidade oferecida pela ferramenta durante sua utilização.

Já na figura 7, que diz respeito a avaliação sobre “Utilidade de Uso”, das 9 afirmações levantadas percebe-se que todas tiveram resultados positivos na opção 7 (“Concordo Totalmente”), com média de 55% de votos. Nesta segunda parte, a questão que atingiu maior índice de votos foi a 17 (“Eu recomendarei o uso da LpT”), com cerca de 70% para a opção 7 (“Concordo Totalmente”). Em contrapartida não houve grandes índices para a opção 6 (“Concordo Fortemente”) nas demais perguntas, ficando acentuado em média de 22,2%.

O resultado satisfatório para a opção 7 (“Concordo Totalmente”) se dá pelo fato de que a maioria dos alunos que participaram da aplicação do questionário estavam cursando a disciplina de ACs. Há também uma minoria que “Discorda Fortemente”(2%) ou “Discorda Totalmente”(1%), o que indica que a pessoa não tem interesse de utilizar a ferramenta nos próximos semestres. Isso pode ser pelo fato do aluno não precisar refazer ou continuar a disciplina de ACs, em sua grade curricular. Além disso, o aprendiz pode não conhecer a ementa das disciplinas futuras que serão cursadas, que podem está relacionadas ao conceito de tradução de diferentes níveis de linguagens de programação.



**Figura 8. Tabela de questões relacionadas a facilidade e utilidade de uso.**

Por fim, a figura 8 apresenta um comparativo entre “Facilidade de Uso” e “Utilidade de Uso”, de acordo com a escala *Likert*. A figura mostra que, respectivamente, aproximadamente 45% e 35% das respostas para “Facilidade de Uso” foram “Concordo Totalmente” e “Concordo Fortemente”. Já para o constructo “Utilidade de Uso” os resultados foram 55% e 25% para “Concordo Totalmente” e “Concordo Fortemente”, na devida ordem. Desta forma, observa-se que a ferramenta LpT obteve resultados mais acentuados para “Facilidade de Uso”, enquanto que na “Utilidade de Uso” os resultados foram em sua maioria para a opção “Concordo Totalmente”. Sendo assim percebe-se que a ferramenta LpT pode auxiliar o aprendiz na compreensão do conteúdo referente a tradução de linguagens de alto nível para linguagens de baixo nível na disciplina de ACs, de acordo

com os resultados obtidos.

## 6. Conclusão e Trabalhos Futuros

O alto grau de complexidade da disciplina de Arquitetura de Computadores evidencia a necessidade de novas metodologias de ensino que possam auxiliar no processo de ensino e aprendizado. Neste trabalho apresentou-se uma ferramenta educacional cujo intuito é facilitar no processo de abstração de tradução de diferentes níveis de linguagem, ou seja, tradução de linguagem de alto nível para linguagem de baixo nível, denominada LpT. Tal ferramenta foi desenvolvida utilizando a linguagem Java, o que facilitou na elaboração de suas interfaces e funções. Também houve a avaliação da ferramenta com os graduandos do curso de Ciência da Computação, onde os mesmos concordam quanto a utilidade e facilidade de uso da ferramenta, evidenciando sua possibilidade de uso para auxiliar no aprendizado de diferentes níveis de linguagem.

Como trabalhos futuros, tem-se a intenção de implementar a forma como ocorre o processo de conversão de linguagens (em alto e baixo nível), assim o usuário poderá visualizar como funciona o processo de conversão de operações aritméticas em baixo nível, sendo que algumas das operações de linguagem *Assembly* que pretende-se efetivar é: AND, OR, OUEXCLUSIVO e ADD.

## Referências

- CASTRO, L. F. S. de; ALVES, G. V. Análise e comparação de frameworks para edição e visualização de grafos. *Anais SULCOMP*, v. 6, 2013.
- DAVIS, F. D.; BAGOZZI, R. P.; WARSHAW, P. R. User acceptance of computer technology: a comparison of two theoretical models. *Management science*, INFORMS, v. 35, n. 8, p. 982–1003, 1989.
- FELIX, A.; POUSA, C.; CARVALHO, M. Dimipss: Um simulador didático e interativo do mips. In: *WEAC*. [S.l.: s.n.], 2006. p. 49–52.
- FERRANDIN, M.; STEPHANI, S. L. Ferramenta para o ensino de programação via internet. *Anais SULCOMP*, v. 1, 2012.
- FRÓES, I. C.; JUNIOR, J. B. B. Tecnologias de informação e comunicação na educação: um estudo exploratório com os alunos do ensino médio de uma escola particular em são luís–ma. *Temática*, v. 12, n. 3, 2016.
- GAMMA, E. *Padrões de Projetos: Soluções Reutilizáveis*. [S.l.]: Bookman editora, 2009.
- ISOTANI, S. et al. Uma ferramenta de apoio à aprendizagem de sistemas operacionais. In: *29o Congresso da Sociedade Brasileira de Computação*. [S.l.: s.n.], 2009.
- JR, V. d. A. M.; CAVAZOTTE, F.; ALVES, I. S. Uso efetivo de ambientes digitais no ensino a distância: Uma extensão do modelo tam. 2011.
- JUCÁ, S. C. S. A relevância dos softwares educativos na educação profissional. *Ciências & Cognição*, v. 8, 2011.
- JÚNIOR, S. D. d. S.; COSTA, F. J. Mensuração e escalas de verificação: uma análise comparativa das escalas de likert e phrase completion. *PMKT–Revista Brasileira de Pesquisas de Marketing, Opinião e Mídia*, v. 15, p. 1–16, 2014.

MACHADO, F. B.; MAIA, L. P. Um framework construtivista no aprendizado de sistemas operacionais—uma proposta pedagógica com o uso do simulador sosim. In: *XII Workshop de Educação em Computação (WEI), XXIV Congresso da Sociedade Brasileira de Computação (SBC), Salvador, BA*. [S.l.: s.n.], 2004.

MARCONDES, M. E. R. et al. Materiais instrucionais numa perspectiva ctsa: uma análise de unidades didáticas produzidas por professores de química em formação continuada. *Investigações em Ensino de Ciências*, v. 14, n. 2, p. 281–298, 2016.

MEDEIROS, T. R. et al. Io simulator: Um simulador de dispositivos de entrada e saída para auxiliar o ensino de sistemas operacionais. In: *Workshop de Educação e Informática. XXXI Congresso da Sociedade Brasileira de Computação*. [S.l.: s.n.], 2011.

MORANDI, D.; RAABE, A. L. A.; ZEFERINO, C. A. Processadores para ensino de conceitos básicos de arquitetura de computadores. In: *Workshop sobre Educação em Arquitetura de Computadores-WEAC*. [S.l.: s.n.], 2006. v. 2006, p. 17–24.

NOSCHANG, L. F.; PELZ, E. A. F.; RAABE, A. Portugol studio: Uma ide para iniciantes em programação. *Anais do CSBC/WEI*, p. 535–545, 2014.

PENNA, P. H.; FREITAS, H. C. Análise e avaliação de simuladores de sistemas completos para o ensino de arquitetura de computadores. 2013.

PINTO, T. D. *Uma Ferramenta para Geração e Execução Automática de Testes Funcionais Baseados na Descrição Textual de Casos de Uso*. Tese (Doutorado) — PUC-Rio, 2013.

REIS, F. P.; COSTA, H. Tbc-so/web: Software educativo para aprendizagem de políoticas de escalonamento de processos e de alocação de memória em sistemas operacionais. *SBIE, UFSC, Florianópolis*, 2009.

RODRIGUES, R. P.; MARTINS, C. d. Ensino e aprendizado de pipeline de modo motivante e eficiente utilizando simuladores didáticos. In: *Workshop sobre Educação em Arquitetura de Computadores-WEAC*. [S.l.: s.n.], 2008.

SCHILDT, H.; SKRIEN, D. *Programação com Java: Uma Introdução Abrangente*. [S.l.]: Bookman Editora, 2013.

SOBREIRA, P. d. L. et al. Competição como uma técnica motivacional no ensino de arquitetura de computadores. In: *Workshop sobre Educação em Arquitetura de Computadores-WEAC*. [S.l.: s.n.], 2007.

TANENBAUM, A. S. *Organização estruturada de computadores*. [S.l.]: Prentice Hall, 5th edition., 2007.

TORRES, A. L.; BRITO, A. V. Ferramenta de auxílio no ensino de organização e arquitetura de computadores: extensão ptolemy para fins educacionais. *International Journal of Computer Architecture Education*, v. 2, p. 13–16, 2012.

ULLMANN, M. et al. Neandersim: Simulador gráfico de apoio ao ensino de arquitetura de computadores. In: CSBC, BRASÍLIA, DF. *XXII Workshop sobre Educação em Computação*. [S.l.], 2014. p. 1646.

ZEFERINO, C. A. et al. Um enfoque interdisciplinar no ensino de arquitetura de computadores. *C. Martins, P. Navaux, R. Azevedo, S. Kofuji. Arquitetura de Computadores: educação, ensino e aprendizado*, 2012.