

PROGLIB: Uma Linguagem de Programação Baseada na Escrita de LIBRAS

Ronnie E. S. Santos¹, Cleyton V. C. Magalhães¹, Jorge S. Correia Neto², Sergio S. L. Paiva Júnior¹

¹Unidade Acadêmica de Serra Talhada– Universidade Federal Rural de Pernambuco(UFRPE)

Fazenda Saco, s/n - CEP: 56.900-000 - Serra Talhada – PE – Brazil

²Unidade Acadêmica de Educação a Distância – Universidade Federal Rural de Pernambuco(UFRPE)

Rua Dom Manoel de Medeiros, s/n, Dois Irmãos - CEP: 52171-900 - Recife/PE – PE – Brazil

{cleyton.vanut, ronnie.gd, jorgecorreianeto, sslpaiva}@gmail.com

***Abstract.** According to the Brazilian Higher Education Census, in 2009 there were 4,660 deaf students in higher education. In this context, the great difficulty of deaf students who attend Computer courses is the initial experiences of the programming logic. This paper aimed to construct a programming language based on the Brazilian sign language (LIBRAS) applied to lessons of logic programming by hearing impairment learners, using an experimental approach, exploratory and descriptive, validated by a case study experiment. The leading contribution of this research was the construction, evaluation and validation of a simplified framework that provides creation of computer programs by hearing impaired students, supported by an IDE planned to provide the activities through a virtual interpret.*

***Resumo.** Conforme dados do Censo Brasileiro da Educação Superior, em 2009 o número de alunos surdos matriculados no ensino superior chegava a 4.660. Neste contexto, a grande dificuldade dos alunos surdos que frequentam cursos de computação está nos conceitos iniciais da lógica de programação de computadores. O presente trabalho objetivou a construção de uma linguagem de programação baseada na de LIBRAS para o estudo de lógica de programação por deficientes auditivos, utilizando uma abordagem experimental, de caráter exploratório-descritivo e empregando um estudo de caso para a validação do experimento. A principal contribuição da pesquisa foi construção, avaliação e validação do framework simplificado para a criação de programas por deficientes auditivos, apoiado por uma IDE projetada para dar suporte as atividades, através de um intérprete virtual.*

1. Introdução

O debate acerca do acesso e permanência de estudantes surdos no ensino superior é um fato relativamente novo e bastante importante no cenário educacional para muitos educadores (MOREIRA, 2008). Segundo Bisol et al. (2010), dados do Ministério da Educação indicam que em 2003 apenas 665 surdos frequentavam a universidade, mas que em 2005 esse número havia aumentado para 2.428, entre instituições públicas e privadas. Conforme dados do Censo Brasileiro da Educação Superior, em 2009 chegava

a 4.660 o número de surdos matriculados no ensino superior (SECRETARIA DE EDUCAÇÃO CONTINUADA, 2011).

Os surdos são capazes de exercer qualquer função na sociedade que não requeira exclusivamente habilidades auditivas, por isto o acesso desta comunidade ao meio acadêmico pode acontecer na grande maioria dos cursos disponíveis nas instituições brasileiras. Uma busca rápida na Internet permite encontrar algumas experiências com surdos em diversos cursos de graduação, desde os relacionados com as áreas humanas até os que são ligados com a computação.

No contexto da computação, como requisitos básicos para a formação nos cursos, os alunos devem-se desenvolver certas habilidades e competências nas matérias introdutórias, incluindo aquelas que abordam a introdução à Lógica de Programação (SANTANA; SANTANA, 2010). É neste período inicial das disciplinas de programação que aparecem as dificuldades mais relevantes encontradas pelos estudantes, seja pela exigência lógico-matemática predominante na disciplina, ou mesmo pela dificuldade de apreensão do ritmo de aprendizagem de cada aluno (ROCHA *et al.*, 2010).

Existem diversos artifícios usados a fim de facilitar o ensino-aprendizagem de lógica de programação, de modo geral. O Portugol, por exemplo, é uma pseudo-linguagem algorítmica muito utilizada na descrição de algoritmos e destaca-se pelo uso de comandos em português, o que facilita o aprendizado da lógica de programação (ESMIN, 1998). Outras técnicas utilizam recursos multimídia que viabilizam uma maneira de aprimorar o aprendizado de lógica de programação (TORI; MAINENTE, 1999). No caso particular do ensino de lógica de programação para surdos a literatura dispõe ainda de pouco material, sendo comumente encontradas propostas que relatam a utilização de dicionários virtuais bilíngues apresentando definições de conceitos indispensáveis para o entendimento de assuntos em língua de sinais (SANTANA; SANTANA, 2010).

A pesquisa em Ciência da Computação permeia praticamente todas as atividades humanas e, portanto, se inter-relaciona com as várias disciplinas existentes (WAZLAWICK, 2008). Neste sentido, o objetivo deste estudo é a construção de uma nova linguagem de programação de alto nível, baseada em Java, para ser aplicada ao ensino de lógica de programação para estudantes surdos. O artigo segue organizado em seis seções, a partir desta introdução. A segunda seção apresenta informações conceituais acerca da temática do assunto. A terceira seção relata experiências anteriores na mesma linha de pesquisa. Na seção subsequente são apresentados os procedimentos metodológicos de pesquisa adotados para realização dos trabalhos. Logo após, na quinta seção, os resultados do estudo são apresentados e discutidos e, a sexta seção trata das considerações finais.

2. Referencial Teórico

2.1 Linguagens de Programação

O computador é uma ferramenta presente em diversas atividades no contexto atual e por causa da grande diversidade deste espaço, existem atualmente diversas linguagens de programação com metas e níveis de abstração diferentes. Uma linguagem de programação de alto nível é caracterizada por um nível de abstração relativamente

elevado, mais próxima da linguagem humana. Por outro lado, as linguagens de programação de baixo nível estão diretamente relacionadas com a arquitetura do computador e utiliza somente instruções de processador, sendo necessário conhecimento dos registradores da máquina (SIEBRA, 2010). Esta relação de nível é que permite que uma linguagem de programação seja escrita a partir de outra linguagem de mais baixo nível.

Segundo Sebesta (2003), um conceito importante que envolve as linguagens de programação são os critérios de avaliação, definidos como características que influenciam o processo de utilização das mesmas. A Legibilidade é o critério que avalia a facilidade com que os programas são lidos e entendidos; a Capacidade de Escrita (*Writability*) é a medida de quão facilmente a linguagem pode ser usada para construir programas. Os critérios de avaliação ainda envolvem a confiabilidade, que está relacionada com o fato de um programa se comportar de acordo com suas especificações sob todas as condições, além dos custos associados a esta linguagem.

O estudo das linguagens de programação é semelhante ao estudo de linguagens naturais: a sintaxe de uma linguagem de programação é a forma de suas expressões, instruções e unidades de programa, e a semântica representa o significado da combinação destes três elementos efetivando a condição de que se o valor atual da expressão for verdadeiro, esta deve ser executada. A descrição formal da sintaxe de uma linguagem inclui unidades de nível mais baixo, os lexemas, que incluem a totalidade de símbolos aceitos pelo contexto (*tokens*) e definem identificadores, operadores e palavras especiais (MELO & SILVA, 2003). O processo para geração de uma linguagem de programação abrange os seguintes passos: análise léxica, análise sintática, análise semântica, gerador do código intermediário e gerador do código objeto, podendo também haver etapas de otimização do código (SEBESTA, 2003).

2.2 Linguagem Brasileira de Sinais

As línguas de sinais são as línguas naturais, de modalidade gestual-visual, utilizadas para comunicação pelas comunidades surdas, e possuem estruturas gramaticais como qualquer língua oral-auditiva. Desse modo, são oficializadas como línguas pelo fato de serem compostas pelos níveis linguísticos: fonológico, morfológico, sintático e semântico. A Língua Brasileira de Sinais – LIBRAS - foi adotada pela nação brasileira após o esforço persistente das pessoas com esta deficiência, sendo, portanto, reconhecida através da Lei nº 10.436, de 24-4-2002 e da Lei nº 10.098, de 19-12-2002. (AZEREDO, 2006).

A estrutura da LIBRAS é diferente da língua portuguesa, uma vez que segue a sequência de ideias que são processadas pelas pessoas surdas, em conformidade com o modo como esta percebe a realidade. Neste contexto, a configuração das mãos representa as formas que as mãos podem assumir durante a realização de um sinal. Estas configurações podem ser diferenciadas a partir do número de dedos estendidos, da contração da mão, ou seja, aberta ou fechada e pelo contato dos dedos.

Assim como os falantes de línguas orais-auditivas expressam emoções através da tonalidade da voz ao falar, os surdos utilizam as expressões faciais para a mesma finalidade de identificar, por exemplo, se uma sentença é interrogativa, afirmativa ou negativa. A orientação/direção dos sinais está relacionada à direção assumida pela mão a

executar um sinal. Nem todos os sinais possuem direção, e os que possuem podem ter a inversão do mesmo para expressar ideia de oposição. É o que acontece, por exemplo, com os sinais de ir e vir. (BRITO, 1995)

Dentre as diversas diferenças entre LIBRAS e o português, é importante salientar que na língua de sinais não são utilizados artigos, preposições, conjunções e outros conectivos. Assim, com características semelhantes às linguagens orais-auditivas, através do uso da LIBRAS é possível um deficiente auditivo expressar tanto ideias sutis como complexas, assim como abstratas, fazendo com que estas pessoas possam se inserir na sociedade como uma pessoa capaz de ouvir normalmente.

3. Trabalhos Relacionados

3.1 Sistema de Ensino de Algoritmos para Surdos

Este artigo realiza uma análise do ambiente atual, no qual o desenvolvimento das tecnologias da informação torna cada vez mais importante realizar uma inclusão social e digital das pessoas que possuem algum tipo de deficiência. Desta forma, é proposto o desenvolvimento de um sistema no qual as pessoas com deficiência auditiva sejam capazes de desenvolver seus próprios programas, utilizando a língua de sinais, facilitando, portanto, o aprendizado e entendimento da lógica de programação.

Os autores acreditam que se o surdo dispuser de um ambiente de desenvolvimento onde possa escrever seus próprios programas, terá um processo de aprendizagem mais simples e melhor, compreendendo os conceitos que norteiam o desenvolvimento de programas. Apesar do projeto apresentado neste artigo não ter sido implementado em sua totalidade, sua proposta se torna significativa pelo fato de que este percebe a dificuldade das pessoas que possuem deficiência auditiva no processo de construção de um *software* (GALLERT, 2010).

3.2 Dicionário Virtual Bilíngue: Uma Proposta para o Ensino e Aprendizagem de Lógica de Programação para Surdos

Partindo da ideia de que o ensino-aprendizagem de Lógica de Programação é uma tarefa complexa pelo fato de exigir certas habilidades do aluno, e que esta dificuldade aumenta significativamente quando o aluno possui deficiência auditiva, este estudo propõe o desenvolvimento de um Dicionário Bilíngue capaz de prover a explicação, tanto em português como em LIBRAS, de conceitos básicos de diversas disciplinas, incluindo informática.

A proposta da apresentação destes conceitos básicos serve para que posteriormente, o aluno surdo seja capaz de compreender com mais facilidade os conceitos mais avançados dos tópicos tratados. O dicionário virtual bilíngue foi implementado para ser utilizado em plataforma *Web* e seu desenvolvimento foi baseado na utilização de elementos visuais, como vídeos, imagens e *SignWriting* (sistema de escrita das línguas gestuais). O escopo da aplicação deste dicionário é amplo, pois não possui foco em informática, mas em diversas outras disciplinas, visando melhorar o ensino-aprendizagem de deficientes auditivos (ROCHA *et al.*, 2010)

4. Procedimentos Metodológicos

Esta pesquisa utiliza uma abordagem experimental, de caráter exploratório-descritivo, empregando um estudo de caso para a validação do experimento. Segundo Luciano (2007), o estudo exploratório é definido como uma pesquisa que tem por objetivo proporcionar maior familiaridade com um problema, a fim de torná-lo mais explícito. A pesquisa descritiva tem característica conclusiva, sendo a abordagem qualitativa utilizada para que haja uma maior percepção e compreensão do contexto do problema (MALHOTRA, 2006). Yin (1989), afirma que “o estudo de caso é uma inquirição empírica que investiga um fenômeno contemporâneo dentro de um contexto da vida real”.

A construção do analisador léxico e sintático da linguagem foi implementada utilizando o GALS (GESSER, 2003), um gerador de compiladores que utiliza conceitos sobre expressões regulares (ER), gramáticas e autômatos. A etapa léxica determinou a definição do conjunto de símbolos suportados pela linguagem e as palavras reservadas para a construção de códigos (*tokens*). Toda a lógica da linguagem foi definida neste momento inicial completamente guiado através de dicionários bilíngues e acompanhado por uma intérprete de LIBRAS para validação dos comandos.

A etapa de etapa semântica da linguagem objetivo a conversão dos códigos escritos a partir das definições do analisador léxico e sintático para a sequência de comandos correspondente em Java, de forma a serem posteriormente executados pela *Java Virtual Machine*, nas etapas finais da definição da linguagem. Para tal, foi desenvolvida uma aplicação utilizada como IDE da linguagem, na qual um método é responsável por relacionar cada *token* da linguagem com o seu correspondente em Java desta forma a tarefa de análise semântica, geração de código intermediário, otimizações e geração de código objeto fica sob o controle do compilador Java.

Por fim, um experimento foi executado com a participação de um grupo formado por 8 estudantes com deficiência auditiva e uma intérprete de LIBRAS. A realização desta etapa permitiu observar, do ponto de vista do principal usuário da linguagem, a legibilidade e capacidade de escrita da linguagem, bem como a facilidade para a criação de programas e as principais dificuldades de interação. Executada na forma de oficina, com duração de três horas e meia, esta etapa da pesquisa produziu conclusões relevantes para o alcance aos objetivos especificados.

5. Resultados

Esta pesquisa resultou em três importantes implicações: a) uma linguagem de programação baseada na escrita em LIBRAS; b) uma interface de IDE simples e objetiva com a apresentação de um intérprete visual para auxílio na construção dos códigos e; c) o relato da experiência no desenvolvimento de programas de computador com um grupo de alunos surdos. A especificação dos resultados está expressa na sequência desta seção.

4.1 Construção da Linguagem: Etapas Léxica e Sintática

A etapa léxica da construção da linguagem resultou em um total de 28 *tokens* que especificam os lexemas necessários ao desenvolvimento de um programa na linguagem proposta. O sistema utiliza palavras específicas para a definição dos blocos

de instrução e do próprio programa (começar, terminar, fazer, parar), aumentando a legibilidade da linguagem, pois elimina a presença de inúmeras chaves ({}) como comandos.

A declaração de variáveis permite a existência de dois tipos de dados: um tipo numérico (*variavel_numero*) e um tipo alfabético (*variavel_palavra*). Esta medida carrega a simplicidade de definição de variáveis necessária ao entendimento por parte do estudante surdo. A atribuição de valor a uma variável está vinculada ao sinal “=”, e as operações lógicas e matemáticas utilizam os símbolos: “>” (maior), “<” (menor), “==” (igual), “+” (soma), “-” (subtração), “*” (multiplicação) e “/” (divisão). O uso destes símbolos está relacionado com a própria experiência dos usuários em disciplinas como a matemática, no ambiente escolar.

Os comandos “*digitar_numero*” e “*digitar_palavra*” referem-se à entrada de dados (numéricos e alfabéticos, respectivamente) para atribuição de valor às variáveis em tempo de execução de programa. Enquanto isso, o comando “*mostrar*” realiza a função de visualização dos resultados de processamento pelo programa criado. As estruturas condicionais são criadas utilizando os comandos “*se*”, “*fazer*” e “*parar*”. As estruturas de repetição são introduzidas através dos comandos “*repetir*”, “*parar*” e “*quando*”. A linguagem ainda dispõe de *tokens* auxiliares na construção dos comandos, são eles: o ponto e vírgula, os parênteses e as aspas.

É importante ressaltar que as palavras reservadas da linguagem foram determinadas com base no vocabulário dos deficientes auditivos, tendo o cuidado de escolher termos existentes na linguagem de sinal. Um exemplo claro disso é o uso do termo “*repetir*” para representar a estrutura de repetição da linguagem, uma vez que o termo “*enquanto*” (*while*) não existe no vocabulário da LIBRAS.

Os *tokens* definidos na fase léxica são utilizados na etapa sintática da linguagem para a elaboração da gramática que irá definir a sequência correta de comandos na escrita do programa (quadro 1).

Quadro 1– Gramática que define a sintaxe da linguagem

A	<code><programa> ::= op_comecar <declarar> <pos_variavel> op_terminar;</code>
B	<code><declarar> ::= <variavel> i; <variavel> ::= tipo_numero pal pv <declarar> tipo_palavra pal pv <declarar>;</code>
C	<code><receber> ::= pal op_receber <recebe> pv pal op_receber <operacao>; <recebe> ::= pal num texto;</code>
D	<code><operacao> ::= abre pal <op> pal fecha pv abre pal <op> num fecha pv abre num <op> pal fecha pv abre num <op> num fecha pv; <op> ::= op_soma op_subt op_mult op_divi;</code>
E	<code><pos_variavel> ::= <escolha_comando> i; <escolha_comando> ::= <comando> <pos_variavel>;</code>
F	<code><comando> ::= op_se <sentenca> op_fazer <declarar> <pos_variavel> op_parar op_repetir <declarar> <pos_variavel> op_parar op_quando <sentenca> pv <receber> op_mostrar abre texto fecha pv op_mostrar abre pal fecha pv op_ler1 abre pal fecha pv op_ler2 abre pal fecha pv;</code>
G	<code><sentenca> ::= abre pal <operadores_logicos> num fecha abre pal <operadores_logicos> pal fecha;</code>
H	<code><operadores_logicos> ::= op_maior1 op_menor1 op_igual1;</code>

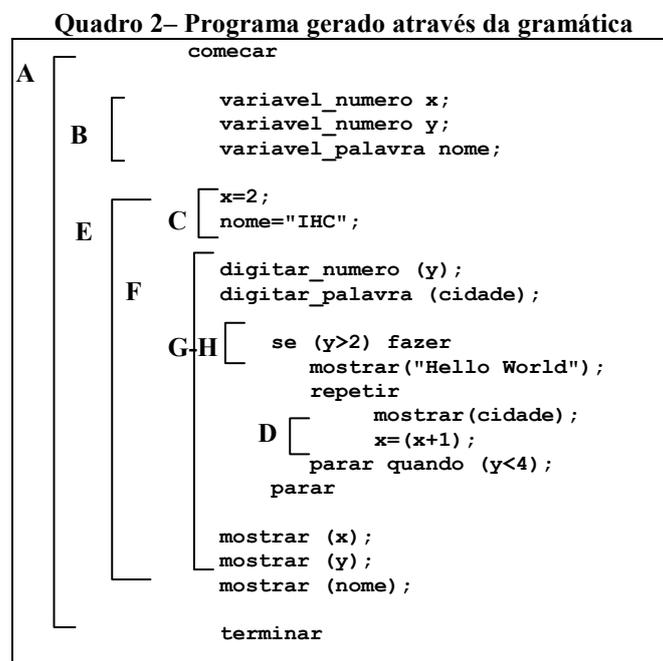
A instrução A indica que o programa deve ser iniciado com o comando “começar”, seguido da declaração de todas as variáveis a serem utilizadas no contexto e logo após com a sequência das demais funções, como atribuição de valores, operações lógico-matemáticas, estruturas condicionais e de repetição, sendo finalizado com o comando “terminar”.

A declaração de variáveis do programa está associada à instrução B. O conjunto das variáveis é definido sempre no início do programa. Cada variável é especificada com um tipo numérico (*variavel_numero*) ou um tipo caractere (*variavel_palavra*), seguido do nome da variável e do ponto e vírgula. Os valores das variáveis podem ser alterados diretamente através da instrução C (*variável = valor numérico* ou *variável = valor String*).

Logo após a declaração de variáveis a instrução E permite que o código criado implemente, através de F, qualquer operação lógico-matemática, de atribuição de valor, de saída e entrada de dados ou a construção de estruturas condicionais e de repetição.

Complementando este conjunto de instruções foram definidas as sentenças lógicas (G - H), possibilitando que uma variável possa ser comparada com outra, ou com um número, ou que exista a comparação entre dois valores numéricos. Além disso, a sintaxe de operações matemáticas expressas em D realiza operações entre números, variáveis e ambos.

O quadro 2 apresenta um programa definido com a sintaxe especificada e utilizando todos os comandos.



4.2 Construção da Linguagem: Etapa Semântica e Código Fonte

Os trabalhos com a análise semântica, geração de código intermediário e geração de código fonte resultaram na criação da IDE nomeada como *Hands*. Esta ferramenta permite a elaboração de códigos na linguagem proposta (fig. 1) e foi projetada para oferecer o maior grau de usabilidade possível para as interações com o usuário.

Todos os comandos são disponibilizados diretamente na interface, permitindo agilizar o processo de elaboração dos programas e produzindo facilidade para aqueles que possuem dificuldade de escrita em português. O sistema permite que o usuário execute o código criado e visualize o seu resultado na tela. O processo de compilação é realizado através da *Java Virtual Machine*.

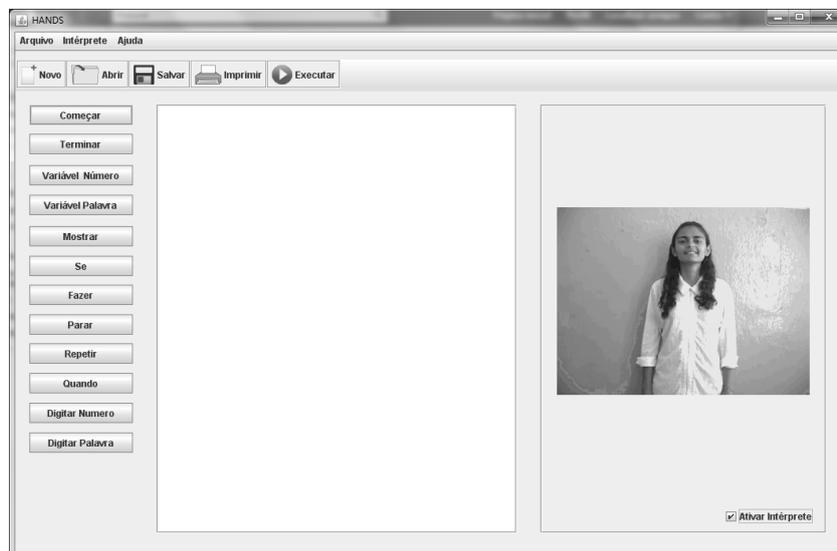


Fig. 1– IDE para geração e execução de códigos na linguagem proposta

O *Hands* possui também uma opção para visualização de um intérprete, elemento considerado importante para o alcance do objetivo desta pesquisa: proporcionar uma forma de linguagem de programação para o estudo de lógica de programação por estudantes surdos. Através do intérprete, cada comando da linguagem, disponibilizado na interface, pode ser visualizado no contexto da LIBRAS. Desta forma, o usuário poderá criar um modelo mental da utilidade do comando. Um exemplo disso é o comando “variável_numero” traduzido pelo intérprete como “local para guardar um número”, ou ainda o comando “se” traduzido em LIBRAS como “se acontecer a proposição”.

4.3 Resultados do Experimento

O experimento realizado teve caráter avaliativo e validador, pois foram desenvolvidas atividades práticas com o uso da linguagem e da ferramenta. Nesta etapa da pesquisa participaram 8 alunos surdos com idade entre 16 e 25 anos, devidamente matriculados entre o nono ano do Ensino Fundamental e o terceiro ano do Ensino Médio, e uma intérprete de LIBRAS, professora da rede pública de ensino. Neste contexto, os estudantes foram apresentados aos conceitos de programação e entenderam como ocorre, num nível bastante básico, o desenvolvimento de um sistema. Conheceram e interpretaram códigos simples de lógica de programação com o auxílio do Portugol e também pequenos códigos executáveis em Java.

Os momentos seguintes das oficinas foram caracterizados por atividades de programação com nível crescente de complexidade, sempre aplicando a linguagem PROGLIB. Durante o desenvolvimento dos códigos pôde-se observar que mesmo sem muitos conhecimentos sobre algoritmos e programação, os participantes não

apresentaram grandes dificuldades em entender os comandos, principalmente pela presença do intérprete do sistema, sendo o maior impasse a compreensão do conceito de variáveis.

A realização da oficina possibilitou concluir que mesmo com pouca experiência sobre os conceitos de programação, o alto grau de legibilidade, apoiado pela facilidade de escrita, permite que deficientes auditivos possam desenvolver programas na linguagem PROGLIB sem grandes dificuldades. Ao mesmo tempo em que a linguagem PROGLIB foi avaliada, a IDE foi validada, tanto pelos usuários alvo do estudo quanto pela professora intérprete, profissional com grande experiência na área de LIBRAS.

Todas as atividades da oficina foram registradas através relatório escrito e vídeo, além de um questionário, objetivando avaliar a opinião do usuário sobre o uso da ferramenta e dos comandos da linguagem, porém entende-se a necessidade da elaboração de um novo experimento, na forma de estudo de caso elaborado para que parte do grupo utilize a linguagem desenvolvida e outra parte trabalhe com outra linguagem de programação conhecida.

6. Conclusões

Este trabalho apresentou a definição de uma linguagem de programação, baseada em Java, para aplicação no processo de ensino-aprendizagem de lógica de programação por estudantes surdos do ensino técnico e superior.

A principal contribuição da pesquisa foi a construção, avaliação e validação do framework simplificado para a criação de programas por deficientes auditivos, apoiado por uma IDE projetada para dar suporte às atividades, através de um intérprete virtual. Além disso, o artigo reúne um conjunto de informações a respeito da atual situação da inclusão de surdos no ensino superior brasileiro e também relata as experiências já aplicadas para o ensino de lógica de programação para alunos neste contexto.

Conclui-se que através do framework o estudante surdo poderá desenvolver disposição preliminar à lógica de programação e ao estudo de algoritmos computacionais, uma vez que a adaptação à outra linguagem será um processo associativo entre os comandos disponibilizados em PROGLIB e os da nova linguagem de programação. O propósito final do trabalho é disponibilizar o material produzido pela pesquisa em código aberto, de modo a ser utilizado por estudantes surdos e professores de lógica de programação com alunos nestas condições ou pesquisadores interessados em desenvolver estudos na área.

Os resultados obtidos na definição da linguagem e os aspectos observados no processo de validação e avaliação encorajam os trabalhos futuros para implementação de novos comandos que possibilitem auxílio ao aluno surdo no entendimento de mais conteúdos da programação de sistemas.

Referências

- AZEREDO, E. Língua Brasileira De Sinais “Uma Conquista Histórica”. Brasília, 2006. Disponível em http://www.cultura-sorda.eu/resources/Reconocimiento_LIBRAS.pdf. Acesso em: 27 maio 2011.

- BISOL, C. A.; VALENTINI, C. B.; SIMIONI, J. L.; ZANCHIN, J. Estudantes Surdos no Ensino Superior: Reflexões Sobre a Inclusão. Cadernos de Pesquisa, v. 40, n. 139, p.147-172, jan./abr. 2010.
- BRITO, Lucinda Ferreira. Por uma gramática de línguas de sinais. Rio de Janeiro: Tempo Brasileiro: UFRJ, Departamento de Linguística e Filologia, 1995.
- ESMIN, A. A. A. Portugol/Plus: Uma Ferramenta de Apoio Ao Ensino de Lógica de Programação Baseado no Portugol. RIBIE 98, IV Congresso da Rede Iberoamericana de Informática Educativa. Brasília, 1998.
- GALLERT C. S.; GUERRA E.; POVALA G., Sistema de ensino de algoritmos para surdos. Computer on the Beach. Florianópolis, 2010
- GESSER, C. E. GALS. Gerador de Analisadores Léxicos e Sintáticos. Universidade Federal de Santa Catarina – UFSC, Florianópolis, 2003.
- LUCIANO, E. M.; TESTA M. G.; ROHDE L. R. Gestão de Serviços de Tecnologia da Informação: Identificando a Percepção de Benefícios e Dificuldades para a sua Adoção. Anais do XXI EnANPAD, Rio de Janeiro, RJ. 2007.
- MALHOTRA, N. K. Pesquisa De Marketing: Uma Orientação Aplicada. 4.ed. Porto Alegre: Bookman, 2006.
- MELO, A.C.V; SILVA, F.S.C. Princípios de Linguagem de Programação. São Paulo: Editora Edgar Blücher, 2003.
- MOREIRA, L. C.; FERNADES, S. Acesso e Permanência de Estudantes Surdos no Ensino Superior, 2008. Acesso em: 02 abr. 2011. Disponível em: www.uel.br/eventos/seminariosurdez/pages/arquivos/palestra_mesa_02_01.pdf.
- ROCHA, P. S.; FERREIRA, B.; MONTEIRO, D.; NUNES, D. S. C. N.; GÓES, H. C. N. Ensino e Aprendizagem de Programação: Análise da Aplicação de Proposta Metodológica Baseada no Sistema Personalizado de Ensino. Revista Novas Tecnologias na Educação, V. 8 N° 3, dezembro, 2010.
- SANTANA, J. E. R. S.; SANTANA, F. J. S. B. Dicionário Virtual Bilíngue: Uma Proposta para o Ensino e Aprendizagem de Lógica de Programação Para Surdos. Instituto Federal de Educação, Ciência e Tecnologia (IFBA), 2010.
- SEBESTA, R. W. Conceitos de Linguagem de Programação. 5ª Edição, Porto Alegre: Editora Bookman, 2003.
- Secretaria de Educação Continuada, Alfabetização, Diversidade e Inclusão. NT 05/2011/ MEC/SECADI/ GAB: Implementação da Educação Bilíngue. Acesso em: 02 maio 2011. Disponível em: <http://www.portalinclusivo.ce.gov.br/index.php/noticias/14-lista-de-noticias/504-mec-numeros-revelam-avanco-da-politica-de-educacao-inclusiva-no-brasil>
- SIEBRA, S. A. Introdução à Programação. Acesso em: 05 maio 2011. Disponível em: <http://pt.scribd.com/doc/18257092/Programacao-1-Linguagem-C-UFRPE>
- TORI, R.; MAINENTE, A. C. Aprendendo Lógica de Programação Via Web. Acesso em: 03 maio 2011. Disponível em: <http://e-spacio.uned.es/fez/eserv.php?pid=bibliuned:1121&dsID=n09tori01.pdf>
- WAZLAWICK, Raul Sidnei. Metodologia de pesquisa para ciência da computação. Rio de Janeiro: Elsevier, 2008.
- YIN, Robert K. - Case Study Research - Design and Methods. Sage Publications Inc., USA, 1989.