# Trabalhando lógica de programação com portadores de deficiência auditiva: a experiência com a Linguagem Proglib e a IDE Hands

Ronnie E. S. Santos<sup>1</sup> Cleyton V. C. Magalhães<sup>1</sup> Jorge S Correia-Neto<sup>2</sup> Leandro M Queiros<sup>2</sup> Guilherme Vilar<sup>2</sup>

Resumo: Conforme dados do Censo Brasileiro da Educação Superior, em 2009, o número de alunos surdos matriculados no terceiro grau chegava a 4.660. Nesse contexto, a grande dificuldade desses alunos 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 em LIBRAS para o ensino 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 a construção, a avaliação e a validação do *framework*, simplificado para a criação de programas por deficientes auditivos, apoiado por uma IDE projetada para dar suporte às atividades, e que se utiliza de um intérprete virtual. Neste texto, também, são sugeridos estudos futuros.

Palavras-chave: Informática na educação. Linguagem de programação. Informática inclusiva.

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 their initial experiences with the programming logic. This paper aimed to construct a programming language based on the Brazilian sign language (LIBRAS) applied to lessons of programming logic 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 interpreter.

Keywords: Informatics in education. Programming language. Accessibility.

# 1 Introdução

O debate acerca do acesso e da permanência de estudantes surdos no ensino superior é um fato relativamente novo e bastante importante no cenário educacional para muitos educadores [1]. Como principais motivos, podem ser apontados a aprovação da lei que estabelece a Língua Brasileira de Sinais (LIBRAS) como língua oficial brasileira e o crescimento das políticas públicas de inclusão que vêm, aos poucos, aumentando o acesso e a participação ativa de pessoas com necessidades especiais em diferentes contextos sociais [2].

Segundo Bisol et al. [3], 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. Além disso, 13.617 cursos superiores incluíram a disciplina de LIBRAS, alguns como disciplina obrigatória, outros como optativa, dentre eles, 61 cursos de Fonoaudiologia e 3.217 cursos de licenciatura [4].

http://dx.doi.org/10.5335/rbca.2014.3145

<sup>&</sup>lt;sup>1</sup> Centro de Informática (CIn) – UFPE

<sup>{</sup>ress, cvcm, lmq}@cin.ufpe.br

<sup>&</sup>lt;sup>2</sup> Departamento de Informática e Estatística (DEINFO) – UFRPE

<sup>{</sup>jorgecorreianeto@gmail.com, guilherme vilar@yahoo.com.br}

Os surdos são capazes de exercer qualquer função que não requeira exclusivamente habilidades auditivas, por isso, o acesso dessa comunidade ao meio acadêmico pode acontecer na maioria dos cursos das 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.

Como requisitos básicos para a formação nos cursos da área da computação, os alunos devem desenvolver certas habilidades e competências nas matérias introdutórias, incluindo aquelas que abordam a introdução à Lógica de Programação [5]. É no período inicial desses cursos que aparecem as dificuldades mais relevantes encontradas pelos estudantes. Existem várias possibilidades de origem das dificuldades no ensino de programação, seja pela exigência lógico-matemática predominante na disciplina, ou pela dificuldade de apreensão do ritmo de aprendizagem de cada aluno [6].

Nesse sentido, existem diversos artificios usados a fim de facilitar o ensino-aprendizagem de lógica de programação. O Portugol, por exemplo, é uma pseudolinguagem algorítmica muito utilizada na descrição de algoritmos que se destaca pelo uso de comandos em português, o que facilita o aprendizado da lógica de programação. A grande desvantagem dessa abordagem é que os algoritmos não podem ser executados no computador, levando o iniciante a ter de imaginar a sua execução [7]. Outras técnicas valem-se, ainda, de recursos multimídia para viabilizar uma maneira de aprimorar o aprendizado de lógica de programação pela utilização de outros meios de visualização e animação de algoritmos [8].

A literatura dispõe de pouco material quanto ao ensino de lógica de programação para surdos. Uma das propostas aplicadas nesse contexto foi a utilização de dicionários virtuais bilíngues que servem de suporte nos estudos, apresentando definições de conceitos indispensáveis para o entendimento de assuntos numa língua compreensível aos surdos [5]. A partir dessa trama conceitual, o presente trabalho teve por objetivo 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 desenvolvimento dessa linguagem está associado à implementação de uma ferramenta que promove a criação de códigos utilizando recursos visuais para a representação dos comandos. Os códigos criados podem ser compilados e o resultado visualizado pelo aluno no momento da interação possibilitou a realização de um estudo de caso para a validação da ferramenta e da linguagem proposta.

O artigo segue organizado em seis seções, a partir desta introdução. A segunda seção apresenta informações conceituais acerca do tema da pesquisa. A terceira 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 última trata das considerações finais.

## 2 Referencial teórico

Esta seção está pautada na apresentação dos conceitos e teorias que envolvem a pesquisa: as definições acerca de linguagens de programação e informações sobre a LIBRAS.

#### 2.1 Linguagens de programação

O computador é uma ferramenta presente em diversas atividades no contexto atual e, por causa da grande diversidade relacionada a esse universo, linguagens de programação com metas diferentes são desenvolvidas. Os primeiros computadores, surgidos a partir da década de 1940, foram inventados para dar suporte às aplicações científicas e, com o passar do tempo, foram utilizados para outros fins, como as aplicações comerciais na década de 1950 e a programação de sistemas que se tornou popular entre as décadas de 1960 e 1970. Esse processo evolutivo da programação foi caracterizado por linguagens que também evoluíram com os sistemas [9].

Uma maneira de categorizar linguagens de programação toma por base o seu nível de abstração. 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 utilizam somente instruções de processador, sendo necessário conhecimento dos registradores da máquina [10]. Essa relação de nível é a condição para que uma linguagem de programação seja escrita a partir de outra linguagem de mais baixo nível.

Segundo Sebesta [9], um conceito importante que envolve as linguagens de programação são os critérios de avaliação, definidos como características que influenciam seus processos de utilização. A legibilidade é o

critério que avalia a facilidade em 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 comportar-se de acordo com suas especificações sob todas as condições. Esse critério é atendido por meio da verificação de tipos, manipulação de exceções ou definição de apelidos. A avaliação de uma linguagem de programação também envolve os custos a ela associados, tais como o custo do treinamento dos programadores que utilizarão a linguagem. Existem, além disso, os custos de escrever programas na linguagem em questão e de compilar um programa escrito nela.

O estudo das linguagens de programação é semelhante ao das linguagens naturais: a sintaxe de uma linguagem de programação é a forma de suas expressões, instruções e unidades de programa, enquanto a semântica representa o significado da combinação desses três elementos, efetivando a condição de que se o valor atual da expressão for verdadeiro, essa 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, bem como palavras especiais [11].

Um gerador de linguagem é um dispositivo que utiliza uma gramática para gerar e descrever a sequência de comandos de outra linguagem. A BNF é uma notação natural, uma metalinguagem para a construção de linguagens de programação, utilizando abstrações para estruturas sintáticas. As abstrações em uma descrição de BNF são conhecidas como não terminais e os lexemas e símbolos das regras são os chamados símbolos terminais [9].

O processo para geração de uma linguagem de programação abrange os seguintes passos (Figura 1): análise léxica, análise sintática, análise semântica, gerador do código intermediário e gerador do código de objeto, podendo, também, haver etapas de otimização do código [9]. O analisador léxico tem a função de ler o código-fonte identificando se cada elemento faz parte do universo dos símbolos da linguagem. O analisador sintático determina se uma cadeia de caracteres pode ser gerada ou não pela gramática que define as expressões da linguagem. O analisador semântico provê métodos pelos quais as estruturas construídas podem ser executadas. O gerador de código intermediário será acionado quando o programa for analisado léxico, sintática (gerada pelo analisador sintático) em uma representação intermediária de código-fonte. Por fim, o gerador de código-objeto tem a função de transformar uma especificação de código intermediário, vinda da etapa anterior, para uma especificação de código *assembly*, podendo, então, ser executado pela máquina. Cada fase possui um identificador de erros e gera um documento enviado para a fase seguinte. E a tabela de símbolos (TS) é usada para guardar informações sobre os nomes declarados em um programa. A TS é pesquisada cada vez que um nome é encontrado no programa-fonte.

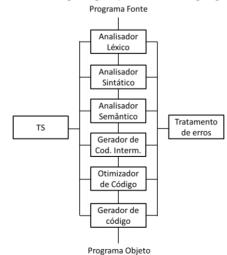


Figura 1: Passos para geração de uma linguagem

Fonte: Sebesta (2003).

#### 2.2 Linguagens brasileiras 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. Mas não existe uma língua universal de sinais. Cada nação possui sua própria língua de sinais, a qual é criada de acordo com a cultura do país, e pode ainda sofrer variações nas diferentes regiões deste. A LIBRAS foi adotada pela nação brasileira após o esforço persistente das pessoas com essa deficiência, por meio das leis nº 10.436, de 24/04/2002, e nº 10.098, de 19/12/2002 [12].

A estrutura da LIBRAS é diferente da estrutura da língua portuguesa, uma vez que segue a sequência de ideias que são processadas pelas pessoas surdas, em conformidade com o modo como percebem a realidade. A estrutura gramatical dessa linguagem é composta pelos seguintes parâmetros: a) a configuração da(s) mão(s); b) o ponto de articulação; c) o movimento; d) a expressão facial; e e) a orientação/direção [13].

A configuração das mãos representa as formas que essas podem assumir durante a realização de um sinal. Tais configurações podem ser diferenciadas a partir do número de dedos estendidos, da contração da mão (aberta ou fechada) e pelo contato dos dedos. Ponto de articulação é o local do corpo do falante onde a mão encontra-se ao ser realizado o sinal. A área restringe-se entre o topo da cabeça e a cintura, e o sinal pode ser feito tocando em alguma parte do corpo ou em locais neutros. Os sinais podem ser estáticos ou dinâmicos. Caso seja dinâmico, o sinal possuirá movimento e estará relacionado com o deslocamento das mãos durante a execução do sinal. Devese ressaltar que o movimento do sinal pode ser realizado apenas uma vez, ou poderá ser marcado por repetições.

Assim como os falantes de línguas orais-auditivas expressam emoções por meio da tonalidade da voz ao falar, os surdos utilizam as expressões faciais para a mesma finalidade. As expressões faciais, bem como as corporais, ajudam a identificar a intensidade da expressão ou se uma sentença é interrogativa, afirmativa ou negativa. Isso significa que a entonação, realizada em uma língua oral-auditiva, é feita por meio das expressões faciais nas línguas de sinais. Esse parâmetro, portanto, é de fundamental importância para o entendimento da língua. A Figura 2 representa, pelas expressões faciais, se uma pessoa está pouco alegre (à esquerda), parcialmente alegre (central) ou muito alegre (à direita).



Figura 2: Expressões faciais representando a intensidade de uma sentença

Fonte: Azeredo (2006).

A orientação/direção dos sinais está relacionada com a direção assumida pela mão ao executar um sinal. Nem todos os sinais possuem direção, mas para os que possuem, há a inversão, expressando ideia oposta. É o que acontece, por exemplo, com os sinais de ir e vir. Dentre as diferenças entre a LIBRAS e o português, é importante destacar que na língua de sinais não são utilizados artigos, preposições, conjunções e outros conectivos.

Assim, com características semelhantes às linguagens orais-auditivas, utilizando LIBRAS é possível que o deficiente auditivo expresse tanto ideias sutis como complexas e abstratas, a o que o leva a inserir-se na sociedade como um indivíduo capaz de ouvir normalmente.

#### 3 Trabalhos relacionados

Esta seção apresenta algumas experiências e abordagens identificadas na literatura que foram usadas para trabalhar o ensino de lógica de programação para estudantes com deficiência auditiva.

Gallert, Guerra e Povala (2010) realizaram uma análise do ambiente atual, no qual o desenvolvimento das tecnologias da informação torna cada vez mais importante realizar a inclusão social e digital dos sujeitos que possuem algum tipo de deficiência. Dessa forma, foi proposto o desenvolvimento de um sistema em que os deficientes auditivos pudessem ser capazes de desenvolver seus próprios programas, por meio da língua de sinais, facilitando, assim, 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 [14].

Ainda em 2010, partindo da ideia de que o ensino-aprendizagem de Lógica de Programação é uma tarefa complexa, pois exige certas habilidades do aluno, e que essa dificuldade aumenta significativamente quando o aluno possui deficiência auditiva, Rocha et al.[6] propõem o desenvolvimento de um dicionário bilíngue capaz de prover a explicação de conceitos básicos de diversas disciplinas, incluindo informática, tanto em português como em LIBRAS. A proposta da apresentação desses 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 Sign Writing (sistema de escrita das línguas gestuais). Após a apresentação da definição do termo, o surdo poderá ver a aplicação desse em uma frase escrita, tanto em português como em LIBRAS [6].

Ambas as propostas discutem a dificuldade das pessoas que possuem deficiência auditiva em estudos ligados à área de computação. Pode-se perceber que o escopo da aplicação de um dicionário, como o utilizado por Rocha et al.[6], é amplo, pois não possui foco em informática, mas em diversas outras disciplinas, visando a melhorar o ensino-aprendizagem de deficientes auditivos. Por outro lado, apesar de a proposta de Gallert, Guerra e Povala [14] ser um importante passo para melhorar os estudos sobre lógica de programação por pessoas com deficiência auditiva, o projeto havia sido implementado em sua totalidade, até então. Dessa forma, nesta pesquisa, é apresentada uma linguagem de programação estruturada com base na LIBRAS, construída em associação com uma IDE na qual os estudantes podem aplicar o seus conhecimentos na prática.

## 4 Procedimentos metodológicos

A pesquisa em Ciência da Computação envolve comumente a criação de um sistema novo, um novo algoritmo ou um novo modelo, e um ponto importante é que a novidade é considerada como algo fundamental [15]. Além disso, a computação permeia praticamente todas as atividades humanas e, portanto, se inter-relaciona com as várias disciplinas existentes [16]. Nesse sentido, o objetivo deste estudo é a construção de uma nova linguagem de programação de alto nível, baseada em Java e estruturada com base na LIBRAS, para ser aplicada ao ensino de lógica de programação para estudantes surdos, e com isso apresentar as percepções da sua aplicação prática.

Quanto aos fins, o estudo exploratório é definido como uma pesquisa que tem o objetivo de proporcionar maior familiaridade com um problema, tornando-o mais explícito, principalmente, por se tratar de um tema pouco investigado ou que não tenha sido abordado anteriormente [17]. A pesquisa descritiva tem característica conclusiva, sendo utilizada a abordagem qualitativa para que haja uma maior percepção e compreensão do contexto do problema [18]. Assim, esta é uma pesquisa de caráter exploratório-descritivo, na qual se buscou avaliar o uso prático dos objetos propostos pela observação de um grupo de estudantes surdos.

#### Fase 1 - Construção da linguagem - etapas léxica e sintática

O analisador léxico e sintático da linguagem foi implementado utilizando o GALS, que é um gerador de compiladores que utiliza conceitos sobre expressões regulares (ER), gramáticas e autômatos [20]. A etapa léxica objetivou a definição do conjunto de símbolos suportados pela linguagem e as palavras reservadas para a construção de códigos (*tokens*). As expressões reconhecidas pelo analisador léxico foram parcialmente definidas por expressões regulares, que, por sua vez, são reconhecidas pelo respectivo autômato determinístico implementado pelo GALS (Fig. 3).

Figura 3: Definição das expressões regulares e tokens da linguagem

```
Definições Regulares

L: [A-Z a-Z]
D: [0-9]
S: [, !:]
AS: ["\"]
WS: [\ \n \t]

Tokens

op_comecar: "comecar"
op_terminar: "terminar"

tipo_numero: "variavel_numero"
tipo_palavra: "variavel_palavra"
op_ler1: "digitar_numero"
op_ler2: "digitar_palavra"
op_receber: "="

op_se: "se"
op_parar: "parar"
op_parar: "parar"
op_repetir: "repetir"
op_quando: "quando"

op_maior1: ">"
op_menor1: ">"
op_menor1: "<"
op_igual1: "=="
```

No analisador sintático, as regras de formação de elementos e frases válidas de uma linguagem foram definidas na gramática da linguagem. Assim, os comandos foram combinados para formar as funções compreendidas pela linguagem: início e fim de programa, declaração de variáveis, atribuições de valor, estruturas condicionais e estruturas de repetição (Fig. 4).

Figura 4: Definição da gramática da linguagem

Toda a lógica da linguagem foi definida, nesse momento inicial, completamente guiada por dicionários bilíngues e com o acompanhamento de uma intérprete de LIBRAS, para validação dos comandos.

# Fase 2 - Construção da linguagem: etapa semântica e código fonte

Esta etapa da pesquisa objetivou 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, para os processos de avaliação e validação dos códigos realizados com os alunos (Fig. 5).

Figura 5: Definição da gramática da linguagem

```
public String converttoJava(int val, String lex){
    switch (val) {
    case 2: return "import
    javax.swing.JOptionPane;\n public class Principal {
    \n public static void main(String[] args) { \n";
        case 3: return "onble";
    case 5: return "String";
    case 5: return "String";
    case 6: return
    "JOptionPane.showMessageDialog";
    case 7: return
    "JOptionPane.showInputDialog";
    case 8: return
    "JOptionPane.showInputDialog";
    case 10: return "f";
    case 11: return "f";
    case 12: return "f";
    case 13: return "s";
    case 14: return "s";
    case 15: return "s";
    case 16: return "s";
    case 17: return "s";
    case 18: return "s";
    case 19: return "s";
    case 19: return "s";
    case 20: return "s";
    case 21: return "s";
    case 22: return lex;
    case 23: return lex;
    case 24: return lex;
    case 25: return "s";
    case 26: return "s";
    case 27: return "s";
    case 28: return lex;
    case 29: return "s";
    case 29:
```

Um método da IDE é responsável por relacionar cada *token* da linguagem com o seu correspondente em Java, e, dessa forma, a tarefa de análise semântica, geração de código intermediário, otimizações e geração de código-objeto ficam sob o controle do compilador Java. A conversão ocorre por meio da associação do código de identificação de cada *token* com o seu correspondente em Java.

Assim, pode-se usar o poder computacional do Java para a execução da análise semântica. Um método é definido para a geração do código intermediário e código-objeto (Figura 6).

Figura 6: Código para geração e execução do código objeto

```
public void rodarClass (String arquivo) {
         try {
            File arq=new
File("Principal.class");
            if (arq.exists()) {
                Process p =
            Runtime.getRuntime().exec("java Principal");
            } else
            JOptionPane.showMessageDialog(null, "Instalar errado.");
            } catch (IOException el) {
                el.printStackTrace();
            }
}
```

A pesquisa não implementou essas etapas do compilador, pois a finalidade do estudo era desenvolver atividades de lógica de programação com alunos surdos, de modo que a simplificação dos comandos, obtida por meio da análise léxica e sintática, foi suficiente para atingir tal objetivo.

#### Fase 3 – Observação da aplicação da ferramenta

Esta etapa foi desenvolvida com a participação de um grupo formado por oito estudantes, com deficiência auditiva e uma intérprete de LIBRAS. A realização dessa etapa permitiu observar, do ponto de vista do principal usuário da linguagem, a legibilidade e capacidade de escrita da linguagem, além do processo de criação de programas e as principais dificuldades de interação com a IDE.

A atividade foi executada na forma de oficina, com duração de três horas e meia. Todos os voluntários eram deficientes auditivos, estudantes de uma escola pública e participantes de um grupo de estudos proporcionado pela intérprete de LIBRAS da escola para acompanhamento de atividades desenvolvidas em sala, incluindo-se a oficina com a ferramenta como parte da atividade do grupo.

## 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 dessa seção.

## Construção da linguagem: etapas léxica e sintática

#### A. Léxico

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 (Figura 7). 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 comados.

Token	Lexema
op_comecar	comecar
op_terminar	terminar
tipo_numero	variavel_numero
tipo_palavra	variavel_palavra
op_mostrar	mostrar
op_ler1	digitar_numero
op_ler2	digitar_palavra
op_receber	-
op_se	se
op_fazer	fazer
op_parar	parar
op_repetir	repetir
op_quando	quando
op_maior1	>
op_menor1	<
op_igual1	
op_soma	+
op_subt	•
op_mult	•
op_divi	/
pal	palavra
num	00
texto	"texto"
pv	i i
virgula	,
abre	(
fecha	)
aspas	

Figura 7: *Tokens* e lexemas da linguagem proposta

A declaração de variáveis permite a existência de dois tipos de dados: um tipo numérico (variavel\_numero) e um tipo alfabético (variavel\_palavra). Essa 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 lexema "=", e as operações lógicas e matemáticas utilizam os símbolos: ">" (maior), "<" (menor), "=" (igual), "+" (soma), "-" (subtração), "\*" (multiplicação) e "/" (divisão). O uso desses símbolos está

relacionado com a própria experiência dos usuários em disciplinas como a Matemática, no ambiente escolar, o que permite uma prévia familiarização dos conceitos, auxiliando na facilidade de escrita.

Os lexemas "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-se os comandos "se", "fazer" e "parar". As estruturas de repetição são introduzidas por meio dos lexemas "repetir", "parar" e "quando". A linguagem ainda dispõe de *tokens* auxiliares na construção dos comandos: 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 língua de sinais. 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. Optou-se por continuar utilizando o ponto-e-virgula (;) ao final de cada linha de comando, pois espera-se que, com o uso dessa ferramenta, o estudante desenvolva uma predisposição para a aprendizagem de outras linguagens de programação, como C e Java, que também utilizam o símbolo como componente léxico.

#### B. Sintático

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 comados na escrita do programa (Fig. 8).

Figura 8: Gramática que define a sintaxe da linguagem

```
A A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A 
A
```

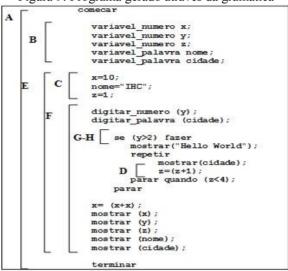
A instrução "A" indica que o programa deve ser iniciado com o comando "começar", seguido da declaração de todas as variáveis a serem utilizadas no contexto e, logo após, com a sequência das demais funções, como atribuição de valores, operações lógico-matemáticas, estruturas condicionais e de repetição, 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. Para facilitar o entendimento do usuário, toda a tarefa de inicialização das variáveis é realizada pela linguagem na etapa semântica, reforçado o critério de confiabilidade na aplicação, necessário nas linguagens de programação. Os valores das variáveis podem ser alterados diretamente por meio 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, por meio de "F", qualquer operação lógico-matemática de atribuição de valor, de saída e entrada de dados, ou, ainda, construção de estruturas condicionais e de repetição.

Complementando esse 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. A Figura 9 apresenta um programa definido com a sintaxe especificada e que utiliza todos os comandos.

Figura 9: Programa gerado através da gramática



Nesse momento inicial, toda a lógica da linguagem foi definida completamente guiada pelos dicionários bilíngues e acompanhada por uma intérprete de LIBRAS para validação dos comandos.

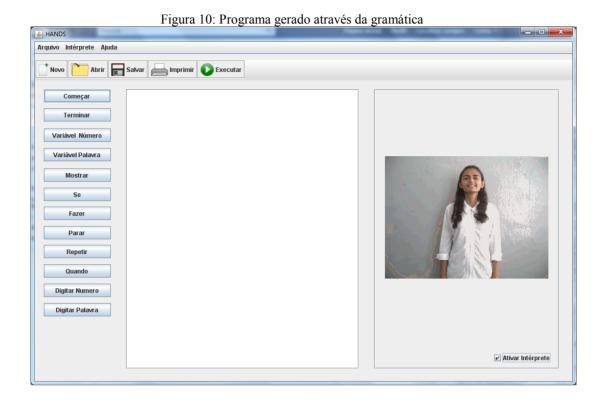
# C. Construção da linguagem: etapa semântica e código fonte

Os trabalhos com a análise semântica, geração de código intermediário e geração de código-fonte resultaram na criação da IDE, aqui nomeada como Hands. Essa ferramenta permite a elaboração de códigos na linguagem proposta (Figura 10) 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 tornando mais fácil a 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.

O projeto de interface foi executado de forma planejada e cuidadosa, produzindo um sistema simples e intuitivo, transparente aos possíveis erros do usuário. As mensagens de erro geradas pela aplicação de comandos errados, construção de expressões inadequadas ou problemas de compilação são apresentadas na tela de modo a facilitar a leitura pelo usuário surdo. O menu de funções do programa utiliza-se de imagens que permitem ao usuário compreender e prever a função de cada botão, partindo dos conceitos da boa modelagem conceitual. Para garantir o maior grau de usabilidade possível, o desenvolvimento do sistema considerou alguns sistemas feitos para o público com deficiência auditiva, além do acompanhamento pela intérprete de LIBRAS que apoiou o projeto.

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. Por meio do intérprete, cada comando da linguagem, disponibilizado na interface, pode ser visualizado no contexto da LIBRAS. Dessa 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".



#### D. Atividade prática

A atividade prática realizada teve caráter avaliativo e validador, sendo desenvolvida com o uso da linguagem e da ferramenta proposta. Nessa etapa da pesquisa, participaram oito 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. 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 no uso da ferramenta, principalmente pela presença do intérprete do sistema. No caso dos comandos aplicados, o maior impasse foi a compreensão do conceito de variáveis, vista na literatura como dificuldade inicial de todos os tipos de estudantes de cursos de informática.

A realização da oficina possibilitou concluir que o alto grau de legibilidade, apoiado pela facilidade de escrita, permite que deficientes auditivos possam desenvolver programas na linguagem PROGLIB, no entanto, deve-se destacar a importância da intérprete para explicar os conceitos básicos de programação antes de os estudantes usarem a ferramenta.

Ao mesmo tempo em que a linguagem PROGLIB foi avaliada, a IDE também foi validada, tanto pelos usuários-alvo do estudo quanto pela professora intérprete, profissional com grande experiência com LIBRAS (Figura 11). Todas as atividades da oficina foram registradas em relatório escrito e em vídeo pelos pesquisadores. Por meio desse material, retirou-se todas as observações e conclusões discutidas neste estudo.



Figura 11: Apoio pedagógico realizado pela intérprete

## 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 está relacionada à concepção, à construção, à avaliação e à validação de uma linguagem de programação para o estudo de lógica de programação por deficientes auditivos, apoiado por uma IDE projetada para dar suporte às atividades, por intermédio 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 nesse contexto.

Conclui-se que com base nos objetos desenvolvidos, que o estudante surdo poderá conhecer e exercitar elementos básicos da lógica de programação e do estudo de algoritmos computacionais. Sendo assim, uma adaptação a outra linguagem de programação poderá ser um processo associativo entre os comandos disponibilizados em PROGLIB e os da linguagem de programação que se deseja estudar. A ferramenta pode ser útil, também, para estudantes sem deficiência auditiva, como uma forma de estudar Portugol.

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 nessas 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

- [1] MOREIRA, L. C.; FERNANDES, S. Acesso e permanência de estudantes surdos no ensino superior, 2008. Disponível em: <a href="https://www.uel.br/eventos/seminariosurdez/pages/arquivos/palestra\_mesa\_02\_01.pdf">www.uel.br/eventos/seminariosurdez/pages/arquivos/palestra\_mesa\_02\_01.pdf</a>. Acesso em: 02 abr. 2011.
- [2] PRESTES, W. C. F.; SOUZA, A. O. *A inclusão do surdo no ensino superior*: uma experiência à legitimidade da LIBRAS na educação dos surdos: uma experiência. INES Instituto Nacional de Educação de Surdos, 2009.
- [3] BISOL, C. A. et al. Estudantes surdos no ensino superior: reflexões sobre a inclusão. *Cadernos de Pesquisa*, v. 40, n. 139, p. 147-172, jan./abr. 2010.

- [4] Secretaria de Educação Continuada, Alfabetização, Diversidade e Inclusão. *NT 05/2011/MEC/SECADI/GAB*: implementação da educação bilíngue. Disponível em: <a href="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>". Acesso em: 02 maio 2011."
- [5] 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.
- [6] ROCHA, P. S. et al. 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.
- [7] ESMIN, A. A. A. *Portugol/Plus*: uma ferramenta de apoio ao ensino de lógica de programação baseado no Portugol. Disponível em: <a href="http://www.c5.cl/ieinvestiga/actas/ribie98/118.html">http://www.c5.cl/ieinvestiga/actas/ribie98/118.html</a>>. Acesso em: 03 maio 2011.
- [8] TORI, R.; MAINENTE, A. C. *Aprendendo lógica de programação via Web.* Disponível em: <a href="http://e-spacio.uned.es/fez/eserv.php?pid=bibliuned:1121&dsID=n09tori01.pdf">http://e-spacio.uned.es/fez/eserv.php?pid=bibliuned:1121&dsID=n09tori01.pdf</a>. Acesso em: 03 maio 2011.
- [9] SEBESTA, R. W. Conceitos de linguagem de programação. 5. ed., Porto Alegre: Editora Bookman, 2003.
- [10] SIEBRA, S. A. *Introdução à programação*. Disponível em: <a href="http://pt.scribd.com/doc/18257092/Programacao-1-Linguagem-C-UFRPE">http://pt.scribd.com/doc/18257092/Programacao-1-Linguagem-C-UFRPE</a>>. Acesso em: 05 maio 2011.
- [11] MELO, A. C. V; SILVA, F. S. C. Princípios de linguagem de programação. São Paulo: Edgar Blücher, 2003
- [12] AZEREDO, E. *Língua Brasileira de Sinais "uma conquista histórica"*. Brasília, 2006. Disponível em: <a href="http://www.cultura-sorda.eu/resources/Reconocimiento\_LIBRAS.pdf">http://www.cultura-sorda.eu/resources/Reconocimiento\_LIBRAS.pdf</a>. Acesso em: 27 maio 2011.
- [13] Língua Brasileira de Sinais. *O que é LIBRAS*. Disponível em: <a href="http://www.libras.org.br/libras.php">http://www.libras.org.br/libras.php</a>>. Acesso em: 27 maio 2011.
- [14] GALLERT C. S.; GUERRA E.; POVALA G. Sistema de ensino de algoritmos para surdos. In: *Anais do Computer on the Beach*. Florianópolis-SC, 2010.
- [15] WAINER, Jacques. *Métodos da pesquisa quantitativa e qualitativa para a ciência da computação*. Disponível em: <a href="http://www.ic.unicamp.br/~wainer/papers/metod07.pdf">http://www.ic.unicamp.br/~wainer/papers/metod07.pdf</a>>. Acesso em: 27 jun. 2010.
- [16] WAZLAWICK, Raul Sidnei. *Metodologia de pesquisa para ciência da computação*. Rio de Janeiro: Elsevier, 2008.
- [17] LUCIANO, E. M.; TESTA M. G.; ROHDE L. R. Gestão de serviços de tecnologia da informação: identificando a percepção de beneficios e dificuldades para a sua adoção. *Anais do XXI EnANPAD*, Rio de Janeiro, RJ. 2007.
- [18] MALHOTRA, N. K. Pesquisa de marketing: uma orientação aplicada. 4. ed. Porto Alegre: Bookman, 2006.
- [19] GESSER, C. E. GALS. *Gerador de analisadores léxicos e sintáticos*. Universidade Federal de Santa Catarina UFSC, Florianópolis, 2003.