



**UNIVERSIDADE ESTADUAL DO CEARÁ**  
**CENTRO DE CIÊNCIAS E TECNOLOGIA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**  
**MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO**

**LIDIANE CASTRO SILVA**

**JLOAD – JAVA LEARNING OBJECT TO ASSIST THE DEAF**  
**UM OBJETO DE APRENDIZAGEM PARA PRÁTICA DE ENSINO DE JAVA BÁSICO**  
**PARA ALUNOS SURDOS**

**FORTALEZA – CEARÁ**

**2016**

LIDIANE CASTRO SILVA

JLOAD – JAVA LEARNING OBJECT TO ASSIST THE DEAF  
UM OBJETO DE APRENDIZAGEM PARA PRÁTICA DE ENSINO DE JAVA BÁSICO PARA  
ALUNOS SURDOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Orientador: Prof. PhD. Francisco C. de M. B. Oliveira

FORTALEZA – CEARÁ

2016

*Deve ser gerada através do preenchimento do Formulário Eletrônico de  
Elaboração da Ficha Catalográfica, disponível no link:  
<http://www.uece.br/biblioteca/index.php/entrega-de-trabalho>.*

X000x

Sobrenome, Nome do 1º autor. (citado na folha de rosto)  
Título principal: subtítulo./Nome completo do 1º autor,  
Nome completo do 2º autor, Nome completo do 3º autor;  
orientação [de]. – Local: ano.  
Nº de folhas.: il.(se houver ilustração); 30 cm.

Inclui bibliografias: f.(nº da folha em que se encontra)  
Trabalho de Conclusão de Curso (Graduação em) –  
Universidade Estadual do Ceará – (UECE).

1. Assunto. 2. Assunto. 3. Assunto. I. Sobrenome, Nome do  
2º autor. II. Sobrenome, Nome do 3º autor. III. Sobrenome,  
Nome do orientador (orient.). IV. Universidade Estadual do  
Ceará – UECE. V. Título.

CDU

LIDIANE CASTRO SILVA

JLOAD – JAVA LEARNING OBJECT TO ASSIST THE DEAF  
UM OBJETO DE APRENDIZAGEM PARA PRÁTICA DE ENSINO DE JAVA BÁSICO PARA  
ALUNOS SURDOS

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências e Tecnologia da Universidade Estadual do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Ciência da Computação

Aprovada em:

BANCA EXAMINADORA

---

Prof. PhD. Francisco C. de M. B. Oliveira (Orientador)  
Universidade Estadual do Ceará – UECE

---

Profa. Dra. Mariela Inés Cortés  
Universidade Estadual do Ceará - UECE

---

Prof. PhD. Paulo Henrique Mendes  
Universidade Estadual do Ceará - UECE

---

Profa. Dra. Elizabeth Sucupira Furtado  
Universidade Estadual do Ceará - UECE

---

Profa. Dra. Cassandra Ribeiro Joye  
Instituto Federal do Ceará - IFCE

Dedico esse trabalho aos meus pais Hermano  
(*in memorian*) e Fátima, com todo meu amor e  
gratidão, por tudo que fizeram por mim ao longo  
de minha vida.

## **AGRADECIMENTOS**

Nenhuma batalha é vencida sozinha, No decorrer dessa luta algumas pessoas estiveram ao meu lado e percorreram esse caminho comigo.

Agradeço ao meu orientador Prof. Fran por ter me recebido e sempre me mostrado o caminho correto a ser seguido, de forma única admirável e exemplar. Desejo poder contribuir a ciência e ao universo acadêmico com a mesma ética e entusiasmo que me transmitiu.

Aos meus pais, por toda dedicação, amor, carinho, exemplo, incentivo e apoio incondicional, em todos os momentos da minha vida.

Agradeço a minha irmã querida que sempre esteve e está do meu lado.

Ao meu filho João Vitor, que é o maior motivo para continuar firme e forte na batalha da vida.

Agradeço a Carol pelo incentivo nos momentos que eu me desmotivada, me mandando ir estudar e me fazendo acreditar no meu potencial.

Ao Éder, Bruno e Joelma pelo companheirismo e parceria diária.

Aos companheiros Marcos, Shara, Michael, Thiago, Fabiano, Alexandre, Felipe pela ajuda, dias e dias de dedicação, envolvimento e por acreditar nessa pesquisa junto comigo.

Ao Alberto por se dedicar no desenvolvimento de algo para ajudar outras pessoas, transformando meu sonho em realidade.

Ao Adriano, por sempre poder contar com você, em todos os momentos e todas as horas.

Aos surdos que participaram no experimento, pelo tempo, envolvimento e dedicação.

Agradeço a todos do Laboratório de Educação a Distância para Pessoas com Deficiência - LE@D, por todo apoio e motivação para seguir essa caminhada.

“Nós somos o que fazemos todos os dias. Deste modo, a excelência não é um ato, mas um hábito.”

(Aristóteles)

## RESUMO

A crescente demanda por programadores de computador apresenta uma boa oportunidade para empregos melhor remunerados para aqueles que são surdos. Cursos online são uma boa opção para uma população geograficamente dispersa, visto que no Brasil 45 milhões de pessoas possui algum tipo de deficiência, o que representa 24% da população. Observa-se altos índice de evasão em cursos de programação Java on-line para surdos causada pela desmotivação. Essa pesquisa estuda, propõe, constrói e avalia o JLoad, um objeto de aprendizado projetado para suavizar a curva de aprendizagem de um aluno surdo. Baseado nas teorias do aprendizado ativo e Zona de desenvolvimento proximal (ZDP) o JLoad possui uma IDE simples e ferramentas de colaboração para permitir uma assistência e um acompanhamento passo a passo do tutor para o aluno de forma remota e assíncrona. Essa colaboração envolve alunos, tutores e intérpretes de Libras. O JLoad foi validado e avaliado por um grupo de 15 surdos, ao mesmo tempo que foi comparado com o Eclipse em diversos pontos: métricas de usabilidade e motivação, tempos para término da oficina, número de auxílios solicitados e notas dos alunos. Em todos os pontos analisados o JLoad apresentou melhores médias e grande parte delas foram ratificadas pelos testes estatísticos apresentando diferenças significativas, indicando um melhor desempenho do JLoad.

**Palavras-chave:** Surdo. Objeto de Aprendizagem. Java. Libras.

## ABSTRACT

Growing demand for computer programmers presents a good opportunity for better paying jobs for those who are deaf. Online courses are a good choice for a geographically dispersed population. In Brazil 45 million people have some form of disability, representing 24% of the population. High dropout rates are observed in Java programming online courses for deaf due to lack of motivation. This research studies, proposes, constructs and evaluates JLoad, an e-learning object designed to smooth the learning curve of a remote deaf Java learner. Based on the active learning theory and proximal development zone theory, JLoad has a simple IDE and collaboration tools to enable asynchronous and remote situated student monitoring and assistance. Such collaboration involves student, tutors and translators. JLoad was validated and evaluated by a group of 15 deaf people. It was compared to Eclipse at several points: usability and motivation metrics, time needed to finish the workshops, number of help requests, and student grades. In all analyzed points, JLoad presented better averages and most of them were ratified by the statistical tests presenting significant differences, indicating a better performance of JLoad.

**Keywords:** Deaf. Learning Object. Java. Libras.

## LISTA DE ILUSTRAÇÕES

Figura 1 – IDE Eclipse. . . . .	22
Figura 2 – Zona de Fluxo. . . . .	29
Figura 3 – IDE <i>Hands</i> . . . . .	36
Figura 4 – Ferramenta iVProg . . . . .	37
Figura 5 – Imagem da ferramenta Learning System . . . . .	38
Figura 6 – <i>Collabode Web</i> IDE colaborativa . . . . .	39
Figura 7 – Login e Gerenciamento da Sessão no <i>COLLECE</i> . . . . .	41
Figura 8 – Estágios do protocolo de colaboração . . . . .	41
Figura 9 – Áreas de trabalho do <i>COLLECE</i> . . . . .	42
Figura 10 – Espaço de Trabalho do Aluno . . . . .	45
Figura 11 – Aluno Submetendo Dúvida ao Tutor . . . . .	46
Figura 12 – Espaço de Trabalho do Tutor . . . . .	46
Figura 13 – Fluxo do JLoad . . . . .	49
Figura 14 – Diagrama de Caso de Uso . . . . .	50
Figura 15 – Acesso ao JLoad . . . . .	51
Figura 16 – Arquitetura em Camadas do JLoad . . . . .	51
Figura 17 – Servidor de Streaming - Red5 . . . . .	52
Figura 18 – Espaço de Trabalho do Aluno . . . . .	54
Figura 19 – Chat entre aluno e tutor . . . . .	55
Figura 20 – <i>Progressive Disclosure</i> . . . . .	56
Figura 21 – Visão Geral Áreas Funcionais do Aluno . . . . .	56
Figura 22 – Espaço de Trabalho do Tutor . . . . .	58
Figura 23 – Espaço de Trabalho do Intérprete . . . . .	59
Figura 24 – Exemplo de Item com Tradução em LIBRAS e Escala de Concordância . . . . .	65
Figura 25 – Nível de Compreensão de Texto em Português . . . . .	69
Figura 26 – Nível de Conhecimento de Eclipse . . . . .	69
Figura 27 – Nível de Conhecimento de Java . . . . .	69
Figura 28 – Diagrama de Caso de Uso dos Atores do JLoad . . . . .	93
Figura 29 – Diagrama de Caso de Uso do Aluno . . . . .	94
Figura 30 – Diagrama de Caso de Uso do Tutor . . . . .	95
Figura 31 – Diagrama de Caso de Uso do Intérprete . . . . .	96

## LISTA DE TABELAS

Tabela 1 – Comparativo das Ferramentas . . . . .	42
Tabela 2 – Conformidade WCAG - JLoad . . . . .	53
Tabela 3 – Contrabalanceamento de Forma Randômica na Entrada . . . . .	61
Tabela 4 – Pontuações do PSSUQ . . . . .	66
Tabela 5 – Média das Métricas de Usabilidade . . . . .	71
Tabela 6 – Média das Métricas de Motivação . . . . .	74
Tabela 7 – Média das Respostas do Questionário Comparativo . . . . .	76
Tabela 8 – Quadro Resumo da Análise Quantitativa das Métricas . . . . .	80

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	14
1.1	PROBLEMÁTICA	15
1.2	OBJETIVOS	16
<b>1.2.1</b>	<b>Objetivo Geral</b>	16
<b>1.2.2</b>	<b>Objetivos Específicos</b>	17
1.3	ORGANIZAÇÃO DO TRABALHO	17
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	19
2.1	SURDEZ	19
2.2	LIBRAS	20
2.3	AMBIENTE INTEGRADO DE DESENVOLVIMENTO	21
2.4	AMBIENTE VIRTUAL DE APRENDIZAGEM	22
2.5	OBJETO DE APRENDIZAGEM (OA)	23
2.6	APRENDIZADO ATIVO	24
2.7	ZONA DE DESENVOLVIMENTO PROXIMAL	25
2.8	TEORIA DA AQUISIÇÃO DE LINGUAGEM, APRENDIZAGEM SITUADA E TEORIA DO FLUXO	27
2.9	MOTIVAÇÃO PARA APRENDIZAGEM	29
<b>2.9.1</b>	<b>A Escala de Motivação Situacional</b>	31
2.10	PRESENÇA SOCIAL	34
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	35
3.1	PROGLIB: UMA LINGUAGEM DE PROGRAMAÇÃO BASEADA NA ESCRITA DE LIBRAS	35
3.2	IVPROG: FERRAMENTA DE PROGRAMAÇÃO VISUAL PARA O ENSINO DE ALGORITMOS	36
3.3	LEARNING SYSTEM (LS)	37
3.4	COLLABODE	39
3.5	COLLECE	40
<b>4</b>	<b>O OBJETO DE APRENDIZADO JLOAD</b>	44
4.1	JLOAD - O PROTÓTIPO INICIAL	44
<b>4.1.1</b>	<b>Design Participativo e Resultados Preliminares</b>	46
4.2	JLOAD - DESENVOLVIMENTO DA FERRAMENTA	48

4.2.1	<b>Requisitos do Objeto</b>	49
4.2.2	<b>Arquitetura</b>	50
4.3	ACESSIBILIDADE	52
4.3.1	<b>Espaço de Trabalho do Aluno</b>	53
4.3.2	<b>Espaço de Trabalho do Tutor</b>	57
4.3.3	<b>Espaço de Trabalho do Intérprete</b>	58
5	<b>AVALIANDO O IMPACTO DO JLOAD NAS OFICINAS</b>	60
5.1	EXPERIMENTO INTRA-SUJEITOS	60
5.2	EXPERIMENTO INTRA-SUJEITOS	60
5.3	DIVISÃO DOS GRUPOS	61
5.4	FASES DO EXPERIMENTO	62
5.5	HIPÓTESES	63
5.5.1	<b>Hipótese Geral</b>	63
5.5.2	<b>Hipótese Corolária Geral</b>	63
5.5.3	<b>Hipóteses Específicas</b>	63
5.6	ANÁLISE SITUADA	63
5.7	INSTRUMENTOS DE COLETA DE DADOS	64
5.7.1	<b>Aplicação da Escala de Motivação Situacional (SIMS)</b>	64
5.7.2	<b>Aplicação do Questionário de Usabilidade para sistema pós teste (PSSUQ)</b>	65
5.7.3	<b>Aplicação do Questionário Comparativo ao Final do Experimento</b>	66
6	<b>RESULTADOS OBTIDOS</b>	68
6.1	PERFIL DOS PARTICIPANTES	68
6.2	USABILIDADE	70
6.2.1	<b>Análise Quantitativa</b>	70
6.2.1.1	Utilidade do sistema (SysUse)	71
6.2.1.2	Qualidade da informação (InfoQual)	71
6.2.1.3	Qualidade da interface (IntQual)	71
6.2.1.4	Média Geral	72
6.2.2	<b>Análise Qualitativa</b>	72
6.3	MOTIVAÇÃO	73
6.3.1	<b>Análise Quantitativa</b>	73
6.3.1.1	Motivação Geral	74
6.3.1.2	Amotivação	74

6.3.1.3	Regulação Externa . . . . .	75
6.3.1.4	Regulação Identificada . . . . .	75
6.3.1.5	Motivação intrínseca . . . . .	75
<b>6.3.2</b>	<b>Análise Qualitativa . . . . .</b>	<b>75</b>
6.4	QUESTIONÁRIO COMPARATIVO DAS FERRAMENTAS JLOAD E ECLIPSE	76
<b>6.4.1</b>	<b>Análise Quantitativa . . . . .</b>	<b>76</b>
6.4.1.1	A ferramenta é fácil de usar . . . . .	76
6.4.1.2	Me comuniquei facilmente com o tutor . . . . .	77
6.4.1.3	Terminei a oficina mais rapidamente na ferramenta . . . . .	77
6.4.1.4	A ferramenta me deixou cansado ou entediado . . . . .	77
6.4.1.5	Estaria motivado a fazer outra oficina na ferramenta . . . . .	77
6.4.1.6	A ferramenta tem recursos para surdos . . . . .	77
<b>6.4.2</b>	<b>Análise Qualitativa . . . . .</b>	<b>78</b>
6.5	TEMPOS, AUXÍLIOS E NOTAS . . . . .	79
<b>6.5.1</b>	<b>Análise Quantitativa . . . . .</b>	<b>79</b>
6.5.1.1	Tempo . . . . .	80
6.5.1.2	Auxílios a conteúdo . . . . .	80
6.5.1.3	Auxílios a Ferramenta . . . . .	81
6.5.1.4	Auxílios Intérprete . . . . .	81
6.5.1.5	Notas . . . . .	81
<b>6.5.2</b>	<b>Análise Qualitativa . . . . .</b>	<b>82</b>
6.6	CONCLUSÃO DOS RESULTADOS . . . . .	83
<b>7</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS . . . . .</b>	<b>85</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>87</b>
	<b>APÊNDICES . . . . .</b>	<b>92</b>
	APÊNDICE A – Diagramas de Caso de Uso do JLoad . . . . .	93
	APÊNDICE B – OFICINAS DESENVOLVIDAS NO EXPERIMENTO . . . . .	97
B.1	OFICINA ECLIPSE . . . . .	97
B.2	OFICINA JLOAD . . . . .	98
	<b>ANEXOS . . . . .</b>	<b>99</b>
	ANEXO A – Questionário PSSUQ em inglês (LEWIS, 1995) . . . . .	100

## 1 INTRODUÇÃO

No Brasil, segundo dados do Censo 2010, realizado pelo IBGE - Instituto Brasileiro de Geografia e Estatística (IBGE, 2010), existem cerca de 45 milhões de pessoas com algum tipo de deficiência, o que representa 24% da população. Apesar da legislação específica existente, a chamada lei de Cotas – Lei 8213/91, pode-se observar que o grande entrave da inserção e manutenção da pessoa com deficiência (PCD) no mercado de trabalho está relacionada à carência na qualificação profissional e aos sistemas de habitação e reabilitação, bem como a falta de estímulos econômicos que facilitem sua contratação pelas empresas, dentre outros. A falta de metodologias e materiais didáticos adaptados para a devida qualificação desse público alvo é perceptível. Como consequência deste problema, a pessoa com deficiência enfrenta dificuldades para alocação no mercado de trabalho e é subjugada a trabalhos de baixa demanda intelectual.

O mercado de desenvolvimento de software cresce significativamente no Brasil e apresenta grandes oportunidades para jovens programadores. O número de vagas aumentou 44,2% em 2015 e estima-se um aumento de 30% em 2016, apesar da atual crise econômica (ZOGBI, 2016). Muitos PCDs ou estão longe de áreas urbanas ou têm dificuldades para obter acesso a escolas regulares. Esse é um dos fatos pelo qual o Laboratório de Educação a Distância para Pessoas com Deficiência - LE@D<sup>1</sup> foi criado pela Dell, em Fortaleza, através da parceria com a Universidade Estadual do Ceará, para pesquisar e desenvolver soluções de ensino a distância que aumentem as oportunidades de empregabilidade de pessoas com deficiência no Brasil, oferecendo 7 cursos, Lógica de programa, Java Básico, Java Intermediário, Java avançado, java Android, Oracle, Gerenciamento de Projetos, todos acessíveis para pessoas com deficiência auditiva, física e baixa visão.

Nesse paradigma de educação inclusiva, os recursos de tecnologia de informação e comunicação utilizados no processo de ensino e aprendizagem precisam também ser acessíveis. Incluem-se, nesse caso, os objetos de aprendizagem (OAs), definidos como uma entidade, digital ou não, que pode ser usada e reutilizada ou referenciada durante um processo de suporte tecnológico ao ensino e aprendizagem (LTS, 2000).

Os objetos de aprendizagem surgiram da necessidade de sistemas mais flexíveis e que fosse de fácil adaptação nos meios educacionais e de disseminação do conhecimento atendendo as exigências por formação da sociedade do conhecimento. Uma característica importante no uso de OAs por pessoas com deficiência é que não há uma barreira rígida de tempo e espaço. A

---

<sup>1</sup> [www.projetolead.com.br](http://www.projetolead.com.br)

aprendizagem pode ocorrer de acordo com o ritmo de cada um, bem como a ordem das atividades pode ser determinada pelo próprio usuário. Ele pode inclusive realizar uma ou mais atividades quantas vezes quiser ou forem necessárias.

## 1.1 PROBLEMÁTICA

O grande desafio da educação a distância (EAD) é expandir, ampliar o acesso a formação e qualificação, assim como diminuir os números de evasão escolar. Segundo o Censo (2014) de EAD no Brasil, as principais causas de evasão apontadas pelas instituições são: (1) falta de tempo do aluno para estudar e participar do curso; (2) acúmulo de atividades no trabalho; e (3) as dificuldades de se adaptar à metodologia. Em se tratando de pessoas com deficiência auditiva, essa evasão é ainda maior, visto que a maioria dos ambientes de EAD não estão preparados para atender alunos surdos, nem possuem material didático produzidos para atender às necessidades em sua primeira língua que é a Libras (Linguagem Brasileira de Sinais).

Além desses problemas, ao codificar, o aluno precisa muitas vezes entender também a língua portuguesa e até mesmo a língua inglesa, tornando o aprendizado de programação mais difícil. O processo de aprendizagem torna-se então bastante cansativo e desgastante para o aluno.

A partir da experiência no laboratório LE@D, tem-se observado uma taxa de evasão entre os alunos surdos de 80%, especialmente durante as primeiras lições do curso. Observa-se que as atividade de programação possui muitas abstrações lógicas, logo, os recursos disponíveis para os alunos deficientes auditivos e as pessoas com surdez não são direcionados para suprir essas necessidades. Durante a pesquisa foram identificados os seguintes problemas:

- a) Muitos alunos sentem-se frustrados por não conseguir desenvolver seus primeiros códigos Java. O objetivo final parece inalcançável e eles desistem ao menor sinal de dificuldade;
- b) É preciso instalar um ambiente de desenvolvimento (IDE) para prática das atividades. Por ser de alta complexidade para quem está iniciando o aprendizado, isso vem causando baixa estima e falta de segurança no aprendizado;
- c) Muitas vezes não é possível para o tutor identificar onde estão as dificuldades do aluno. Além disso o processo de comunicação aluno/tutor não possui um fluxo simples, o que desestimula o aluno a solicitar ajuda;
- d) O aluno, nas primeiras aulas, precisa estar motivado ao ser estimulado para continuar seus estudos. Trata-se de um aluno iniciante que é surdo, está remo-

tamente fazendo o curso, e possui nenhum ou poucos conhecimentos na área. Muitas vezes, ele se sente desmotivado porque precisa instalar e configurar o seu ambiente de desenvolvimento com grande parte das instruções em inglês. Todos esses fatores contribuem para a conseqüente desistência do curso pelo aluno.

Os PCDs precisam de suporte tecnológico para avançar as etapas iniciais de suas vidas como programadores. Por isso, abre-se espaço para outros objetos de aprendizagem e fomenta a produção de tecnologias assistivas, programas que de fato tenham utilidade no mundo real.

A ferramenta está integrada a um Ambiente Virtual de Aprendizagem (AVA) denominado *Dell Accessible Learning* (OLIVEIRA *et al.*, 2016), no qual os alunos são assistidos na realização de oficinas práticas, através de um passo a passo que orienta a criação, compilação e execução dos códigos, além da possibilidade de tirar dúvidas online na própria ferramenta por meio de troca de mensagens textuais e/ou em Libras. Por se tratar de uma ferramenta web, o aluno surdo consegue realizar as atividades em um menor número de passos, tendo em vista que o mesmo não precisará, durante seus estudos, abrir, instalar ou configurar outros programas de criação de códigos que não são acessíveis. Além disso, o aluno poderá estudar e ter acesso aos conteúdos e oficinas realizadas a partir de qualquer lugar com acesso a Internet. Dessa maneira, o mesmo foca na tarefa aumentando as chances de executá-la com sucesso.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo Geral

O objetivo dessa pesquisa é estudar, propor, desenvolver e validar um objeto de aprendizagem chamado JLoad (*Java Learning Object to Assist the Deaf*) que é especialmente para o ensino da linguagem Java. Além disso, ele é inclusivo para surdos, tendo como objetivo diminuir a evasão do aluno surdo no processo de criação de seus primeiros programas em Java visando analisar o impacto na realidade do aluno surdo causado pela introdução de um objeto de aprendizagem à distância, que tem as seguintes características principais: o aluno poder codificar dentro de um AVA; não ser necessário instalar uma IDE para dar os primeiros passos; ser auxiliado pelo tutor à distância; a oficina ser realizada através de um passo a passo; e a possibilidade de enviar todas as dúvidas em Libras.

### 1.2.2 Objetivos Específicos

a) **Aumentar o índice de rendimento acadêmico do aluno surdo;**

Objetiva-se que os alunos se apropriem dos conceitos repassados para que consigam executar toda a oficina. Com o conceito de Zona de Desenvolvimento Proximal, apresentam-se tarefas modularizadas para facilitar o desenvolvimento dos primeiros programas em Java. Dessa maneira, diminui-se a frustração dos alunos e conseqüentemente a evasão.

b) **Promover maior engajamento dos alunos nas oficinas;**

Espera-se permitir que o aluno surdo se sinta mais engajado ao realizar as oficinas, visto que o mesmo poderá interagir com o tutor através dos vídeos de Libras, ao mesmo tempo que pode destacar a dúvida no código desenvolvido e anexar o código para enviar ao tutor pelo chat, melhorando sua interação e motivação na realização da atividade.

c) **Aumentar a presença social do tutor promovendo uma maior interação com o aluno e permitindo aprendizagem situada;**

O JLoad tem o propósito de facilitar a comunicação do tutor e o aluno além de prover ao tutor ferramentas para promover uma análise situada das dúvidas do aluno.

### 1.3 ORGANIZAÇÃO DO TRABALHO

No Capítulo 2, são abordadas algumas definições como surdez, Libras, ambiente integrado de desenvolvimento, ambiente virtual de aprendizagem, objeto de aprendizagem, aprendizado ativo e zona de desenvolvimento proximal. Também são elencadas as teorias de aquisição de linguagem, aprendizagem situada e fluxo, além dos conceitos de motivação para aprendizagem, escala de situação motivacional e presença social.

O Capítulo 3 apresenta cinco trabalhos identificados como mais próximos da presente pesquisa. São eles: o *Proglib*, uma linguagem de programação baseada na escrita de Libras; *iVProg*, uma ferramenta de programação visual para o ensino de algoritmos; o *Learning System* (LS), uma plataforma de ensino à distância especialmente concebida para os surdos; o *Collabode*, um ambiente web de desenvolvimento integrado para Java; e o *COLLEGE*, uma ferramenta colaborativa que permite a programação aos pares distribuída (PPD).

Em seguida, no Capítulo 4 é apresentado o objeto de aprendizagem desenvolvido

nesta pesquisa denominado JLoad. Inicialmente é retratado seu processo de design participativo, logo a seguir, aspectos referentes ao desenvolvimento da ferramenta e, ao final, os espaços de trabalho dos atores.

O Capítulo 5 descreve toda a metodologia para a realização do experimento da pesquisa, juntamente com as hipóteses e instrumentos de coleta de dados utilizados.

Os dados coletados foram submetidos a testes estatísticos e todos os resultados obtidos são apresentados no Capítulo 6, juntamente com uma análise quantitativa e qualitativa baseada nos depoimentos dos participantes.

Por fim, no Capítulo 7, as considerações finais acerca dos resultados encontrados pela pesquisa são apresentadas, juntamente com a descrição dos objetivos atingidos e seus desdobramentos. O capítulo é finalizado com recomendações para novas pesquisas a serem abordados nessa área.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo fornece a fundamentação teórica sobre os conceitos relacionados ao escopo deste trabalho. Inicialmente, são abordados os conceitos que estão relacionados ao público estudado: surdez e Libras, que é a primeira língua dos sujeitos de pesquisa. Em seguida, alguns aspectos sobre a IDE (*Integrated Development Environment*) Eclipse são elencados, visto que ela é utilizada nas atividades de codificação no comparativo com o objeto JLoad. O Ambiente Virtual de Aprendizagem onde está inserido o objeto também é mostrado. Todos os estudos são baseados nas seguintes teorias: aprendizado ativo, teoria da aquisição de linguagem, zona de desenvolvimento proximal, diminuição da carga cognitiva.

### 2.1 SURDEZ

Quando refere-se às pessoas surdas, consideram-se aqueles que são usuários da língua de sinais e que construíram uma identidade a partir das experiências em comunidades de surdos (THOMA, 2009). Pessoas que têm perda auditiva e que utilizam a língua de sinais como forma de comunicação são chamados de ‘surdos’, termo com o qual essas pessoas referem-se a si mesmas e a seus pares. É possível definir uma pessoa surda como aquela que vivencia um déficit de audição que a impede de adquirir, de maneira natural, a língua oral/auditiva usada pela comunidade majoritária e que constrói sua identidade calcada principalmente nessa diferença, utilizando-se de estratégias cognitivas e de manifestações comportamentais diferentes da maioria das pessoas que ouvem.

Em alguns estudos, a expressão ‘deficiente auditivo’ não é utilizada, numa tentativa de re-situar o conceito de surdez, visto que esta expressão é a utilizada, com preferência, no contexto médico-clínico, enquanto que o termo ‘surdo’ está mais afeito ao marco sociocultural da surdez. Nesses estudos se enfatiza a diferença, e não a deficiência, porque “cremos que é nela que se baseia a essência psicossocial da surdez: ele (o surdo) não é diferente unicamente porque não ouve, mas porque desenvolve potencialidades psicoculturais diferentes das dos ouvintes” (BEHARES, 2000). Ora, a distinção entre surdos e ouvintes envolve mais que uma questão de audiologia, é uma questão de significado: os conflitos e diferenças que surgem referem-se a formas de ser.

Considerando que as pessoas surdas percebem as experiências sociais de maneira distinta dos ouvintes, é importante ressaltar que há peculiaridades no processo de aprendizagem desse público, demandando metodologias específicas que estejam atreladas à forma de compre-

ensão dessas pessoas. A história educação de surdos (STROBEL, 2009) está fundamentada na busca por condições indispensáveis na aprendizagem das crianças que nascem com surdez, e ao longo dos anos pode ser verificado que a estratégia predominante é a metodologia bilíngue, na qual a criança aprende Libras como primeira língua (L1) e a língua oral do país onde ela nasce como segunda língua (L2).

No contexto do Brasil, o modelo atual de escola inclusiva ainda enfrenta diversos entraves para atingir um padrão educacional eficaz devido à falta de profissionais qualificados em Libras nas escolas (ALVES, 2012), o que dificulta o aprendizado dos surdos, tendo como consequência a dificuldade de inserção no mercado de trabalho. O acesso por parte desse público ao ensino superior é árduo, o que estreita as oportunidades de ascensão profissional.

## 2.2 LIBRAS

Os surdos no Brasil utilizam a Língua Brasileira de Sinais (Libras), língua reconhecida pela Lei Federal 10.436 de 12 de abril de 2002, e regulamentada pelo decreto 5.626 de 22 de dezembro de 2005 como a língua oficial de comunicação e expressão das comunidades surdas do país. Assim como outras diversas línguas existentes, a Libras é composta por níveis linguísticos, tais como fonologia, morfologia, sintaxe e semântica, da mesma forma que nas línguas orais/auditivas (QUADROS; KARNOPP, 2007). Existem termos, expressões e estruturas específicas nas línguas de sinais, sendo que a diferença principal é a sua modalidade de articulação: visual-espacial. Dentro de um ambiente virtual de aprendizagem (AVA) pode-se trabalhar com conteúdos bilíngues (português e Libras), utilizando recursos de imagens para demonstrar a sintetização de ideias e gerando oportunidades para desenvolvimento de novas tecnologias.

No Brasil, instituições como associações e institutos liderados por surdos organizaram-se no intuito de manter e facilitar o contato entre surdos de diferentes regiões do país, favorecendo a herança da língua de sinais a gerações posteriores de surdos e famílias surdas. A conquista da legalização da Libras se deve principalmente ao esforço de mobilizações coletivas que reivindicaram a importância dessa língua na formação individual e coletiva dos sujeitos surdos.

Atualmente, a demanda das pessoas surdas gira em torno de acessibilidade plena, onde em todos os âmbitos da vida eles tenham a garantia de acesso à comunicação. No âmbito educacional, no que se refere aos processos de tradução para Libras de conteúdos didáticos (processo em que o material é revisado por um profissional bilíngue), como também na interpretação para Libras (processo imediato de tradução), há grande urgência de acessibilidade

linguística.

O reconhecimento da profissão dos tradutores e intérpretes de Libras pela Lei n 12.319 de 2010 (DOU - BRASIL, 2010), a luta pelas escolas bilíngues (BARROS, 2015) e o constante desenvolvimento de estudos e pesquisas sobre como inserir de maneira mais eficaz a partir de outros recursos, como com a legenda acessível (CHAVES, 2012), sugere uma plataforma de educação à distância acessível para pessoas com deficiência onde os conteúdos bilíngues (Português e Libras) garantem uma ampla capacitação de surdo, auxiliando o mercado a formar novos profissionais qualificados.

### 2.3 AMBIENTE INTEGRADO DE DESENVOLVIMENTO

Um ambiente de desenvolvimento integrado, do inglês *Integrated Development Environment* (IDE), é um software que fornece facilidades àqueles envolvidos no desenvolvimento de aplicações. Uma IDE normalmente possui um editor de código-fonte, ferramentas de compilação automatizadas e um *debugger*. Em alguns casos, uma IDE também pode apresentar um sistema de controle de versões além de ferramentas para simplificar a construção de interface gráficas (GUI).

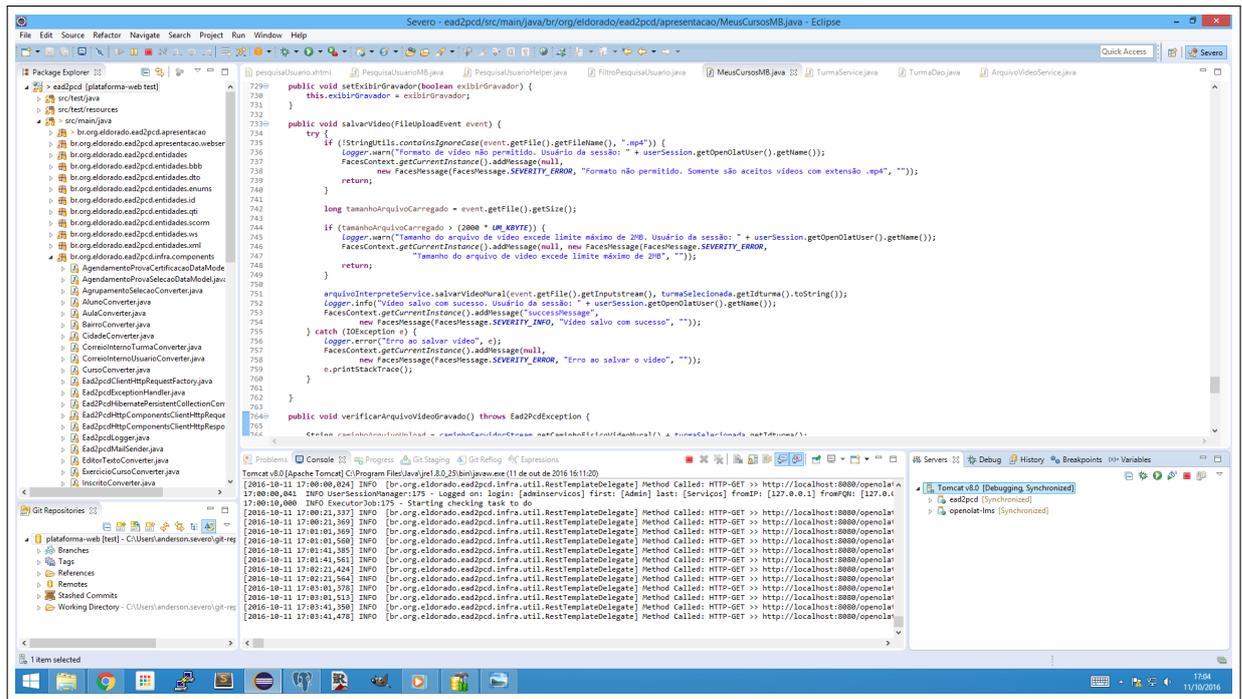
O Eclipse, como pode ser visto na Figura 1, é um *framework* de código fonte aberto, adotado por todas as comunidades Java do mundo. A IBM desenvolveu o projeto Eclipse visando unificar as diferentes plataformas de desenvolvimento usadas dentro empresa. O Eclipse conta com uma coleção de *plugins* que podem interagir conjuntamente e criar um ambiente de desenvolvimento integrado. As principais dificuldades encontradas pelo aluno surdo ao usar/instalar o Eclipse são:

- a) estar à distância;
- b) programar em um ambiente fora do AVA que ele está acostumado durante o aprendizado;
- c) realizar a instalação e configuração de uma IDE se ele nunca desempenhou essa atividade antes;
- d) compreender o funcionamento do Eclipse;
- e) estar limitado às barreiras inerentes da língua presentes nos instrumentos usados para estudo, como livros e internet que são essenciais para o aprendizado do desenvolvedor;
- f) comunicar-se com o tutor que está remoto, muitas vezes não sabe Libras e não

tem acesso ao ambiente de trabalho do aluno.

Todas essas tarefas são novas, desafiadoras e podem acabar desmotivando o aluno no caminho para desenvolver seus primeiros programas.

Figura 1 – IDE Eclipse.



Fonte: (ECLIPSE, 2016)

## 2.4 AMBIENTE VIRTUAL DE APRENDIZAGEM

De acordo com Almeida (2011), Ambiente Virtual de Aprendizagem (AVA) "relaciona-se a sistemas computacionais, destinados ao suporte de atividades mediadas pelas tecnologias de informação e comunicação", podendo integrar diferentes mídias e recursos, apresentando as informações de forma organizada, possibilitando interações entre pessoas e objetos de aprendizagem, visando atingir determinados objetivos.

Os ambientes virtuais de aprendizagem são bons para trabalhar com o público alvo porque podem auxiliar na resolução de problemas dos PCDs, como por exemplo, a dificuldade de deslocamento e ao fato de que estão geograficamente dispersos. Portanto, os AVAs podem gerar grande benefício pois possuem a possibilidade de chegar até onde eles estiverem.

Segundo Maciel (2002), o mundo social e do trabalho precisa cada vez mais de pessoas que possam conviver e produzir coletivamente. O formato de comunicação nos ambientes

virtuais de aprendizagens constitui um fator decisivo para mudar os paradigmas comunicacional e educacional.

Dessa forma, é importante discutir sobre acessibilidade digital e propriedades pedagógicas possíveis que favorecem o desafio de aprender bem e diminuir a curva de aprendizagem. Nos ambientes virtuais de aprendizagem, pretende-se desenvolver nos alunos habilidades de aprendizagem autônoma, embora preferencialmente coletiva, fomentar habilidade de construção de conhecimento, motivar a aprendizagem sem fim.

As pessoas com surdez têm participado e exigido cada dia mais seus direitos na sociedade. Diante disso, surge a necessidade de pesquisar e produzir materiais didáticos digitais que utilizem tecnologias assistivas e promovam o aprendizado através dos recursos de acessibilidade e adaptabilidade.

A acessibilidade em materiais digitais, segundo Dias (2007), significa que qualquer pessoa, incluindo PCD, usando qualquer tipo de tecnologia virtual (gráficos, textuais, especiais para cegos ou para sistemas de computação móvel) seja capaz de visitar e interagir com esses conteúdos, compreendendo claramente as informações neles apresentadas.

Nesse paradigma de educação inclusiva, os recursos de tecnologia de informação e comunicação utilizados no processo de ensino e aprendizagem precisam também ser acessíveis. Incluem-se, nesse caso, os objetos de aprendizagem que serão melhor descritos a seguir.

## 2.5 OBJETO DE APRENDIZAGEM (OA)

Conforme Fabre, Tamusiunas e Tarouco (2003), os OAs são quaisquer recursos que possam ajudar no processo de ensino e serem reutilizados em diversos contextos. De acordo com Gomes (2012), apesar de existirem várias conceituações sobre os OAs, elas apresentam em comum a ideia de que este é "uma entidade educacional reutilizável", ou seja, um recurso com fim educativo que pode ser reaplicado no processo de aprendizagem, sempre que o mesmo seja necessário.

A construção dos objetos de aprendizagem segue alguns princípios que buscam integrar a usabilidade do design, usabilidade da interface, eficiência, eficácia e a usabilidade pedagógica. De acordo com Martins (2004), a usabilidade de design engloba estudos na área da ergonomia, focados em usuário-interface-sistema, conceito que busca definir as características da utilização, do desempenho na interação e leitura das e nas interfaces computacionais pelo usuário. Já a usabilidade pedagógica se refere à necessidade de aprendizagem significativa e à

utilização de ambientes para aprendizagem construtivista. As características da aprendizagem e o uso da tecnologia são inter-relacionados, interativos e interdependentes.

Uma característica importante no uso de OAs dentro de AVAs é a flexibilidade de horários e locais que facilitam o processo de aprendizagem. A aprendizagem pode ocorrer de acordo com o ritmo de cada um, bem como a ordem das atividades pode ser determinada pelo próprio usuário. Ele pode inclusive realizar uma ou mais atividades quantas vezes quiser ou forem necessárias. Como o JLoad pode ser aplicado em diversas oficinas, ele realiza essa missão de um objeto de aprendizagem, a do reuso.

## 2.6 APRENDIZADO ATIVO

As observações de Lowman (2000) corroboram uma afirmação de que o ensino universitário exemplar deve abranger um aprendizado ativo, não apenas em fatos teóricos e metodológicos, mas incluindo relações entre os diferentes ramos do conhecimento, promovendo o pensamento, as habilidades de comunicação e de resolução de problemas, ou seja, deve estimular ao máximo quem recebe a informação, para que o mesmo possa desenvolver um pensamento crítico e colocá-lo em prática.

O construtivismo propõe um modo de ensino em que acontece uma mudança de dinâmica em relação à aula tradicional, em que existe uma transmissão direta e passiva de conteúdo e informação do professor aos alunos (CASTORINA *et al.*, 2006; TAILLE; OLIVEIRA; DANTAS, 1992). Segundo o modelo construtivista a proposta é a participação ativa dos alunos nas atividades escolares, por meio da interação com outros colegas e com o professor, através do uso de artefatos mediadores.

Baseado nesse modelo construtivista, define-se o aprendizado ativo como um estilo de aula no qual os estudantes são envolvidos, levando-os a ter um papel ativo em seu próprio processo de aprendizado. De acordo com Bernhard (1999), o aprendizado ativo pode ser explicado como o responsável pelo estudo científico da mente e da inteligência, percebendo a necessidade de confronto e ruptura com os modelos mentais já conhecidos dos alunos de forma a permitir que eles possam aprender novos conceitos e adquiram novas informações. Esse processo deve acontecer de forma clara, plausível e desafiadora.

Logo, a interação entre os alunos no desenvolvimento das suas atividades, assim como na interação do professor com os alunos durante e após as dinâmicas são aspectos fundamentais do aprendizado ativo. Em relação ao professor é necessário que seu feedback não seja

apenas uma transmissão simples e geral de informação, devendo ser, ao contrário, específica e individual, medindo e quantificando através de padrões de comparação o desempenho de cada um dos alunos.

Segundo Chickering e Gamson (1987), um dos meios para incorporar o aprendizado ativo seria inserir demonstrações ou exemplos durante a aula juntamente com propostas de exercícios e seu posterior debate e discussão com os alunos. Esse modelo de interação e discussão, conforme McKeachie (1986), seria inclusive uma das estratégias mais importantes na promoção do aprendizado ativo, já que ao contrário da aula tradicional permitiria aos alunos uma retenção a longo prazo da informação, a motivação para a pesquisa de leituras adicionais, a possibilidade de aplicação e associação dos conceitos a outros contextos, como também o incentivo para que os alunos desenvolvam novas habilidades e formas de pensamento.

O AVA utilizado nessa pesquisa implementa o conceito de aprendizado ativo através das oficinas, que são as atividades práticas de desenvolvimento na área de Java, nas quais o aluno pode exercitar os conceitos aprendidos na aula, contudo, era preciso utilizar um ambiente de desenvolvimento externo ao AVA e não acessível. Logo, o JLoad veio para resolver esse problema, colocando o aluno surdo em um ambiente onde ele pode desenvolver seus primeiros códigos, enviar dúvidas por chat, e realizar as oficinas por etapas, auxiliando na aprendizagem.

## 2.7 ZONA DE DESENVOLVIMENTO PROXIMAL

A Zona de Desenvolvimento Proximal (ZDP) abrange todas as funções e atividades que a criança ou o aluno consegue desempenhar apenas se houver ajuda de alguém. Essa pessoa que intervém para orientar a criança pode ser tanto um adulto (pais, professor, responsável, instrutor de língua estrangeira) quanto um colega que já tenha desenvolvido a habilidade requerida (VYGOTSKY, 1934).

A distância entre o nível de desenvolvimento real e o nível de desenvolvimento potencial, caracteriza o que Vygotsky denominou de ZDP: "A ZDP define aquelas funções que ainda não amadureceram, mas que estão em processo de maturação, funções que amadureceram, mas que estão, presentemente, em estado embrionário"(VYGOTSKY, 1934).

A constatação de um segundo nível de desenvolvimento e, conseqüentemente, a postulação de uma ZDP, foi decorrente da percepção de diferenças no nível de resolução de problemas entre crianças que, aparentemente, apresentavam os mesmos níveis de desenvolvimento real. Aplicando testes de inteligência nessas crianças, Vygotsky constatou uma equiparação

ao nível do quociente intelectual, ou seja, ambas conseguem resolver sozinhas os mesmos problemas.

Entretanto, ao interagir com essas crianças, ao propor-lhes exercícios mais complexos, além das suas capacidades de resolução independente, esse autor constatou que uma das crianças conseguia, com ajuda, resolver problemas que indicavam uma idade mental superior à da outra que, sob as mesmas orientações; não conseguia solucionar os problemas que a primeira resolvia. Concluiu-se, então, que apesar da aparente homogeneidade dessas crianças quanto ao nível de desenvolvimento efetivamente alcançado, elas, na verdade, diferiam sobremaneira quanto às possibilidades futuras de aprendizagem e desenvolvimento.

Essa diferença entre o que as crianças resolvem independentemente e o que conseguem resolver com a ajuda de um adulto ou colega mais experiente é o que Vygotsky denominou ZDP. Conforme indicam Wertsch, Zanón e Cortés (1988), a ZDP é a região dinâmica que permite a transição do funcionamento interpsicológico para o funcionamento intrapsicológico pois, segundo Vygotsky, todas as funções psicológicas superiores resultam da reconstrução interna de uma atividade social, partilhada.

O conceito de ZDP, elaborado por Vygotsky (1934) em *Mind in Society*, para explorar relações entre educação e desenvolvimento, é comumente associado à distância existente entre o nível de desenvolvimento real e o nível de desenvolvimento potencial (ou possível) de aprendizagem dos indivíduos. O nível de desenvolvimento real refere-se ao nível no qual um indivíduo é capaz de resolver problemas sozinho (ainda que equivocadamente) e o nível de desenvolvimento potencial designa o nível em que esse indivíduo precisa da mediação de outro indivíduo ou da colaboração de outros indivíduos mais competentes (numa determinada habilidade) para avançar na aprendizagem ou ampliar seus conhecimentos. Uma vez internalizados, esses conhecimentos se tornam parte do desenvolvimento de cada indivíduo.

De acordo com Meira e Lerman (2009), tal associação, sozinha, não é suficiente para expressar o escopo da noção de ZDP. Meira e Lerman mostram que pode-se identificar três formulações ou fases no desenvolvimento dessa noção nos trabalhos de Vygostky, vinculadas, nessa ordem, ao desempenho, à interação e à mediação simbólica. A vinculação ao desempenho representa uma crítica aos tradicionais testes de QI e uma tentativa de comparar diferentes patamares entre os desempenhos individuais e assistidos na resolução de problemas. A vinculação à interação é uma extensão da primeira fase, porém priorizando os aspectos sociais de assistência e suporte em detrimento da avaliação de desempenho. Por fim, a vinculação à mediação semiótica vai além da situação sociointeracional imediata, enfocando mais o mundo mediado

simbolicamente que permeia a situação ou as atividades de resolução de problemas.

Ampliando o entendimento de educação para além dos muros da escolarização formal, e pensando nas múltiplas interações sociais existentes, percebe-se o contínuo processo de aprendizagem uns com os outros, seja qual for a esfera considerada – formal, informal ou cognitivo-afetiva. Em muitas dessas interações, mudamos nossa maneira de pensar e/ou agir em relação ao que pensávamos e/ou agíamos antes. Ora, se somos capazes de pensar e/ou agir de forma diferente do que fazíamos antes, é porque nos tornamos uma pessoa diferente, no sentido de que aprendemos e desenvolvemos em certo domínio. E se isso acontece não é porque interagimos pelo mero ato de interagir, mas, sim, porque nos deixamos envolver e envolvemos outros em atos comunicativos. Isso já seria o bastante para verificar o quão potencial é o conceito de ZDP, descrito conforme às formulações anteriores, para explicar como ensinamos, aprendemos e nos desenvolvemos como seres humanos.

No contexto escolar, a sala de aula é um ambiente particularmente fértil e profícuo para a produção de ZDPs, isto é, para a produção de movimentos de aproximação do nível de desenvolvimento real de competências, em relação a determinados conhecimentos/habilidades, para o nível de desenvolvimento potencial de indivíduos. Isso porque uma sala de aula é supostamente configurada tendo-se em mente um ambiente onde haverá uma inter-relação entre ensino e aprendizagem. E esse ensino é, supostamente, atribuído a professores que possuem mais experiência nesses conhecimentos do que aqueles que supostamente vão aprendê-los – os alunos. Ora, se a sala de aula não cumpre essa função, qual seja, a de levar os alunos a progredirem de certo nível de entendimento para outro, então sua existência não faz sentido. Contudo, na sala de aula, é perfeitamente possível conseguir isso, ainda que nem todos progridam no mesmo ritmo ou mesmo que nem todos sejam “motivados” de uma só vez para a aprendizagem.

Através do JLoad os alunos podem realizar as oficinas tendo um acompanhamento de perto do tutor, implementando o conceito da teoria de ZDP.

## 2.8 TEORIA DA AQUISIÇÃO DE LINGUAGEM, APRENDIZAGEM SITUADA E TEORIA DO FLUXO

A hipótese de *input* da teoria de aquisição de linguagem de Krashen (1982), fortemente baseada na proposição da ZDP de Vygotsky, preconiza que as pessoas adquirem uma nova língua com mais facilidade quando a entrada é um pouco acima do seu nível de competência. Nessa mesma teoria, a hipótese do Monitor afirma que alguém que conhece as regras da nova

Língua deve supervisionar todas as etapas no processo de aprendizagem.

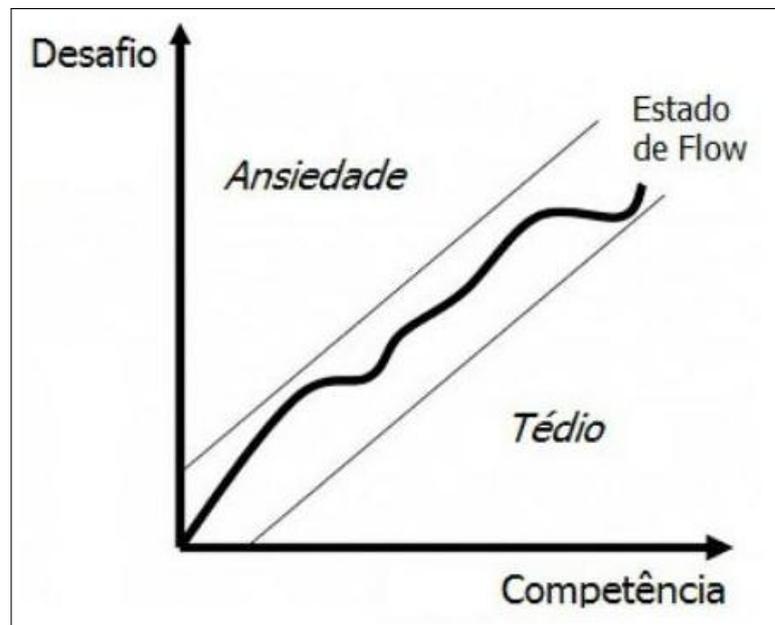
Finalmente, a teoria oferece a hipótese do filtro afetivo, que pode tanto favorecer quanto prejudicar o aprendizado. Basicamente, essa hipótese propõe o aprendizado em contextos de baixa ansiedade e tem como principais fatores a autoconfiança e a motivação. Nas primeiras lições do curso, no qual os alunos enfrentam muitas tarefas não familiares (instalar uma IDE, aprender como compilar, executar, encontrar e corrigir erros simples) que devem ser executadas remotamente, e devido à falta de ferramentas para auxiliá-los, podendo então ocorrer um aumento dos níveis de ansiedade, e diminuição da autoconfiança e motivação.

Os conceitos de aprendizagem situada também são utilizados. Para Lave e Wenger (1991), o processo de aprendizagem pode envolver a interação com pessoas que utilizam uma rede de computadores. A socialização, mesmo à distância, com uma pessoa mais experiente permite que o aluno atinja níveis mais elevados de pensamento e conhecimento.

Um tutor pode oferecer orientações e informações à medida que o aluno aprende, lidando com problemas e questões específicas. Aprendizagem situada é reforçada pelos espaços de trabalho compartilhados. Vários estudos revelam que mostrar o espaço de trabalho é mais importante do que mostrar os rostos dos alunos quando eles estão trabalhando remotamente (WHITTAKER; GEELHOED; ROBINSON, 1993; KRAUT; GERGLE; FUSSELL, 2002). Em (WHITTAKER; GEELHOED; ROBINSON, 1993), ve-se que a adição de um espaço de trabalho compartilhado melhora o desempenho de vários tipos de tarefas, incluindo tarefas de edição colaborativa. Outros autores sugerem que essa melhora vem do aumento da consciência situacional (KARSENTY, 1999; MCCARTHY *et al.*, 1993). Simplificando, os participantes estão cientes das circunstâncias em que as suas ações e as ações de seus parceiros são realizadas. A ação está acontecendo no espaço de trabalho. Tal consciência do espaço reduz os custos comunicacionais, tornando a compreensão das intenções do outro menos trabalhosa. No JLoad, o espaço de trabalho compartilhado é composto de um IDE simples, um chat que é onde os conversantes fazem referência a elementos presentes no espaço de trabalho compartilhado, e as etapas da oficina.

A teoria do fluxo de Csikszentmihalyi (1991) afirma que quanto mais distante é o conjunto de habilidades do usuário daquelas necessárias para a execução de uma tarefa, maior o seu nível de ansiedade, com um impacto direto sobre o desempenho. A Figura 2 mostra a zona de fluxo, no qual a tarefa exige habilidades do usuário compatíveis. À medida que o usuário adquire mais habilidades, uma tarefa mais exigente é necessária para que ele permaneça na zona de ótima experiência.

Figura 2 – Zona de Fluxo.



Fonte: (CSIKSZENTMIHALYI, 1991)

Desta forma, quando os desafios da tarefa ultrapassam as habilidades do usuário, eles tendem a ficar ansiosos. Para uma experiência perfeita, segundo o autor, entre outros fatores, a tarefa deve: (1) Proporcionar uma sensação de controle para o usuário; (2) Permitir um feedback direto e imediato; (3) Ter objetivos claros. Postula-se que durante as primeiras aulas do curso online de Java, os alunos enfrentam tarefas cujas exigências não são compatíveis com as habilidades que eles possuem, por serem insuficientes, os níveis de ansiedade e frustração aumentam. É importante notar que a teoria exige um equilíbrio entre o desafio e a competência. Postula-se que, atualmente, não existe tal equilíbrio durante as primeiras lições do curso e que a introdução de uma ferramenta adequada pode trazer esse equilíbrio.

Essa teoria influencia no JLoad quando quebra-se a oficina em etapas que são incrementalmente mais complexas, uma etapa é mais complexa que a imediatamente anterior. Isso proporciona ao aluno uma sensação de satisfação pelas conquistas e o motiva a buscar melhores resultados também nas etapas com nível de desafio maior.

## 2.9 MOTIVAÇÃO PARA APRENDIZAGEM

O homem, desde o nascimento, demonstra uma grande curiosidade e interesse em aprender e explorar o novo. A motivação é um elemento primordial e natural do ser humano, ela é importante para o seu desenvolvimento cognitivo, social e afetivo.

De acordo com (SISTO *et al.*, 2001), a motivação é a principal variável para a

aprendizagem. Para ele, a motivação é a iniciação e manutenção de comportamento com o objetivo de se atingir uma determinada meta. Através da teoria de metas de realização (AMES; ARCHER, 1988), podemos ter dois grupos focais de metas: a meta para aprender, diretamente relacionada com a tarefa, na qual as pessoas são intrinsecamente motivados, visando aperfeiçoar os conhecimentos; e a outra é a meta de performance, na qual se busca primordialmente destacar as capacidades ou esconder lacunas nas mesmas.

As teorias cognitivas relacionadas com a motivação para a aprendizagem demonstram que existem pelo menos duas formas principais de motivação: a intrínseca e a extrínseca (AMABILE *et al.*, 1994; CSIKSZENTMIHALYI; NAKAMURA, 2014; FORTIER; VALLERAND; GUAY, 1995; HARACKIEWICZ; ELLIOT, 1993; MANDERLINK; HARACKIEWICZ, 1984). Um aluno é intrinsecamente motivado quando ele faz uma atividade porque está interessado nela em si, se envolvendo e gerando satisfação pessoal. Por outro lado, um aluno é extrinsecamente motivado quando ele realiza uma tarefa, com o objetivo de obter recompensas externas, materiais ou sociais. Um aluno que faz determinada tarefa fica motivado a tirar boas notas e conseguir terminar toda a atividade em menos tempo. O JLoad se apoia nisso ao encurtar e facilitar que o aluno coloque seus primeiros programas para funcionar, isso seguramente gera satisfação pessoal.

Além disso, os estudos sobre a motivação em ambiente escolar evidenciam a preocupação de educadores e estudiosos sobre o tema, sendo considerada como um dos principais fatores que favorecem a aprendizagem dos estudantes. O aluno, quando está motivado, consegue demonstrar um maior envolvimento e esforço na aquisição do conhecimento; enfrentar tarefas mais desafiadoras; conseguir realizar as tarefas com persistência utilizando estratégias mais adequadas; e em casos de insucesso, ele não desanima. Por outro lado, a ausência de motivação está presente nos alunos que pouco estudam, além de demonstrarem pouca persistência e, conseqüentemente, aprendem pouco (BORUCHOVITCH *et al.*, 2001).

O desenvolvimento do JLoad surgiu para suprir uma necessidade de qualificar surdos, tendo como desafio manter a motivação na aprendizagem e a identificação da desistência de alunos que realizavam o curso de Java Básico na plataforma EaD utilizada nessa pesquisa. Eles desistiam por se sentirem desmotivados com as dificuldades enfrentadas nos primeiros passos para desenvolver um programa em Java. O JLoad é uma solução tecnológica que busca focar na desmotivação dos alunos, modularizando a oficina, através das etapas, propondo passos menores mais fáceis de ser atingidos e que, baseado na Teoria do Flow, evitam que o aluno fique frustrado. Outra característica do JLoad é que ele já está pronto para o uso, sendo necessário apenas um

dispositivo com acesso a Internet. Não é preciso que o aluno instale e configure um ambiente de desenvolvimento, motivo pelo qual muitos iniciantes se desmotivam e abandonam o curso. Além disso, existe uma facilidade maior de acesso ao Tutor, através de um chat, o que permite que o aluno resolva mais facilmente suas dificuldades.

O JLoad foi concebido para deixar o aluno motivado ao engajá-lo em uma atividade recompensadora: atingir seu objetivo principal que é ter seu primeiro programa funcionando. A modularização da oficina, juntamente com o auxílio situado do tutor e a característica “ready to use” do JLoad aumentam as chances de sucesso.

### **2.9.1 A Escala de Motivação Situacional**

Ao longo dos anos, o interesse pelo estudo da motivação tem crescido, sobretudo no âmbito acadêmico, devido ser um fator importante para a qualidade da aprendizagem e para o desempenho dos alunos.

Dentre as diversas pesquisas relacionadas à motivação, a Teoria da Autodeterminação (*Self-Determination Theory*, SDT) (RYAN; DECI, 2008) se sobressaiu e vem sendo constantemente utilizada para a investigação de casos nesse âmbito. No geral, a teoria defende que o grau de autodeterminação depende do nível de regulação do indivíduo, e esse, por sua vez, varia entre três tipos de motivações: Motivação Intrínseca, Motivação Extrínseca e Amotivação.

A Motivação Intrínseca refere-se ao tipo de motivação onde o indivíduo age por realização pessoal, sendo este nível o que se verifica o maior nível de autodeterminação. E considerando a importância desse tipo de motivação para a qualidade e a criatividade nas aprendizagens, é fundamental a compreensão dos fatores que a inibe e a estimula.

A Motivação Extrínseca está relacionado com o tipo de motivação onde o indivíduo age por razões externas. Diretamente ligado a ela estão quatro níveis de regulação: (i) a Regulação Externa, que é a menos autodeterminada, porque consiste na busca de uma recompensa (a satisfação do aluno ao ver ser programa funcionando) ou evitar uma punição; (ii) a Regulação Introjogada, que consiste em agir para evitar culpa ou ansiedade, ou seja, a preservação de si mesmo (no JLoad, existe uma maior presença social, visto a participação mais próximo do tutor); (iii) a Regulação Identificada, que é a segunda mais autodeterminada, porque consiste no reconhecimento da importância de um determinado comportamento para atingir certos objetivos; e (iv) a Regulação Integrada, que é a mais autodeterminada, porque consiste na congruência entre os valores e as necessidades do indivíduo, o que permite uma plena internalização. Por fim,

a Amotivação refere-se à ausência de qualquer motivação face a determinado fim, e devido a isso, é o nível mais baixo de autodeterminação.

É de grande importância a diferenciação desses níveis para observar a interação mútua e, principalmente, para observar as diversas mudanças na qualidade da motivação a nível situacional de um indivíduo ao longo do tempo, devido ao número sucessivo de interações com o ambiente e a sua conseqüente internalização (VALLERAND; PELLETIER; KOESTNER, 2008).

A Escala de Motivação Situacional (*Situational Motivation Scale*, SIMS) (GUAY; VALLERAND; BLANCHARD, 2000) foi feita, inicialmente, com o intuito de avaliar a motivação situacional no contexto educativo e tem como referência o Modelo Hierárquico de Motivação Intrínseca e Extrínseca (*Hierarchical Model of Intrinsic and Extrinsic Motivation*, HMIEM) de Vallerand (1997). Esse modelo define 3 formatos para a motivação e diferentes níveis de generalidade. Através do grau de especificidade é definido uma hierarquia (global, contextual e situacional). A hierarquia global é o nível mais elevado. Ele define a motivação como uma tendência de indivíduo para interagir com o ambiente, isso pode ser intrinsecamente ou extrinsecamente. Logo, por ser algo estável, está relacionada com a personalidade. Na hierarquia contextual, a motivação se apresenta como orientação preferencial do sujeito para sua vida, como o trabalho. Por fim, a hierarquia situacional considera que a motivação é instável, vista que sofre influência de fatores externos, tais como aspectos do ambiente, funcionando em um certo momento para a realização de uma tarefa específica.

A SIMS é dividida em 16 itens que avaliam os motivos pelos quais um indivíduo se mantém motivado na realização de uma certa atividade. No caso do JLoad, a realização das oficinas de programação. Dentro desses 16 itens, existe uma divisão em quatro subclasses baseadas nos tipos de motivação da Teoria da Autodeterminação:

- a) Motivação Intrínseca, cujo o motivo é o interesse por determinada atividade;
- b) Motivação Extrínseca via Regulação Identificada, onde o motivo é porque é importante e é para o bem da pessoa;
- c) Motivação Extrínseca via Regulação Externa, na qual o motivo é porque é o que é esperado que seja feito;
- d) Amotivação, onde pode haver o reconhecimento da importância de certa atividade, mas a pessoa não se motiva.

Os autores da SIMS não incluíram itens relativos à Regulação Introjetada e à Regulação Integrada, por considerarem que acabariam por não se diferenciar das restantes.

Estudos mostram que a eliminação dos itens 10 e 11 do questionário SIMS original melhora a consistência interna das subclasses (MARTÍN-ALBO; NÚÑEZ; NAVARRO, 2009; GAMBOA; VALADAS; PAIXÃO, 2013), ou seja, a correlação entre diferentes itens da escala aumenta. Dessa maneira, a seguir, apresenta-se os 14 itens do questionário SIMS que são respostas para a pergunta “*Por que estou envolvido nessa atividade no momento?*”:

1. Porque considero que esta atividade seja interessante.
2. Porque estou fazendo isto em benefício próprio.
3. Porque eu preciso fazer esta atividade no momento.
4. Podem existir razões para a realização desta atividade, mas pessoalmente não me identifico com nenhuma.
5. Porque acredito ser uma atividade agradável de ser realizada.
6. Porque acredito que possa me beneficiar com esta atividade.
7. Esta atividade refere-se a algo que eu tenho de fazer.
8. Eu realizo esta atividade, mas não estou certo se compensará.
9. Acredito ser uma atividade divertida de ser realizada.
10. Eu não tenho conhecimento a respeito do que esta atividade possa vir a me agregar.
11. Porque eu me sinto bem com a realização desta atividade.
12. Eu acredito que esta atividade seja importante para mim.
13. Porque eu sinto que eu deveria realizar esta atividade.
14. Eu realizo esta atividade, mas eu não tenho certeza se devo continuar.

Cada uma das dimensões é calculada analisando as respostas (grau de concordância de 1 a 7) de um subconjunto dos itens:

- a) Motivação Intrínseca: itens 1, 5, 9 e 11;
- b) Regulação Identificada: itens 2, 6 e 12;
- c) Regulação Externa: itens 3, 7 e 13;
- d) Amotivação: itens 4, 8, 10 e 14.

O valor associado a cada dimensão é calculado como a média dos valores obtidos nos seus itens correspondentes. Um valor global também é computado como o valor médio dos 14 itens. Alguns itens são positivos e outros negativos. Os itens referentes à dimensão de Amotivação são negativos, as pontuações desses itens devem ser invertidas a fim de computar o

valor global ou quando ela for comparada com as outras dimensões.

## 2.10 PRESENÇA SOCIAL

O conceito de presença social surgiu através de Short, Williams e Christie (1976). Os autores definem o grau de importância entre dois comunicadores, intermediado por um outro meio de comunicação. Eles afirmam que existe diferença em grau de presença social por diferentes meios de comunicação. Essa diferença é o principal papel na forma de interação das pessoas. Logo, conceituam presença social como uma qualidade que pode determinar a maneira como as pessoas interagem e se comunicam.

A presença social faz com que os alunos se sintam à vontade nos ambientes virtuais. As estratégias e atividades desenvolvidas pelos educadores devem levar em consideração a facilidade da interação. Os alunos sempre possuem muitas dúvidas; eles perguntam; respondem; e querem trocar ideias, referenciando a mensagem do outro. Essa comunicação é cheia de saudações, vocativos, referências ao grupo e aos colegas pelo nome, uso de pronomes (nós), além de expressões pessoais de emoções, crenças e valores.

Segundo Sérgio (2007), a percepção da presença social se relaciona com a satisfação dos alunos em relação aos seguintes aspectos:

- a) como se sente incluído em relação ao curso;
- b) os relacionamentos estabelecidos entre os participantes do curso; e
- c) a percepção do aluno sobre o envolvimento do professor, destacado pelo engajamento do professor nas discussões, no incentivo a participação dos alunos nas atividades e o rápido feedbacks das dúvidas apresentadas pelos alunos.

No JLoad, utiliza-se o conceito de presença social no acompanhamento do tutor em relação às atividades que o aluno está desenvolvendo. O aluno pode destacar o código para realçar sua dúvida e enviar ao tutor pelo chat. O tutor observando a localização exata da dificuldade pode auxiliar mais rapidamente. Comentários a respeito das etapas podem ser trocados, fazendo com que o aluno se sinta incluído já que suas dúvidas podem ser enviadas em Libras, bem como suas respostas. Tudo isso permite ao aluno uma maior integração e interação com o tutor através do JLoad.

### 3 TRABALHOS RELACIONADOS

Observa-se uma crescente iniciativa para o desenvolvimento de ferramentas que auxiliem pessoas com deficiência a ter um melhor acesso à aprendizagem. Neste capítulo, são apresentados algumas aplicações que tratam do ensino de programação, mas poucas são focadas no ensino para surdos que é o objetivo dessa pesquisa.

#### 3.1 PROGLIB: UMA LINGUAGEM DE PROGRAMAÇÃO BASEADA NA ESCRITA DE LIBRAS

*Hands* (SANTOS *et al.*, 2011) é um projeto realizado pela Universidade Federal Rural de Pernambuco que tem como a principal contribuição da pesquisa a construção, avaliação e validação de um *framework* simplificado para a criação de programas por pessoas com surdez, apoiado por um desenvolvimento de software integrado, projetado para dar suporte aos alunos para aprender lógica de programação.

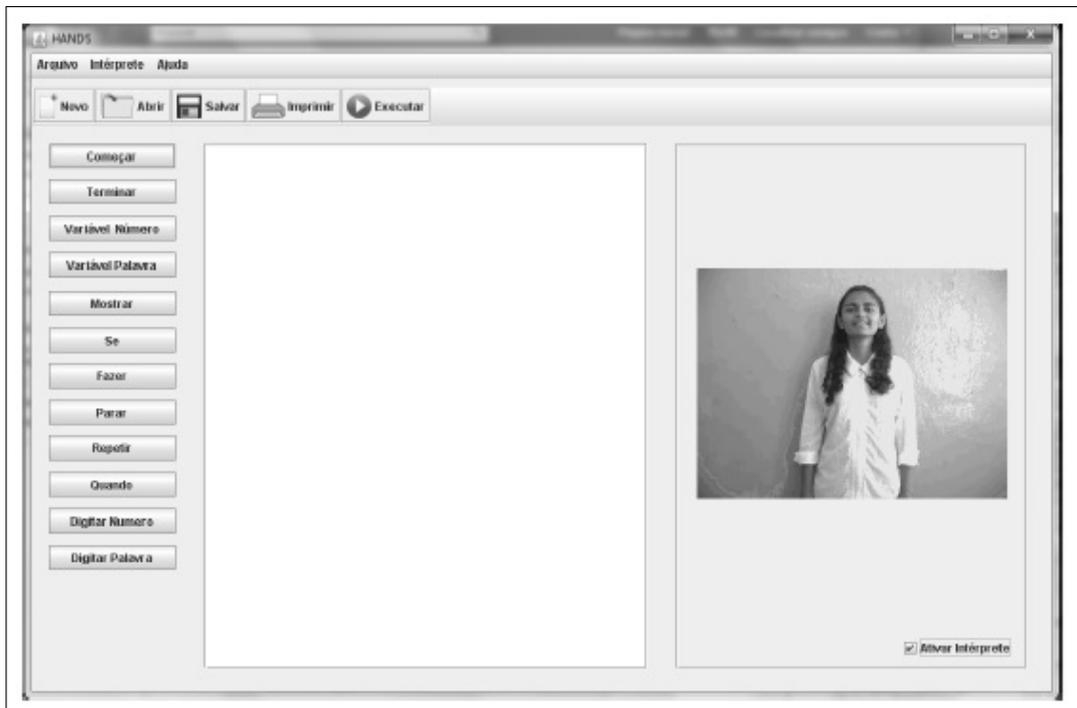
Os trabalhos com a análise semântica, geração de código intermediário e geração de código fonte resultaram na criação da IDE nomeada como *Hands* mostrada na Figura 3. Essa ferramenta permite a elaboração de códigos na linguagem proposta e foi projetada para oferecer o maior grau de usabilidade possível para as interações com o usuário.

Para suavizar a curva de aprendizado, os criadores do *Hands* propuseram uma linguagem de programação semelhante ao *Algol*, a chamada *Proglib* (linguagem de programação baseada em Libras).

A abordagem deste trabalho possui algumas vantagens:

- a) Foco nos conceitos básicos de algoritmos, o que é bom não só para iniciantes, além disso suporta boas práticas de programação;
- b) Menos passos para aprender até conseguir os primeiros programas em execução;
- c) Facilidade para criação de uma interface na IDE mais simples, aplicando conceitos básicos.

A abordagem, no entanto, pode não ser adequada para a aprendizagem de linguagens de programação orientadas a objeto, como Java. Outra questão, mais difícil de resolver, é o fato de que, para aprender uma língua, o aluno tem de dominar primeiro uma outra língua. O cenário é especialmente desafiador para aqueles que já lutam com os desafios de linguagem. *Hands* não foi projetado para educação a distância, nem seus criadores previram colaboração assíncrona entre os usuários do ambiente.

Figura 3 – IDE *Hands*

Fonte: (SANTOS *et al.*, 2011)

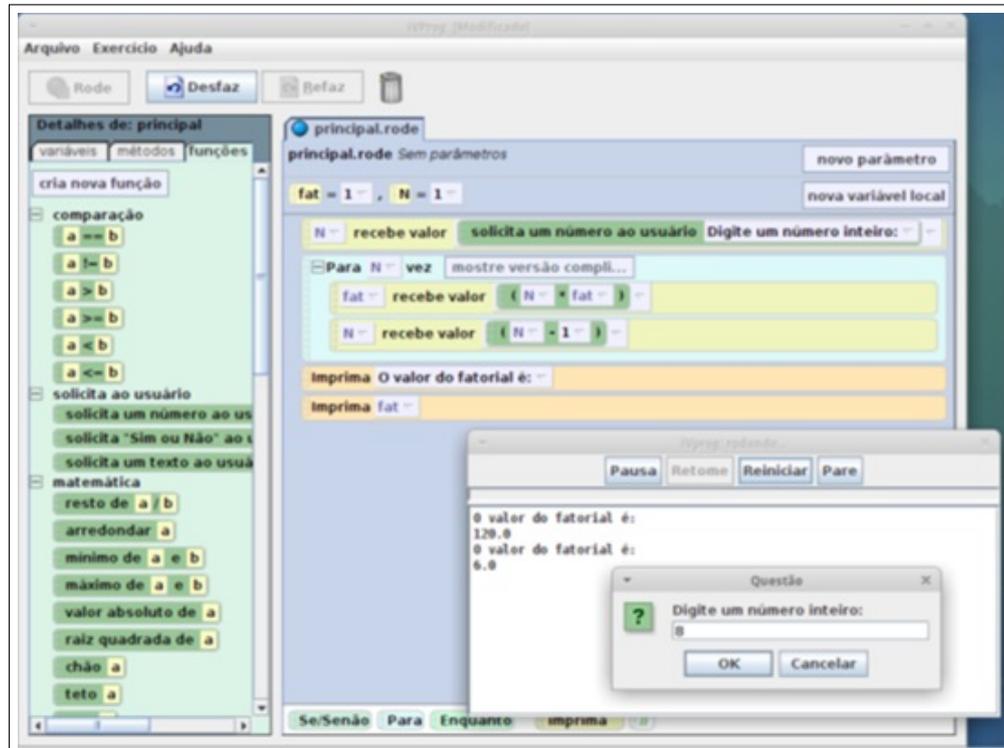
### 3.2 IVPROG: FERRAMENTA DE PROGRAMAÇÃO VISUAL PARA O ENSINO DE ALGORITMOS

iVProg (BRANDAO; BRANDAO; RIBEIRO, 2012) é uma ferramenta baseada no paradigma de Programação Visual, com o intuito de diminuir a sobrecarga cognitiva referente ao aprendizado da sintaxe da linguagem e de ambiente de compilação, desenvolvido por um grande grupo, formado por equipes da Instituto de Matemática e Estatística – Universidade de São Paulo e Escola Politécnica (USP), pretendendo com essa pesquisa aumentar o engajamento, motivação, aprendizado e aproveitamento dos alunos.

No geral, sabe-se que os professores dos cursos de Introdução à Computação gastam mais tempo ensinando detalhes de linguagens (estruturas e sintaxe) do que com o processo de programação propriamente dito (CASPERSEN; KOLLING, 2009). Tendo isso em mente, foi desenvolvida uma ferramenta baseada no paradigma de Programação Visual, o iVProg, com o intuito de diminuir a sobrecarga cognitiva referente ao aprendizado da sintaxe da linguagem e de ambiente de compilação. O iVProg permite que o aprendiz se concentre no problema proposto, focando-se na tentativa de projetar um algoritmo que o resolva. Conseqüentemente, o iVProg permite ao professor dar maior atenção ao ensino das estratégias para reconhecer e resolver problemas do que aos detalhes da linguagem.

A ferramenta iVProg, ilustrado na Figura 4, não foi adaptada para pessoas com deficiência e certamente seria interessante ver algum desenvolvimento nessa direção.

Figura 4 – Ferramenta iVProg



Fonte: (BRANDAO; BRANDAO; RIBEIRO, 2012)

Programação visual, como proposto por ferramentas como o iVProg, tem um apelo interessante para a comunidade surda, uma vez que dependem fortemente de comunicação visual. O paradigma de programação visual reduz a demanda cognitiva da sintaxe da linguagem de aprendizagem e do ambiente de compilador, permitindo que os alunos concentrem-se melhor sobre o problema proposto, especialmente no algoritmo.

### 3.3 LEARNING SYSTEM (LS)

O Learning System (LS) (DRIGAS *et al.*, 2005) é uma plataforma de ensino à distância especialmente concebida para os surdos que oferece vídeos na Linguagem Grega de Sinais (LGS) para cada texto presente no ambiente. No LS, algumas das necessidades especiais dos alunos surdos são satisfeitas: a exibição de informações bilíngüe, alto nível de visualização, e aprendizagem através de vídeo-conferência. Dessa maneira, a plataforma permite o aprendizado utilizando a própria LGS.

O trabalho também relata uma realidade semelhante a brasileira no qual as taxas de

desemprego são maiores entre PCDs. Nesse contexto, os autores justificam a necessidade do LS e explicam o porquê dos conceitos envolvidos.

O uso da internet se faz essencial no desenvolvimento de soluções para inclusão dessa categoria visto que ela permite a flexibilização necessária para as inividualidades de cada PCD. O LS também enfatiza que o uso de comunicações de vídeo possibilita melhores experiências de telecomunicação quando se trabalha com surdos. O resultado de um questionário aplicado pelos autores permite concluir que a maioria dos estudantes preferem que o conteúdo seja apresentado na língua de sinais e que consista principalmente de imagens e vídeos.

O sistema, mostrado na Figura 5 abaixo, oferece as seguintes funcionalidades propriamente adaptadas aos requisitos especiais dos surdos: ferramenta de busca; mapa do site; glossário bilíngue; ferramenta de extração de conteúdo; conjunto de estatísticas para o professor; ferramentas de testes e exames para avaliação dos estudantes pelo professor.

Figura 5 – Imagem da ferramenta Learning System



Fonte: (DRIGAS *et al.*, 2005)

O LS é baseado na vigilância assíncrona do processo educacional e comunicação síncrona através de chat. Ele possui três níveis de usuário: do professor, do estudante e do administrador.

Ao professor é permitido anunciar um novo curso via e-mail ou quadro de avisos; adicionar material extra via links especiais; separar a classe em menores grupos para realização

de certos trabalhos; além de responder e anunciar as dúvidas dos alunos utilizando meios assíncronos de comunicação.

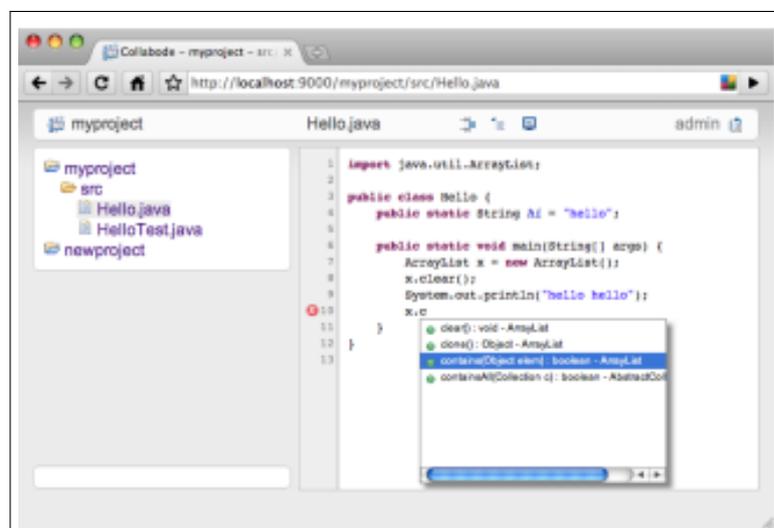
O aluno pode se registrar em vários cursos; adicionar dúvidas para o professor; acessar o *help* da ferramenta; acessar o glossário; selecionar certas informações do curso para suas anotações pessoais; participar das discussões; monitorar o seu progresso via pequenos testes; e adicionar mais material e bibliotecas eletrônicas para pesquisa de trabalhos do curso.

O administrador coordena e gerencia o ambiente determinando as permissões dos usuários utilizando as ferramentas de administração, comunicação e informação. O LS se foca em uma educação bilíngüe mas, até onde sabe-se, ele não tem sido utilizado para o ensino de uma linguagem de programação para surdos.

### 3.4 COLLABODE

O *Collabode*, *Collaborative Coding in the Browser* (GOLDMAN; LITTLE; MILLER, 2011), é um ambiente web de desenvolvimento integrado para Java projetado para auxiliar trabalhos colaborativos com uma maior eficácia e velocidade na integração do código. O que por sua vez facilita para usuários uma estreita colaboração dos códigos armazenados na Web. *Collabode* (Figura 6) permite que seus usuários tenham as modificações no código quase que simultaneamente para que possa haver uma colaboração eficaz e rápida, da mesma forma que possibilita analisar se os mesmos estão trabalhando em conjunto em uma mesma linha de raciocínio.

Figura 6 – *Collabode* Web IDE colaborativa



Fonte: (GOLDMAN; LITTLE; MILLER, 2011)

Para apoiar várias edições e usuários simultâneos, o *Collabode* usa um editor de código aberto chamado *EtherPad*<sup>1</sup>, possibilitando qualquer quantidade de colaboradores editando um mesmo código em um único momento. Do lado do servidor usa também o Eclipse para o gerenciamento de projetos e serviços de IDE. Devido a esse fato, o ambiente web colaborativo suporta arquivos de projetos em Eclipse.

Uma das características mais marcantes é a sua familiaridade com o Eclipse, assim possibilitando uma grande facilidade para qualquer programador que já tenha tido algum contato com o Eclipse. O código é executado no servidor, assim tendo como ponto negativo o desenvolvimento de programas com interface gráfica de usuários desktop.

Goldman, Little e Miller (2011) examinam o problema da codificação colaborativa em face dos erros de compilação do programa introduzidos por outros usuários, o que torna a colaboração mais difícil devido que o erro persiste para todos usuários que editam o código até que seja resolvido, assim impossibilitando uma compilação e testes de outras partes do código.

O algoritmo e a interface são avaliados por meio de dados gravados a partir de experiências piloto anteriores com *Collabode* e através de um estudo do usuário com o estudante e programadores profissionais. É relatado que o *Collabode* oferece benefícios apreciáveis sobre a simples sincronização contínua sem erros e sobre um controle de versão manual. Embora trate-se de uma IDE online, o *Collabode* a princípio não é voltado propriamente para o ensino, além de não oferecer funcionalidades que sejam acessíveis aos surdos e pessoas com deficiência auditiva.

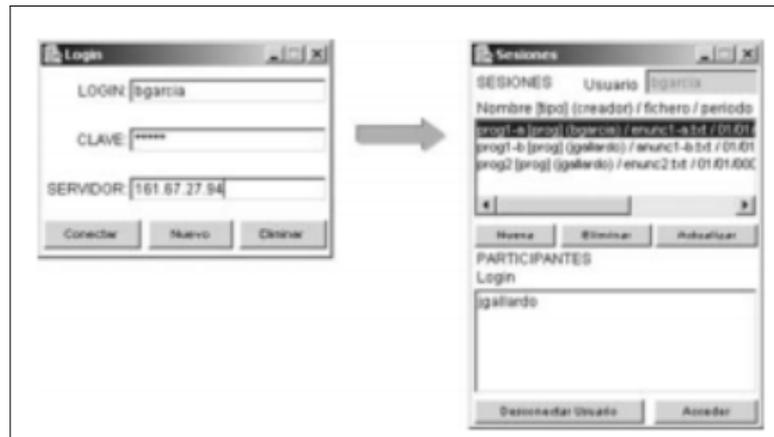
### 3.5 COLLECE

O *COLLECE*, *COLL*aborative *E*dition, *C*ompilation and *E*xecution (DUQUE; BRAVO, 2008), é uma ferramenta colaborativa que permite a programação aos pares distribuída (PPD). Ela foi desenvolvida utilizando a tecnologia Java e opera em uma arquitetura cliente-servidor. Acessível a partir de uma página web, a ferramenta permite a criação de programas a partir de diferentes estações de trabalho e colaboração simultaneamente na resolução da mesma tarefa. Ao se logar no *COLLECE*, o usuário encontra um gestor da sessão disponível (Figura 7). Uma sessão é definida pelo nome, um tipo, um arquivo e um calendário que mostra até que dia a sessão poderá ir. No arquivo contém as informações referentes a formulação da problemática a ser resolvida, e estará sempre disponível para consultas no *COLLECE*. Além disso, fornece

<sup>1</sup> <http://etherpad.org>

um subsistema de análise que armazena em um banco de dados as interações realizadas pelos usuários. Esse repositório de dados permite que o trabalho de cada usuário seja analisado automaticamente. O *COLLECE* possibilita também aos usuários saberem quais são os outros participantes conectados na sessão naquele momento.

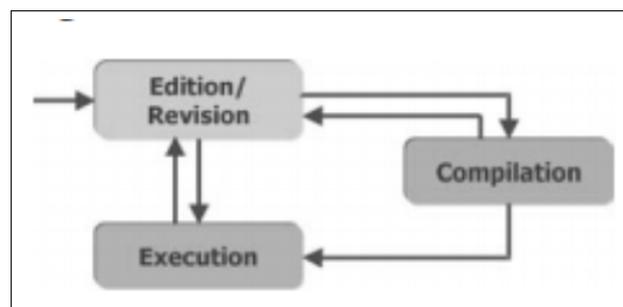
Figura 7 – Login e Gerenciamento da Sessão no *COLLECE*



Fonte: (DUQUE; BRAVO, 2008)

Duque e Bravo (2008) identificam que o protocolo de colaboração é dividido em 3 estágios, pelos quais cada estágio seria uma área de trabalho específica. Esses estágios são: edição/revisão de tarefas, compilação e execução de tarefas (Figura 8).

Figura 8 – Estágios do protocolo de colaboração

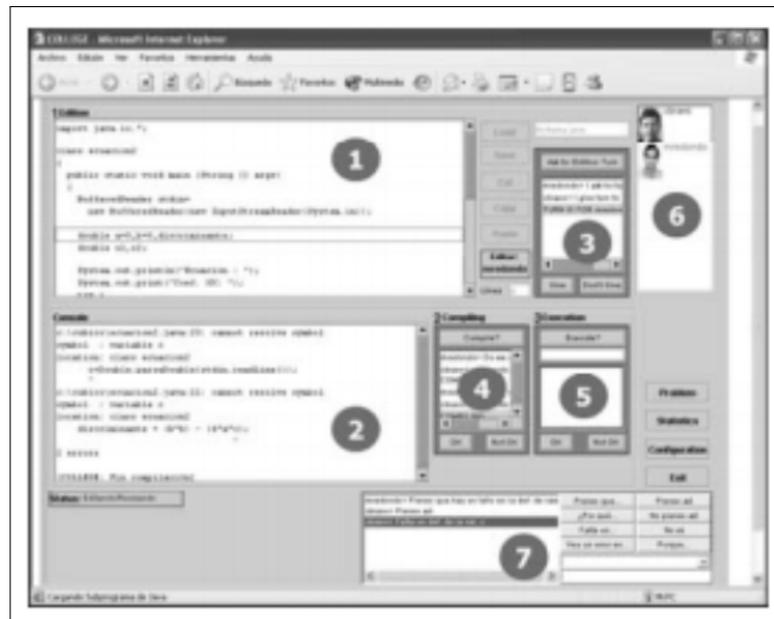


Fonte: (DUQUE; BRAVO, 2008)

Dessa forma, na interface do *COLLECE* tem-se as seguintes áreas de trabalho do sistema: edição / revisão de tarefas (Figura 9-1); compilação e execução das tarefas (Figura 9-2); três áreas para os processos de proposição (Figura 9-3, 9-4 e 9-5); o painel de sessão (Figura 9-6); e o bate-papo estruturado (Figura 9-7). Assim, o sistema tem uma melhor organização para um sistema colaborativo mais complexo.

Apesar de todas as funcionalidades trazidas pelo *COLLECE*, este não é voltado para

Figura 9 – Áreas de trabalho do COLLECE



Fonte: (DUQUE; BRAVO, 2008)

o aprendizado de uma linguagem, logo não possui funcionalidades que, por exemplo, fornecem uma comunicação com um tutor. Além disso, não é uma ferramenta acessível, então não possui recursos que proporcionem a algum usuário surdo uma boa interação com o ambiente.

Todos os trabalhos de pesquisa apresentados neste capítulo influenciaram a proposta desta dissertação de alguma forma. Uma IDE simplificada, focada na necessidade do usuário surdo, o uso maior de elementos visuais e o uso de informação bilíngue devem estar presentes em qualquer plataforma de ensino à distância para surdos. A Tabela 1 mostra alguns aspectos das ferramentas apresentadas.

Tabela 1 – Comparativo das Ferramentas

	<b>Acessível</b>	<b>EAD</b>	<b>Tutoria</b>	<b>Colaborativo</b>	<b>Online</b>	<b>Linguagem</b>
<b>Hands</b>	x					Proglib
<b>iVProg</b>						Programação Visual
<b>Learning System</b>	x	x	x	x	x	
<b>Collabode</b>				x	x	Java
<b>COLLECE</b>				x	x	Java
<b>JLoad</b>	x	x	x		x	Java

No entanto, o objetivo deste trabalho é ensinar outra linguagem para os surdos de

forma remota e assíncrona. Só depois de aprender uma linguagem de programação efetiva, como Java, programadores surdos terão uma chance concreta no mercado de trabalho.

## 4 O OBJETO DE APRENDIZADO JLOAD

Este capítulo apresenta o JLoad, uma ferramenta de ensino com a qual os aprendizes de programação surdos podem criar seus primeiros programas em Java. Os usuários podem realizar oficinas práticas nas quais são assistidos com um passo a passo para orientar a criação, compilação e execução dos códigos, além da possibilidade de tirar dúvidas online na própria ferramenta. O JLoad é integrado a um ambiente virtual de aprendizagem (AVA) denominado Dell Accessible Learning, diminuindo, portanto, o esforço cognitivo do aluno surdo que não precisa, nas aulas iniciais de Java, abrir, instalar e configurar outros programas de criação de códigos que não são acessíveis. Além disso, como todo o conteúdo e atividades práticas realizadas estão disponíveis no AVA, o aluno pode estudar e ter acesso aos seus dados em qualquer lugar com acesso a Internet.

### 4.1 JLOAD - O PROTÓTIPO INICIAL

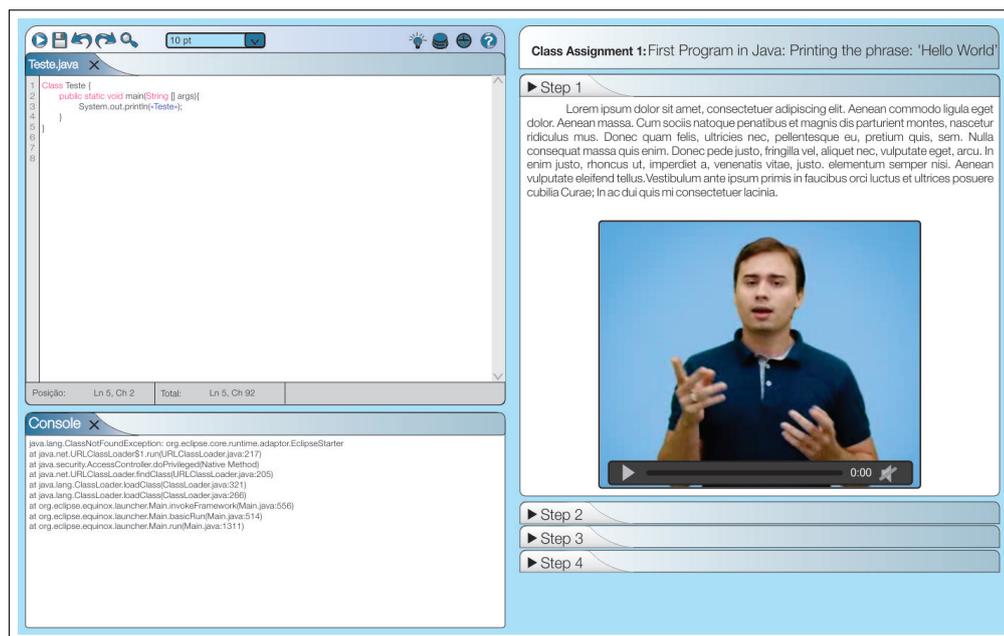
O JLoad permite que os conteudistas dos cursos de EAD do AVA criem oficinas. Uma oficina é constituída por uma lista de atividades de programação. Cada atividade tem seu próprio conjunto de instruções (exibidas em Português e Libras). Normalmente, uma atividade JLoad requer: (a) alguma codificação, compilação e execução de um programa Java; (b) submissão do código para a avaliação; e (3) solicitação e obtenção de ajuda situada - tudo isso realizado dentro do JLoad. Esta abordagem *all-in-one* evita que o aluno tenha que instalar e aprender a usar um ambiente de desenvolvimento integrado como o Eclipse antes de executar seu primeiro programa.

Suportado pelas teorias apresentadas anteriormente, as oficinas de programação do JLoad são divididas em pequenas subtarefas, mais simples, com metas alcançáveis, permitindo um progresso gradual e monitorado. A interface simples permite manter os alunos focados nos objetivos das subtarefas. Os alunos podem enviar perguntas para os tutores através do próprio JLoad. Tais perguntas são acessadas na área do tutor. Todas as informações como: vídeos em Libras, conteúdo das oficinas e dúvidas sobre codificação são armazenadas no repositório do JLoad. O JLoad também é responsável pela compilação e execução dos códigos criados pelo aluno, tudo isso remotamente.

As figuras a seguir retratam os protótipos iniciais do JLoad obtidos a partir de design participativo com surdos e intérpretes (SILVA *et al.*, 2014). A Figura 10 mostra o espaço de trabalho do estudante em uma oficina do JLoad. A codificação e gravação dos códigos é feita na

parte superior do painel esquerdo, que se assemelha a uma IDE simples. As saídas de compilação e execução são mostradas no painel inferior esquerdo. Os alunos podem salvar quantas versões do código-fonte desejarem, dentre as versões salvas, ele pode enviar uma para tutor avaliar como sua resposta para cada etapa da oficina. O painel direito mostra as etapas da oficina, uma de cada vez.

Figura 10 – Espaço de Trabalho do Aluno



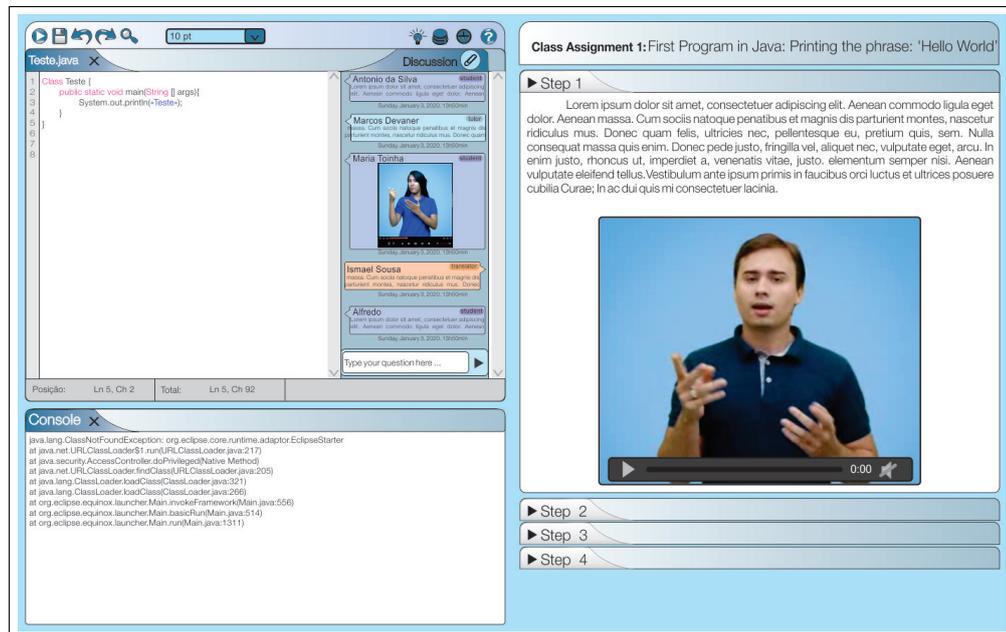
Fonte: A autora.

Como pode ser visto na Figura 11, o aluno pode enviar dúvidas ao tutor sobre a tarefa de uma oficina. O aluno pode ainda destacar a parte do código que ele está em dúvida. O código em destaque pode ser anexado a uma mensagem no chat. O tutor depois de receber uma notificação, utilizando também JLoad, pode inspecionar o código, compilar e tentar executá-lo, ele também pode destacar onde estava o mal-entendido e notificar o aluno novamente.

A Figura 12 mostra o espaço de trabalho do tutor. No painel esquerdo, localiza-se a lista de perguntas dos alunos. Ao clicar em uma dúvida, o tutor tem acesso ao código do aluno (painel central), a um *chat* e as etapas da oficina.

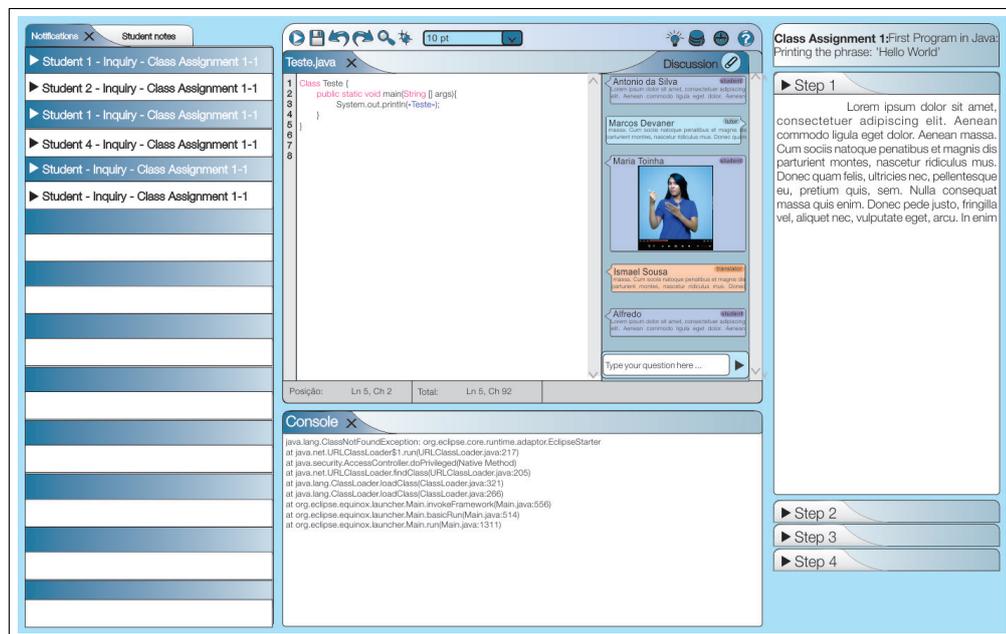
Depois de descobrir o que deu errado, o tutor responde (via *chat*) e, eventualmente, pode notificar o intérprete de Libras para responder adequadamente o aluno que é surdo. A mediação do intérprete de Libras também pode ocorrer quando o aluno envia uma dúvida ao tutor.

Figura 11 – Aluno Submetendo Dúvida ao Tutor



Fonte: A autora.

Figura 12 – Espaço de Trabalho do Tutor



Fonte: A autora.

#### 4.1.1 Design Participativo e Resultados Preliminares

Quatro estudantes surdos (com idade entre 23 e 38 anos) e um tutor (29 anos) foram convidados para as sessões de design participativo. Todos tinham concluído o curso de programação Java online e utilizado a IDE Eclipse nas oficinas do curso.

O engajamento inicial do público alvo usuário desta ferramenta permitiu avaliar

como o JLoad seria recebido pelo aluno surdo interessado em fazer um curso de programação Java, além de definir o fluxo de trabalho e detalhes de interação. A equipe envolvida, com a ajuda de um intérprete de Libras, apresentou o projeto de pesquisa, seus objetivos e finalidades. Os protótipos de baixa fidelidade e o fluxo de trabalho foram apresentados. Os participantes se revezavam interagindo com os *widgets* de tela, cada um elaborou uma sugestão e foram instruídos a explicar como eles usariam o modelo proposto. Essa interação permitiu que os participantes se envolvessem no debate, mediado pelo intérprete de Libras e envolvendo a equipe da pesquisa. Todas as sessões foram filmadas.

Os participantes fizeram contribuições importantes para JLoad, entre elas:

- a) Legendas devem ser utilizadas sempre que não há uma tradução clara para Libras (por exemplo, cálculos e diagramas);
- b) Deve ser possível capturar a tela para enviar junto com a dúvida;
- c) As etapas da oficina escritas em Português devem ter *links* para um glossário Libras (um glossário onde os termos relacionados à computação estão descritos em Libras);
- d) Exportar oficinas do JLoad para o Eclipse;
- e) Os participantes também propuseram muitos arranjos de *widgets* que foram prontamente incorporados à ferramenta.

Os participantes foram informados e responderam a um questionário padronizado na escala Likert, que é um método que possibilita a medição do grau de intensidade de satisfação com o objeto em estudo, foi desenvolvida por Rensis Likert, e é um método bem apurado para obtenção de dados.

Três dos quatro indivíduos concordaram que o JLoad os ajudou a ficar concentrados na tarefa. Apenas um pensa que ele dificultaria uma futura transição para o Eclipse. 75% deles valorizam o fato de que o tutor pode acessar seus programas à medida que eles são escritos. Todos concordam que é importante o fato do tutor ser capaz de compilar e executar seus programas, identificar seus erros/dúvidas e responder às suas perguntas de forma situada.

Todos os participantes concordaram que os vídeos em Libras para descrever as tarefas das oficinas são indispensáveis. Todos concordaram que a proposta melhora a experiência em ensino à distância. Eles também afirmaram que a IDE Eclipse deveria ter algumas das características apresentadas no JLoad. Essa descoberta abre uma possibilidade de construção de um plug-in Eclipse como trabalho futuro.

O uso da técnica de design participativo, envolvendo membros da comunidade, pro-

vou ser extremamente útil para avaliação precoce de conceitos-chave e validação das principais características do JLoad.

Com a proposta original em mãos, o envolvimento precoce da comunidade de usuários foi utilizado para a validação do conceito e avaliação da aceitação. Os usuários chave não só confirmaram a necessidade de tal intervenção, mas também apresentaram importantes contribuições. Os participantes estavam muito animados para trabalhar em suas primeiras oficinas no JLoad. A ferramenta entra, portanto, em processo de construção e depois segue para fase de testes.

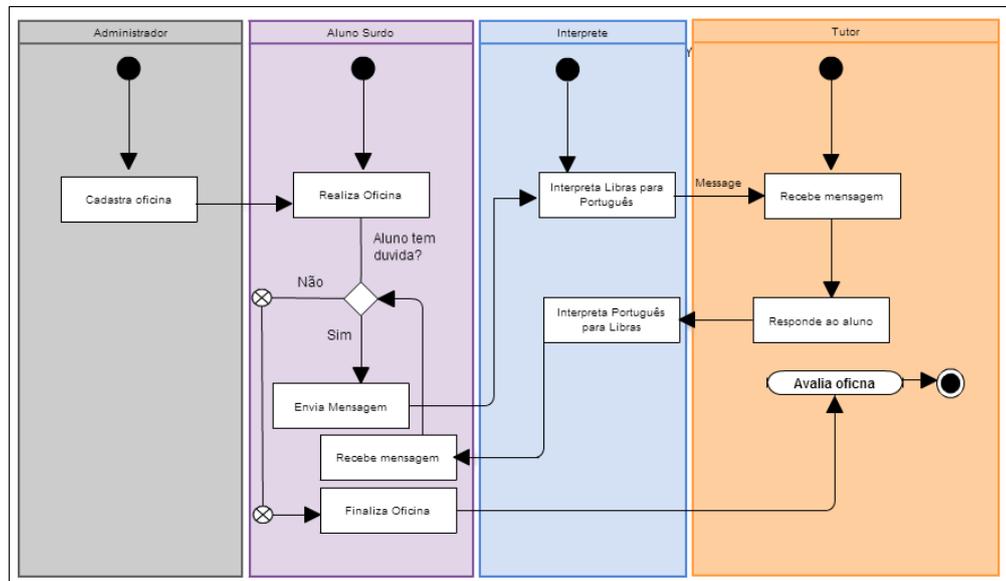
## 4.2 JLOAD - DESENVOLVIMENTO DA FERRAMENTA

Os principais atores envolvidos no JLoad são ilustrados na Figura 13. O autor do curso (através do perfil *Administrador*) é responsável por postar o conteúdo das oficinas no JLoad. A oficina é um dos objetos de aprendizagem que está inserido no AVA, ela é composta por uma atividade de programação dividida em etapas, onde cada etapa tem seu próprio conjunto de instruções (exibido em Libras e em Português). O aluno surdo é responsável por acessar o AVA, em seguida, se direcionar à oficina disponibilizada no JLoad onde realizará a atividade prática de desenvolvimento de software Java. Já o tutor é responsável por acompanhar os alunos a distância e auxiliar na aprendizagem, da melhor forma possível, além de solucionar as dúvidas dos alunos através do *chat*. O Intérprete de Libras auxilia os alunos surdos na tradução das suas dúvidas, assim como na intermediação da comunicação entre o aluno e o tutor Java (Português/Libras). Cada ator envolvido possui um ambiente de trabalho no JLoad, suas interfaces serão apresentadas e detalhadas a seguir.

Conforme as teorias apresentadas, as atividades de programação do JLoad estão divididas em etapas menores e mais simples, com metas alcançáveis, permitindo o progresso gradual e monitorado do aluno. O JLoad apresenta a atividade em etapas permitindo que os alunos foquem em uma etapa por vez.

Ele também possibilita que estes submetam as suas dúvidas aos tutores através do próprio JLoad em português ou Libras. O Tutor, por sua vez, possui também uma interface própria no JLoad onde pode acessar as dúvidas enviadas pelos alunos através do chat, carregar a atividade que está sendo realizada pelo aluno e acessar o seu código.

Figura 13 – Fluxo do JLoad



Fonte: A autora.

#### 4.2.1 Requisitos do Objeto

Os requisitos do objeto foram coletados a partir de entrevistas e da prototipação de baixa fidelidade das telas através de design participativo com os surdos. Os requisitos foram validados junto aos PCDs e demais usuários e especificados em diagramas UML.

A Figura 14 apresenta o diagrama de casos de uso contendo as principais funcionalidades do JLoad para os atores aluno, tutor e intérprete. O aluno pode exercitar seus conhecimentos realizando uma oficina prática. A oficina contém um passo a passo das atividades a serem executadas, seu conteúdo é disponibilizado tanto em Português como em Libras. Além disso, o aluno tem acesso a uma IDE juntamente com um terminal para escrita, compilação e execução de seus códigos.

Mensagens de texto ou vídeo podem ser enviadas ao tutor através de um chat, bem como, a tradução de uma mensagem para Libras ou Português pode ser solicitada ao intérprete. Por fim, recursos de acessibilidade podem ser ativados conforme a necessidade do aluno.

Além de recursos para o aluno, o JLoad disponibiliza interface e funcionalidades para tutor e intérprete que permitem, por exemplo, correção, lançamentos de notas e traduções de mensagens. No Apêndice A, é possível verificar os diagramas de caso de uso detalhados referente aos diferentes atores do sistema.

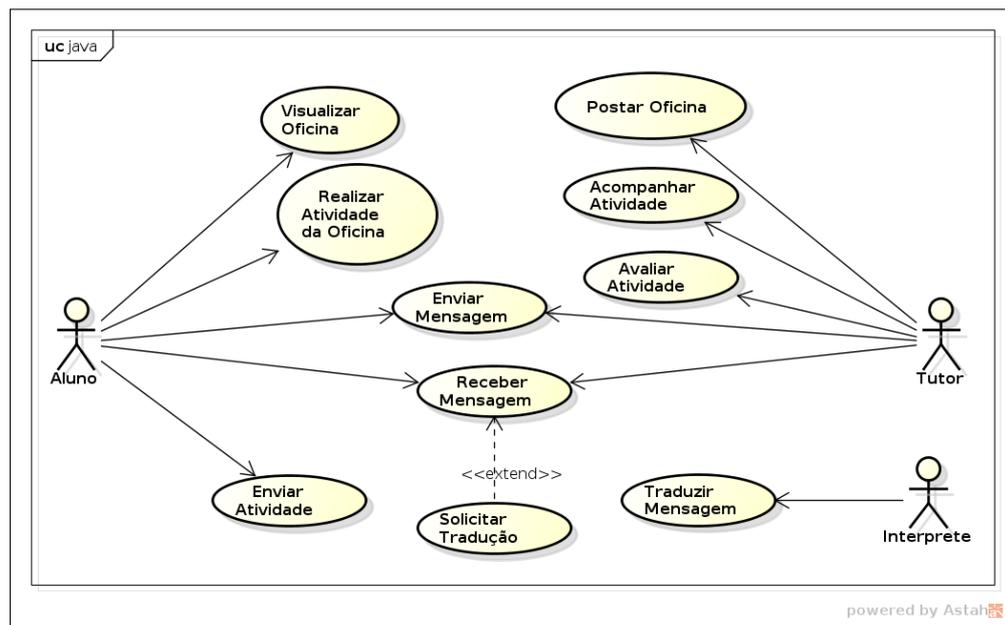
A seguir, são resumidos alguns dos principais requisitos não-funcionais do sistema:

- acessível;

- fácil customização;
- fácil usabilidade; e
- grande armazenamento para vídeo.

Dentre as tecnologias para implementar o JLoad, foram utilizadas: *JavaScript*, *CSS* e *HTML* (no lado do cliente); *JavaServer Pages (JSP)*, *JDK* e *Hibernate* (no lado do servidor).

Figura 14 – Diagrama de Caso de Uso



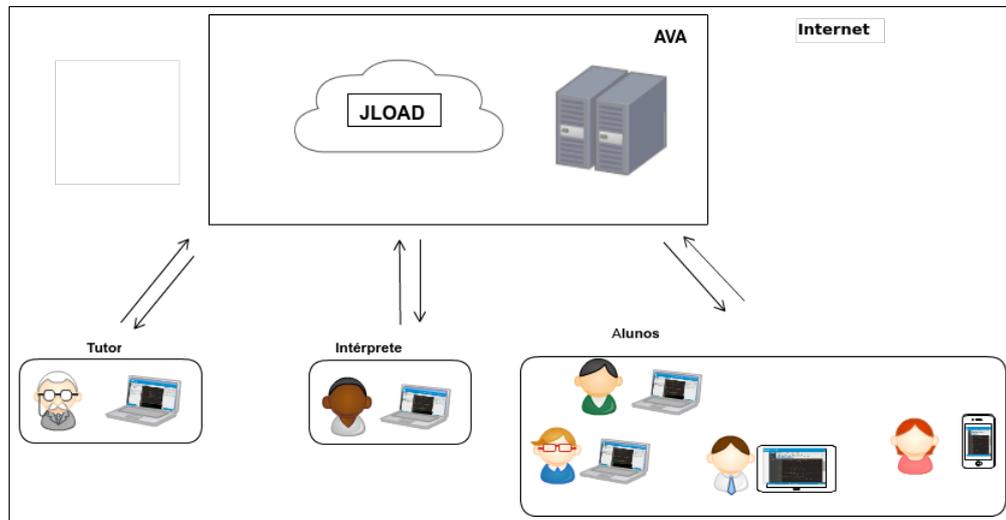
Fonte: A autora.

#### 4.2.2 Arquitetura

O AVA, juntamente com o JLoad, está hospedado em um servidor remoto. Dessa maneira, os usuários da ferramenta podem acessar suas funcionalidades via Internet. O JLoad também foi projetado para ser responsivo e, portanto, pode ser utilizado em diferentes dispositivos como laptops, tablets e outros computadores (Figura 15).

A arquitetura em camadas do JLoad é apresentada na Figura 16. Ela é composta por uma user-interface (UI) acessível por um browser do lado do cliente. No lado do servidor, há o núcleo (*core*) da aplicação com as funcionalidades principais, bem como o componente de persistência onde é executado um banco de dados MySQL contendo os dados dos alunos, tutores, turmas, oficinas, projetos, códigos-fonte desenvolvidos pelos alunos e bibliotecas. Como o JLoad deve ser acessível permitindo a comunicação em Libras, é crucial que a ferramenta se comunique com um serviço de *streaming* (RTP, *Real-Time Transport Protocol*). Nesse componente é

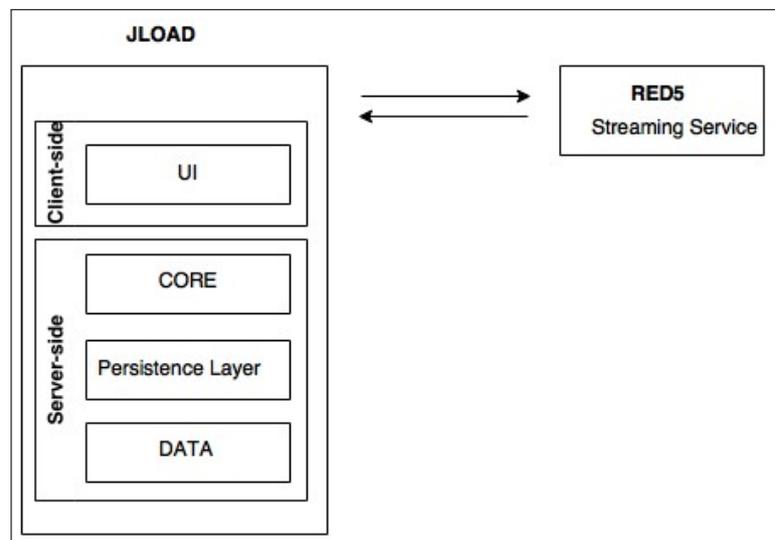
Figura 15 – Acesso ao JLoad



Fonte: A autora.

utilizado o software Red5 para gravação, transmissão e reprodução de vídeos. Os vídeos são utilizados, por exemplo, no passo a passo da Oficina e na comunicação tutor-aluno via *chat*.

Figura 16 – Arquitetura em Camadas do JLoad

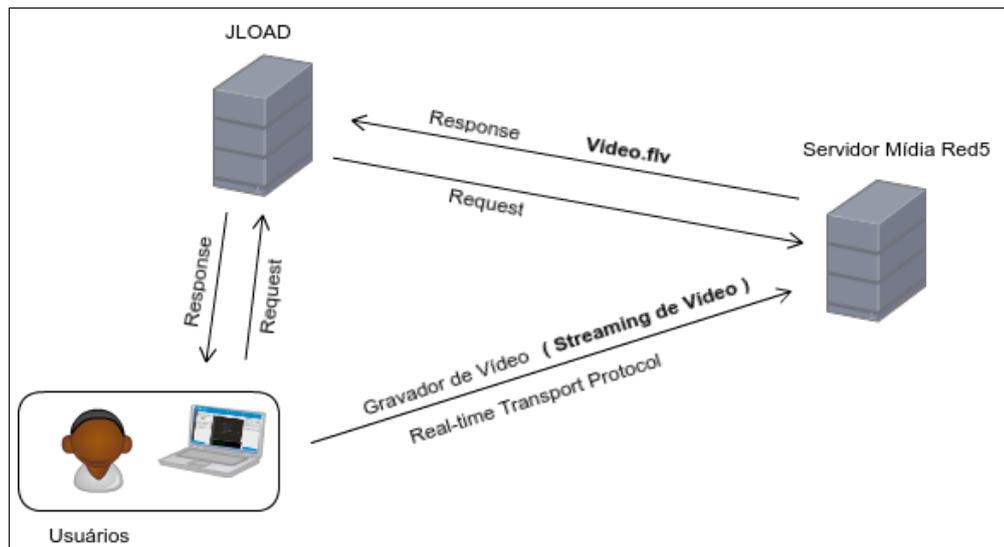


Fonte: A autora.

O armazenamento dos dados gerados no chat, vídeos e códigos, bem como o processo de gravação e transmissão de vídeos, pode exigir uma alta capacidade de desempenho do servidor. Logo, a concentração de todos os componentes da ferramenta em uma mesma máquina pode causar uma eventual sobrecarga da mesma. Devido a isso, o JLoad permite a distribuição desses componentes em mais de um servidor. Pode-se separar, por exemplo, o módulo *core* da aplicação, o módulo de persistência de dados e o serviço RTP em servidores distintos.

Na Figura 17, é ilustrada uma situação onde o *core* da aplicação e o componente de *streaming* estão em servidores distintos. Através da ferramenta de chat, o JLoad executa um gravador de vídeo enviando o fluxo de *streaming* ao Red5 que, por sua vez, o converte em vídeo e, por final, devolve o *link* do vídeo ao JLoad.

Figura 17 – Servidor de Streaming - Red5



Fonte: A autora.

O JLoad foi desenvolvido utilizando o padrão arquitetural MVC, *Model-View-Controller* (KRASNER; POPE *et al.*, 1988), com módulo de transparência de persistência. A arquitetura MVC facilita a divisão e compreensão do código pois promove a separação modelo-visão, possibilitando que as modificações não se espalhem ao precisar fazer alterações na *user-interface*, não alterando regra de negócio.

Por fim, para melhorar o desempenho, o JLoad faz uso de chamadas assíncronas utilizando tecnologias web (AJAX) do lado do cliente. Os recursos AJAX são utilizados, por exemplo, ao carregar os diretórios dos arquivos de projeto, carregar as mensagens no chat e ao analisar o código em tempo de digitação na IDE.

#### 4.3 ACESSIBILIDADE

O JLoad possui uma série de recursos para que pessoas com deficiência possam ter uma experiência de uso eficaz, visando diminuir as barreiras de aprendizado e interação entre alunos e tutores. Esses recursos foram implementados com base nas diretrizes da WCAG 2.0, sendo este um guia de diretrizes propostas pela *World Wide Web Consortium - W3C*, onde são

Tabela 2 – Conformidade WCAG - JLoad

Recurso	Conformidade WCAG
Vídeos em Libras para chat e conteúdo	A
Auto contraste	AA
Redimensionamento de fonte	AA
Teclas de atalho	A
Etiquetas ou instruções	A
Adaptável	A

propostos recursos para que conteúdos na web possam ser acessados e compreendidos por uma gama maior de usuários com deficiência (CONSORTIUM *et al.*, 2008).

As diretrizes propostas na WCAG são classificadas em níveis que apontam os usuários atendidos por cada diretriz. São 3 os níveis de conformidade: (A), (AA) e (AAA). Eles são mapeados conforme os critérios de sucesso convencionados pela WCAG, sendo A o nível mais baixo e o AAA, o mais elevado (CONSORTIUM *et al.*, 2008). Na Tabela 2 são apresentados os recursos disponíveis no JLoad e os seus níveis de conformidade.

A WCAG utiliza alguns validadores de código para verificação de conformidade geral de páginas web. Foi utilizado o validador AccessMonitor<sup>1</sup> para fazer a validação do JLoad. O *link* da ferramenta foi submetido ao validador e verificou-se que o Jload apresenta um nível de conformidade AA . A WCAG não recomenda fixar o Nível AA como requisito normativo para todas as páginas, em alguns casos, não é possível satisfazer todos os critérios de sucesso de Nível AAA (WCAG, 2008).

#### 4.3.1 Espaço de Trabalho do Aluno

Uma oficina consiste na elaboração de atividades práticas de codificação com base no conteúdo apresentado na aula. Elas são elaboradas por conteudistas especialistas no assunto, que dividem uma tarefa em etapas, para que os alunos possam evoluir seus códigos de forma sequencial motivando-se a cada etapa realizada. O JLoad é um objeto de aprendizagem que pode ser aplicado a todas as oficinas, sendo necessário apenas a adaptação do conteúdo conforme o assunto da aula.

Basicamente o espaço de trabalho do aluno é composto pelas atividades da oficina, uma *mini-IDE* (editor de código e console) e o recurso de *chat*, como apresentado na Figura 18. A ideia é que o aluno acesse o conteúdo da oficina por etapas em português e em Libras. Além disso, ele deve utilizar a *mini-IDE* para realizar as tarefas da oficina. Dentre as funcionalidades

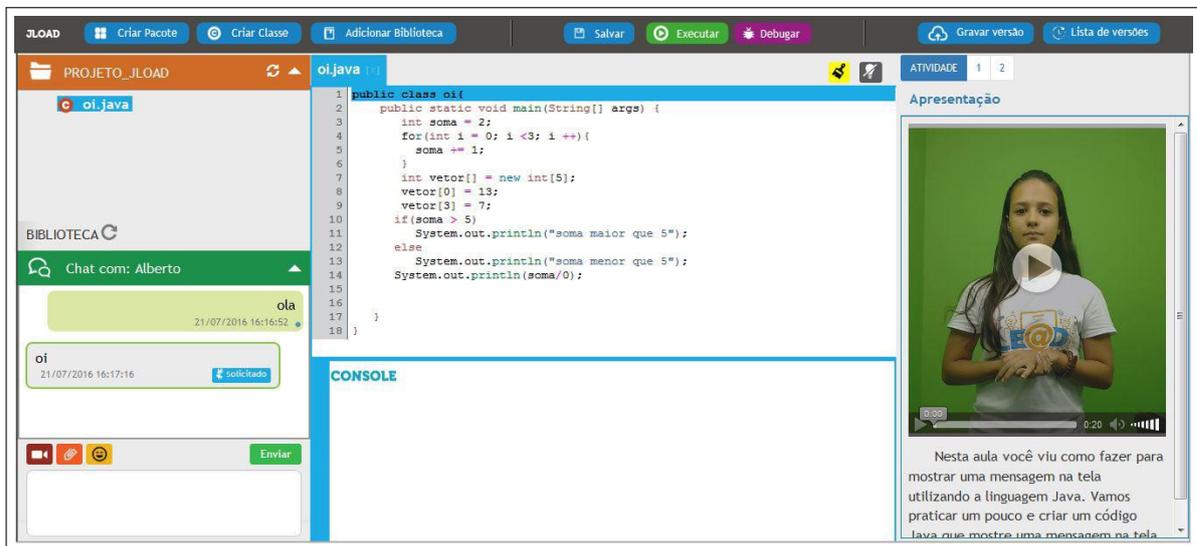
<sup>1</sup> Disponível em <http://www.acessibilidade.gov.pt/accessmonitor>

disponibilizadas para o aluno nesse âmbito tem-se:

- criação de pacotes e classes;
- adição de bibliotecas;
- gravação de uma versão do código; e
- execução do código com visualização do feedback no Console.

Quando o aluno finaliza todas as etapas, ou parte delas, ele envia o resultado de seu trabalho ao tutor para avaliação. Os alunos podem salvar quantas versões do código quiserem, e podem selecionar qual delas apresentar ao tutor como resposta no fim da atividade.

Figura 18 – Espaço de Trabalho do Aluno



Fonte: A autora.

A Figura 19 ilustra outro recurso muito importante no JLoad: o chat do aluno com o tutor. Caso o aluno tenha dúvidas, ele pode utilizar esse recurso para conversar com o tutor em português ou em Libras através de vídeos gravados pela webcam. Também é possível anexar o código à mensagem para que o tutor o veja. Além disso, é possível realizar marcações no código antes de enviá-lo ao tutor para realçar os pontos de dificuldade. O Aluno e o Tutor podem solicitar a tradução das mensagens de Libras ou Português por meio de opção disponível na mensagem recebida no *chat*. Neste momento o Intérprete recebe notificação das solicitações.

O recurso do chat ajuda a promover a presença social do tutor. O conceito está diretamente ligado a interação colaborativa, isto é, a interação entre os participantes é essencial para atingir um determinado objetivo. Portanto, o chat se apresenta como uma alternativa eficiente para retirar as dúvidas dos alunos durante o processo de resolução da oficina. Com uma melhor interação aluno-tutor, espera-se uma melhora nos indicadores de motivação, já que o alto

nível de presença social propicia um ambiente mais agradável e acolhedor para o aluno.

Figura 19 – Chat entre aluno e tutor



Fonte: A autora.

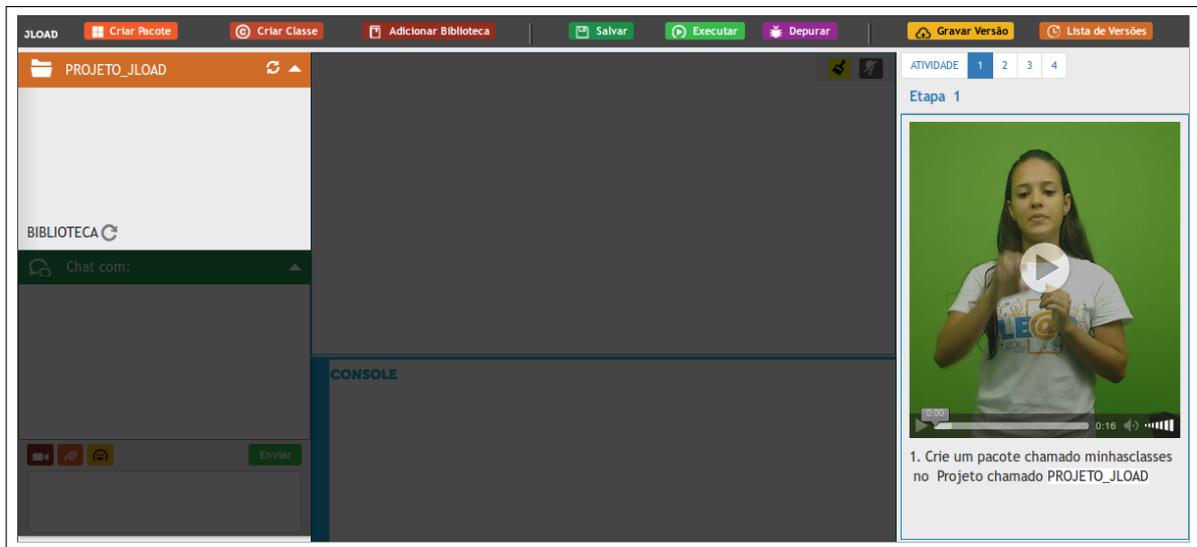
Uma ferramenta normalmente deve prover muitas funcionalidades para atender às diversas necessidades de seus usuários, mas ela também deve apresentar-se de forma simples pois os mesmos usuários não dispõem de tempo para aprender a usar todos os recursos ofertados. Nesse contexto, a técnica de *Progressive Disclosure* (NIELSEN; BUDIUI, 2013) é utilizada no espaço de trabalho do aluno.

No *Progressive Disclosure*, inicialmente é mostrado aos usuários apenas um pouco dos principais recursos. Os recursos secundários são apresentados apenas por demanda. O fato de um recurso ser apresentado assim que a ferramenta aparece mostra ao usuário que aquele recurso é importante. Usuários iniciantes podem então priorizar sua atenção, evitando erros e economizando tempo ao lidar com recursos avançados que talvez não fossem necessários no momento. Dessa maneira, a técnica melhora o aprendizado da ferramenta, a eficiência de uso e reduz a taxa de erro.

A Figura 20 ilustra o uso da técnica. Note que as regiões onde se quer que o usuário foque sua atenção são apresentadas de forma mais clara, à medida que ele vai utilizando a ferramenta, novas regiões vão sendo apresentadas de forma progressiva.

Em suma, como pode-se ver na Figura 21, o espaço de trabalho do aluno pode ser subdividido em quatro partes, denominadas áreas funcionais: (1) Menu e Área de Navegação; (2)

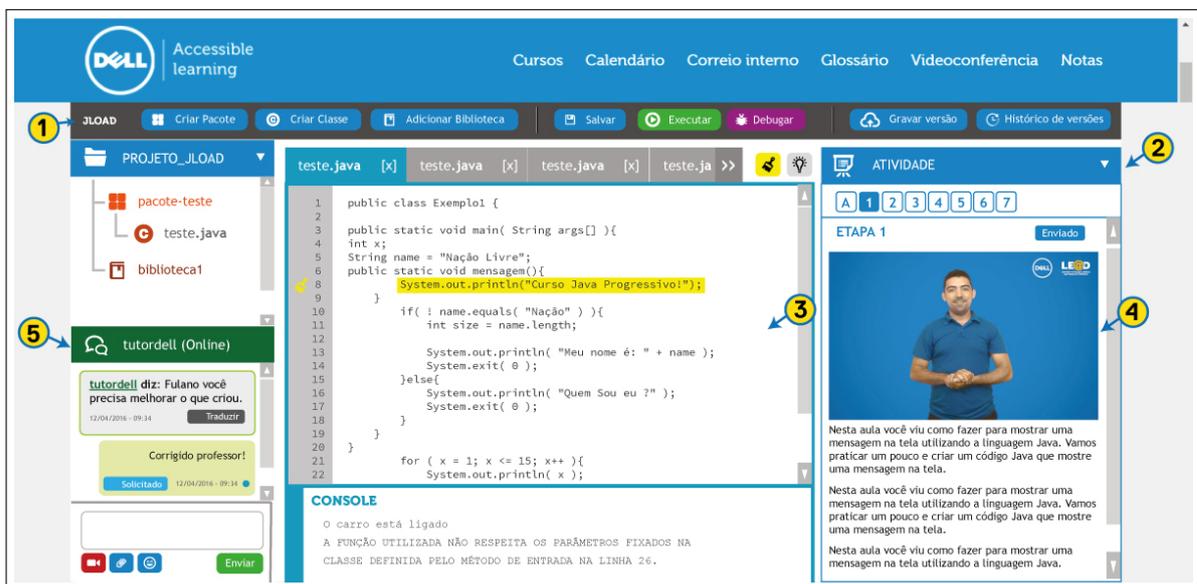
Figura 20 – Progressive Disclosure



Fonte: A autora.

e (4) Detalhamento da Oficina; (3) Desenvolvimento e Feedback; e (5) Interação Tutor-Aluno.

Figura 21 – Visão Geral Áreas Funcionais do Aluno



Fonte: A autora.

A seguir, as funcionalidades principais de cada uma das áreas é descrita.

**(1) Menu e Área de Navegação:** Nessa área funcional o aluno tem acesso às principais funções do JLoad. Ela é composta por uma barra de menu e uma área de navegação por projeto onde é possível acessar os pacotes e classes criadas.

**(2) e (4) Detalhamento da oficina:** Na lateral direita do JLoad existe o detalhamento da oficina. Através de um passo a passo, o aluno pode saber qual atividade irá desenvolver por

etapa, toda a oficina também possui vídeo de Libras para auxiliar a compreensão dos alunos.

**(3) Desenvolvimento e Feedback:** O JLoad disponibiliza ao aluno uma série de oficinas desenvolvidas por um professor especialista em programação, para que o aluno possa exercitar e fixar melhor os conteúdos estudados. O JLoad implementa um ambiente de desenvolvimento, que pode ser visto na Figura 21 ao centro. O aluno pode escrever os códigos de resposta aos exercícios propostos sem precisar sair do AVA para isso. Com uma navegação em abas, essa área funcional permite ao aluno um melhor controle na execução das atividades. Também é disponibilizado um console de saída, que mostra o resultado do processo de compilação remota do código, bem como as saídas do programa gerado, caso a compilação obtenha êxito.

**(5) Interação Tutor-Aluno:** Localizada abaixo da área de navegação. A área de interação é composta por um chat que permite a comunicação entre o aluno que está fazendo os exercícios e seu tutor. Dessa forma, caso o tutor esteja *online*, é possível tirar dúvidas em tempo real, potencializando seu aprendizado. O chat permite, além do diálogo por texto, enviar vídeos, possibilitando a comunicação com LIBRAS.

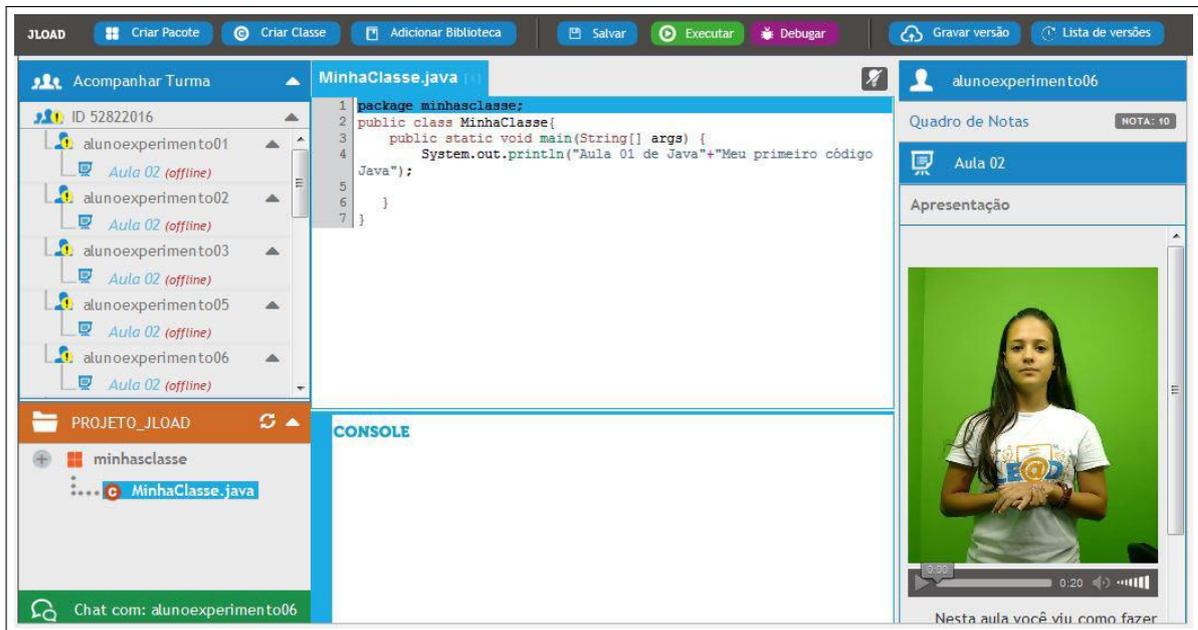
### 4.3.2 Espaço de Trabalho do Tutor

O ambiente de trabalho do Tutor é bastante similar ao do aluno, sendo diferente no tocante à listagem de alunos, como pode ser visto na Figura 22. No painel à esquerda estão listados as turmas que são acompanhadas pelo Tutor e seus respectivos alunos com as atividades que estão sendo realizadas por eles.

Ao clicar na atividade de um aluno, automaticamente são carregados os dados da atividade do aluno e o seu histórico de chat com o mesmo. O tutor então pode acessar o código do aluno no painel central, podendo interagir com ele através do chat, além de poder observar as etapas realizadas da oficina.

No momento em que o aluno envia uma dúvida pelo chat, o tutor pode analisar diretamente o código, visualizando as marcações, assim como pode alterá-lo e executá-lo. Dessa forma, pode identificar mais facilmente os possíveis problemas do código e assim promover uma assistência situada ao aluno. Vale ressaltar que, quando o tutor carrega um código, ele acessa uma cópia do mesmo, ficando a versão do aluno intacta. O tutor também pode também solicitar a mediação do intérprete nas traduções das mensagens.

Figura 22 – Espaço de Trabalho do Tutor



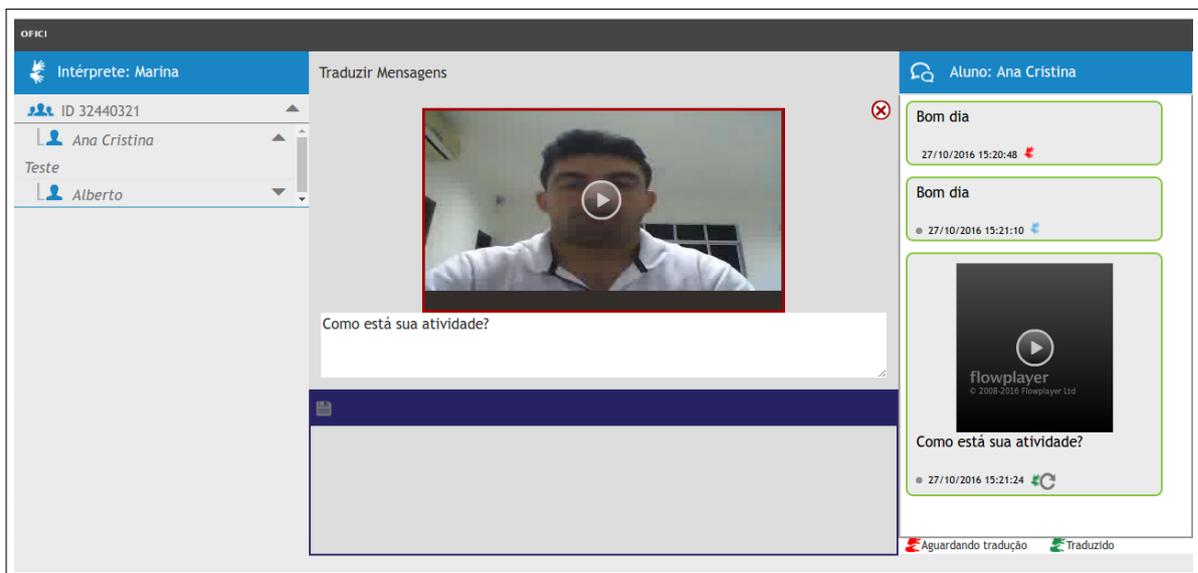
Fonte: A autora.

### 4.3.3 Espaço de Trabalho do Intérprete

O ambiente de trabalho do intérprete é bastante similar ao do tutor, sendo diferente somente a área de desenvolvimento, como é a apresentado na Figura 23. No painel à esquerda estão listadas as turmas que são acompanhadas pelos tutores e seus respectivos alunos juntamente com as solicitações de interpretações que foram efetuadas por eles.

Ao clicar no nome de um aluno, automaticamente são carregadas as solicitações existentes na conversa daquele aluno com o tutor. Normalmente, os alunos solicitam interpretação para Libras de um texto em Português enviado pelo tutor. Já o tutor solicita interpretação para Português de um vídeo em Libras enviado pelo aluno.

Figura 23 – Espaço de Trabalho do Intérprete



Fonte: A autora.

## 5 AVALIANDO O IMPACTO DO JLOAD NAS OFICINAS

O presente capítulo discorre sobre o desenho do experimento realizado com o JLoad. Fala-se brevemente sobre um dos tipos de experimentos mais comumente utilizados na área de IHC (Interação Humano Computador), o intra-sujeitos (*within-subjects*), e explica-se quando ele é utilizado. Em seguida, o tamanho da amostra, as etapas do experimento, as ferramentas e as técnicas utilizadas na sua execução são apresentadas.

### 5.1 EXPERIMENTO INTRA-SUJEITOS

Em um experimento intra-sujeitos, cada participante contribui realizando todas as condições da pesquisa. Caracteriza-se como um tipo de experimento que necessita de uma amostra com tamanho menor, no entanto, são necessários mais testes para cada participante (MACKENZIE, 2013). Normalmente, é realizado contrabalanceamento para diminuir os efeitos de interferência no participante de uma condição de teste para a outra.

### 5.2 EXPERIMENTO INTRA-SUJEITOS

A avaliação de usabilidade das ferramentas, ao final do experimento, foi obtida a partir de um instrumento para coleta de dados que consiste em um questionário adaptado de Lewis (1995), o *Post-Study System Usability Questionnaire* (PSSUQ). Ele foi escolhido devido ao uso de medidas padronizadas. Esses questionários são amplamente utilizados para medir a satisfação do usuário com produtos interativos. O preenchimento do PSSUQ por usuários fornece uma avaliação subjetiva global do sistema que utilizaram (LEWIS, 1992).

O questionário, como pode ser visto no Anexo A, é composto por 19 itens, onde cada um deles possui uma escala Likert que varia de 1 (concordo fortemente) a 7 (discordo fortemente). Os itens 1 a 19 fornecem um fator global de satisfação do usuário. Além disso, é possível avaliar fatores de qualidade a partir de um subconjunto de itens:

- Usabilidade do Sistema (SysUse): dos itens 1 a 8;
- Qualidade da informação (InfoQual): dos itens 9 a 15;
- Qualidade da interface (IntQual): dos itens 16 a 18.

A análise quantitativa da avaliação se dá pela média das respostas dos itens de cada fator. Tem-se, portanto, quatro pontuações: a média geral de satisfação (média das respostas de todos os itens), SysUse, InfoQual e IntQual.

Para a análise dos dados é importante ressaltar alguns pontos com relação às especificidade do PSSUQ. Entre eles está o fato de que a média para obter os resultados não afeta as propriedades estatísticas das pontuações, sendo assim possível comparar e interpretar os dados. Caso o usuário opte por não responder a um item, esse fato não inviabiliza e nem compromete os resultados, pois um item não faz parte de um fator, se um item de um fator não for respondido a média será realizada apenas com os itens respondidos. Os itens são compostos por questões positivas. Logo, ao escolher, por exemplo, o valor 1 na escala, ele está concordando fortemente, com isso define-se que valores menores são avaliados positivamente (LEWIS, 1993).

### 5.3 DIVISÃO DOS GRUPOS

O experimento a ser realizado tem caráter avaliativo e validador, pois foram desenvolvidas atividades práticas com o uso das ferramentas. Nesta etapa da pesquisa participaram 15 surdos, um tutor e uma intérprete de Libras.

Utilizou-se contrabalanceamento de forma randômica na entrada onde cada participante é associado a um número identificador aleatório, que o identifica e diz a ordem das condições de teste avaliadas. Esse contrabalanceamento é apresentado na Tabela 3.

Tabela 3 – Contrabalanceamento de Forma Randômica na Entrada

<b>Identificador</b>	<b>Condição de Teste 1</b>	<b>Condição de Teste 2</b>
01	JLoad	Eclipse
02	Eclipse	JLoad
03	JLoad	Eclipse
04	Eclipse	JLoad
05	JLoad	Eclipse
06	Eclipse	JLoad
07	JLoad	Eclipse
08	Eclipse	JLoad
09	JLoad	Eclipse
10	Eclipse	JLoad
11	JLoad	Eclipse
12	Eclipse	JLoad
13	JLoad	Eclipse
14	Eclipse	JLoad
15	JLoad	Eclipse

Com o objetivo de não tornar o experimento muito cansativo, a primeira condição de teste é aplicada em um dia e a segunda condição em outro. Além disso, os participantes com identificador ímpar serão designados como Grupo 1, ou seja, aqueles que iniciam com o JLoad;

os demais constituem o Grupo 2.

**Grupo 1:**

- Dia 01: JLoad.
- Dia 02: Eclipse.

**Grupo 2:**

- Dia 01: Eclipse.
- Dia 02: JLoad.

#### 5.4 FASES DO EXPERIMENTO

Antes do experimento, é realizada uma breve apresentação dos objetivos do estudo, da confidencialidade da informação recolhida, bem como do caráter voluntário da participação.

- **Fase 1:** os participantes respondem um questionário de perfil e também participam de uma aula que aborda os princípios de lógica e orientação a objetos com a linguagem de programação Java. Essa aula é realizada à distância utilizando os recursos de um AVA e conta com o auxílio de tutor. Cada aluno realiza a Aula 02 do curso de Java básico onde são apresentados os conceitos iniciais da linguagem;
- **Fase 2:** nesta fase, os participantes realizam a oficina referente à aula assistida na fase anterior. A ferramenta utilizada na Fase 02 é aquela que corresponde à primeira condição de teste conforme o identificador do sujeito. Em seguida, são aplicados os questionários pós experimento de motivação e usabilidade referentes à ferramenta utilizada;
- **Fase 3:** os participantes realizam a outra oficina com a ferramenta que corresponde a segunda condição de teste conforme seu identificador. Em seguida, são aplicados os questionários pós experimento de motivação e usabilidade referentes a ferramenta utilizada e ao final do segundo dia um questionário comparativo entre as duas ferramentas utilizadas.

As oficinas (atividades práticas de desenvolvimento, nas quais o aluno exercita os conceitos aprendidos) utilizadas no experimento podem ser vistas no Apêndice B deste documento.

## 5.5 HIPÓTESES

O estudo tem por objetivo comparar o desempenho (tempo de execução, número de auxílios solicitados e notas) dos sujeitos de pesquisa na execução das oficinas, o que inclui a IDE e consultas ao conteúdo que está presente no AVA, realizado por alunos iniciantes surdos ao utilizarem duas ferramentas: o JLoad e o Eclipse.

### 5.5.1 Hipótese Geral

- Observa-se aumento nos indicadores gerais de participação de um aluno resolvendo oficinas em EAD, além de menor tendência à evasão ao curso de Java básico.

### 5.5.2 Hipótese Corolária Geral

- Observa-se aumento nos índices de motivação para aprender também dentre os alunos ouvintes que utilizarem o JLoad.

### 5.5.3 Hipóteses Específicas

- Os níveis de motivação são maiores utilizando o JLoad. A motivação é avaliada por meio da Escala de Motivação Situacional (SIMS) (FORTIER; VALLERAND; GUAY, 1995);
- O índice de rendimento acadêmico obtido através das notas na resolução das oficinas de programação desenvolvidas pelos alunos que utilizam o JLoad são superiores às notas dos alunos que usam o método tradicional (Eclipse). As notas são obtidas através das avaliações das oficinas realizadas nas ferramentas.
- Os tempos para finalização da oficina pelos alunos são menores ao utilizarem o JLoad que ao utilizar o Eclipse.
- O nível de engajamento dos alunos surdos, medido através de interações entre aluno e tutor será maior utilizando a ferramenta JLoad.

## 5.6 ANÁLISE SITUADA

As pessoas normalmente criam planos de ação para certas ocasiões. Muitas vezes elas precisam mudar esses planos, dependendo do que realmente está acontecendo em uma

situação específica. Eles usam suas habilidades incorporadas ou experiências passadas para levá-los através da situação. Ao invés de realizar uma análise longe das circunstâncias, a abordagem de análise situada também considera as circunstâncias e o contexto para alcançar uma ação inteligente (SUCHMAN, 1987).

A seguir, são definidas as técnicas de análise situada utilizadas durante o experimento:

- **Gravação de vídeos:** todas as atividades realizadas pelos participantes do experimento foram gravadas através do de uma ferramenta do tipo *screen capture*, que possibilita a captura e gravação de tudo que aparece no monitor do computador, permitindo um registro real do que está acontecendo.
- **Observação síncrona no momento da oficina:** todas as interações dos alunos foram gravadas, seu desempenho e suas dificuldades, fazendo análise disso.

## 5.7 INSTRUMENTOS DE COLETA DE DADOS

São aplicados três instrumentos para a coleta de dados dos participantes:

- **Questionário Sociodemográfico:** informações de natureza demográfica e acadêmica são solicitadas na Fase 1 do experimento para traçar o perfil dos participantes (nome, idade, gênero, grau de instrução, tipo e grau de surdez, oralizado, nível de compreensão de português, experiência com eclipse, conhecimento Java).
- **Escala de Motivação Situacional (FORTIER; VALLERAND; GUAY, 1995):** aplicado ao final das Fases 2 e 3 do experimento, tendo como objetivo avaliar os níveis de motivação de cada participante;
- **Questionário de Usabilidade PSSUQ (LEWIS, 1995):** aplicado ao final das Fases 2 e 3 com o objetivo de avaliar a usabilidade de cada ferramenta.
- **Questionário comparativo entre JLoad e Eclipse:** aplicado ao final do experimento, onde os sujeitos comparam as ferramentas.

### 5.7.1 Aplicação da Escala de Motivação Situacional (SIMS)

O questionário é aplicado ao final das Fases 2 e 3, posterior à realização da oficina. Ele é preenchido no momento em que o participante finaliza a atividade, de forma a garantir a fidelidade das respostas em relação ao seu envolvimento no experimento e com o objetivo final de averiguar a motivação do usuário ao utilizar a ferramenta.

Após a coleta, os questionários são numerados com o identificador de cada participante e as suas respostas introduzidas numa base de dados.

O questionário dispõe de tradução em LIBRAS em forma de vídeo para todo conteúdo textual de seus 14 itens (FORTIER; VALLERAND; GUAY, 1995). O nível de concordância com o item é apresentado em forma de escala numérica, de 1 a 7, onde o número 1 representa “*não concordo*” e o número 7, “*concordo totalmente*”, como pode ser visto na Figura 24.

Figura 24 – Exemplo de Item com Tradução em LIBRAS e Escala de Concordância

1 - Porque considero que esta atividade seja interessante.



Pergunta \*

Não concordo.

1 2 3 4 5 6 7

Concordo totalmente.

Fonte: A autora.

Ao final do experimento, cada um dos 15 participantes possuirá pontuações referentes a cada uma das dimensões da motivação (Motivação Intrínseca, Regulação Identificada, Regulação Externa e Amotivação) e uma pontuação de motivação global. Totalizando em 10 pontuações por participante, 5 para cada ferramenta.

Por fim, a depender da normalidade dos dados verificada através do teste Shapiro-Wilk (SHAPIRO; WILK, 1965), as pontuações podem ser submetidas ao teste não paramétrico *Wilcoxon Signed-Rank* ou a um teste estatístico paramétrico como *ANOVA* ou o *t-test pareado*.

### 5.7.2 Aplicação do Questionário de Usabilidade para sistema pós teste (PSSUQ)

O Questionário PSSUQ foi utilizado para avaliar a satisfação do usuário com relação ao JLoad e ao Eclipse. Cada participante fornece, portanto, ao final da Fase 3, oito pontuações,

quatro referentes ao JLoad e quatro referentes ao Eclipse, conforme Tabela 4.

Tabela 4 – Pontuações do PSSUQ

<b>Fatores de Usabilidade</b>	<b>Pontuação</b>	
	<b>JLoad</b>	<b>Eclipse</b>
Satisfação Geral (Média Geral)		
Utilidade do Sistema (SysUse)		
Qualidade das Informações (InfoQual)		
Qualidade da Interface (IntQual)		

As pontuações do PSSUQ também podem ser submetidas ao teste *Wilcoxon Signed-Rank*. E, em caso de normalidade dos dados, ao *ANOVA* ou o *t-test pareado*.

### 5.7.3 Aplicação do Questionário Comparativo ao Final do Experimento

Ao final do experimento, para finalizar o segundo dia, o sujeito de pesquisa responde a perguntas, onde será possível comparar como ele analisa as duas ferramentas utilizadas: JLoad e Eclipse. As perguntas estão em escala Likert de 01 a 05, de discordo totalmente a concordo totalmente. Todas as perguntas possuem vídeo de Libras corresponde ao texto português/Libras.

Perguntas do questionário comparativo das ferramentas:

- a) O JLoad é fácil de usar.
- b) O Eclipse é fácil de usar.
- c) Me comuniquei facilmente com o tutor no JLoad.
- d) Me comuniquei facilmente com o tutor pelo AVA sobre o Eclipse.
- e) Terminei a oficina mais rápido no JLoad.
- f) Terminei a oficina mais rápido no Eclipse.
- g) O JLoad me deixou cansado ou entediado.
- h) O Eclipse me deixou cansado ou entediado.
- i) Eu estaria motivado a fazer outra oficina no JLoad.
- j) Eu estaria motivado a fazer outra oficina no Eclipse.
- k) O JLoad possui recursos para os surdos.
- l) O Eclipse possui recursos para os surdos.

Ressalta-se que os participantes utilizam o correio interno do AVA para se comunicar com o tutor quando a condição de teste é o Eclipse.

Com a aplicação desse questionário, pretende-se avaliar os seguintes aspectos: facilidade de usar a ferramenta; fácil comunicação com o tutor na ferramenta; término mais rápido

da oficina; se a ferramenta deixou o usuário cansado ou entediado; motivação para fazer outra oficina na ferramenta; e se a ferramenta possui recursos para surdos.

## 6 RESULTADOS OBTIDOS

Neste capítulo, o resultado será descrito uma contextualização acerca da contribuição deste trabalho. Em seguida, mostraremos os dados para validação das hipóteses desta pesquisa. Por fim, será realizada uma análise crítica dos dados e proposta de evolução.

### 6.1 PERFIL DOS PARTICIPANTES

O instrumento de coleta dos dados teve como resultado o perfil das 15 pessoas que participaram do experimento, todas elas deficientes auditivas. Os sujeitos de pesquisa convidados se inscreveram no AVA para realizar o processo seletivo e fazer o Curso de Lógica de Programação, ou seja, sujeitos interessados e motivados a adquirir conhecimento na área desenvolvimento de software.

A faixa etária dos sujeitos de pesquisa era entre 16 e 29 anos. O público masculino era de 11 pessoas (73,3%), e feminino de 4 pessoas (26,7%). Observou-se também que 10 pessoas (66,7%) têm ensino médio, 4 (26,7%) ensino superior e 1 (6,7%) o ensino fundamental. Dos 15 participantes, 11 (73,3%) tem surdez congênita, 3 (20%) possui surdez adquirida e 1 (6,7%) não sabe informar. Sobre o grau da surdez, 10 (66,7%) possuem surdez profunda, 3 (20%) moderada, 1 (6,7%) severa e 1 (6,6%) leve. Dos participantes, 7 (53,3%) disseram que não são oralizados, ou seja, se expressam predominantemente através da Libras e que não falam outra língua, e 8 (46,7%) são oralizados, portanto tem bom conhecimento e compreensão do Português.

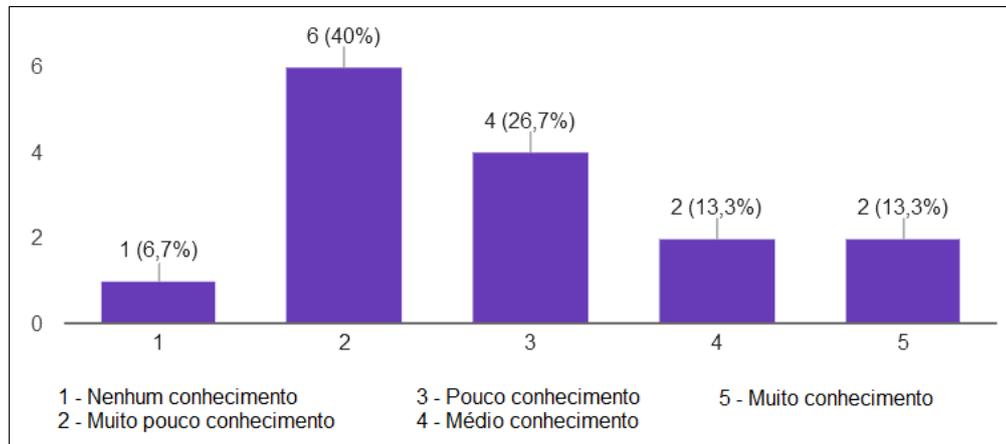
Dentro do questionário aplicado, algumas perguntas procuravam coletar o nível de conhecimento dos participantes. As respostas de tais perguntas utilizam a escala likert, onde apenas uma das opções é escolhida: 1 (nenhum conhecimento); 2 (muito pouco conhecimento); 3 (pouco conhecimento); 4 (médio conhecimento); e 5 (muito conhecimento). Os gráficos a seguir apresentam os resultados obtidos.

A Figura 25 mostra o nível de compreensão de texto em português numa escala de 1 a 5. De acordo com os dados, percebe-se que a maioria dos participantes são surdos cujo maioria tem pouca fluência com textos em português.

A Figura 26 diz nível de conhecimento de Eclipse também em uma escala de 1 a 5. De acordo com os dados, percebemos que a maioria dos participantes não tem experiência com a ferramenta Eclipse.

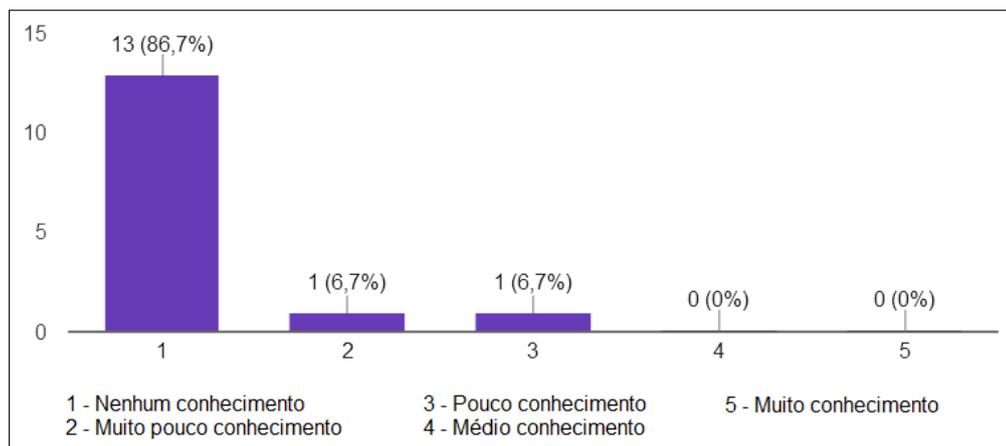
Já a Figura 27 mostra o nível de conhecimento em Java numa escala de 1 a 5. De

Figura 25 – Nível de Compreensão de Texto em Português



Fonte: A autora.

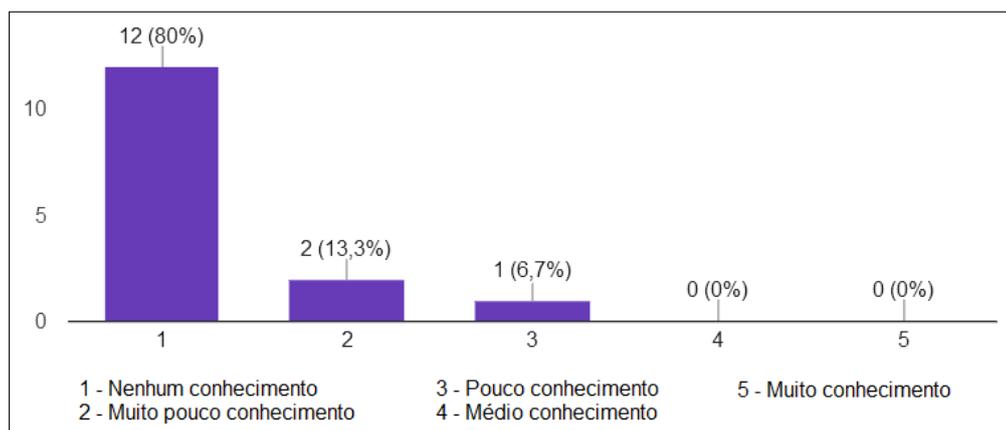
Figura 26 – Nível de Conhecimento de Eclipse



Fonte: A autora.

acordo com os dados, percebe-se que a maioria dos participantes não possuem conhecimento na área de Java.

Figura 27 – Nível de Conhecimento de Java



Fonte: A autora.

Portanto, a partir da análise desses resultados, é possível perceber que os usuários participantes do experimento se adequam ao perfil ideal de usuário final do JLoad, tendo em vista que a ferramenta se preocupa em tornar acessível o aprendizado da área de programação em Java para alunos surdos em todos os níveis de conhecimento e fluência linguística bilíngue (Português e Libras). A partir dos testes realizados com surdos inexperientes na área, porém interessados nesse aprendizado, podemos colher resultados confiáveis e ponderados a respeito da ferramenta.

## 6.2 USABILIDADE

A aplicação do questionário PSSUQ permitiu fazer um comparativo da satisfação dos participantes em relação a usabilidade das ferramentas JLoad e Eclipse. Ao responder o questionário com questões relacionadas a Utilidade do Sistema (*SysUse*); Qualidade da Informação (*InfoQual*); e Qualidade da Interface (*IntQual*), foram geradas médias por categoria e uma média geral, apontando o grau de satisfação em relação a usabilidade das ferramentas. Além de fornecer um *feedback* de fatores importantes do sistema como utilidade, qualidade da informação e qualidade de interface.

Ou seja, ao final do experimento, a aplicação do questionário permitiu gerar, para cada participante, quatro escores de usabilidade (*SysUse*, *InfoQual*, *IntQual* e Média Geral) para o JLoad e quatro escores para o Eclipse, totalizando oito escores por sujeito de pesquisa.

### 6.2.1 Análise Quantitativa

Primeiramente, as diferenças entre as métricas foram submetidas ao teste de normalidade Shapiro-Wilk (SHAPIRO; WILK, 1965). Após esse passo, a normalidade dos dados foi assumida e aplicou-se o teste estatístico paramétrico *t-student* (ou simplesmente teste *t* pareado) com objetivo de verificar se existem diferenças significativas entre os resultados das duas ferramentas.

Vale ressaltar que no *t-student* os resultados com valor de  $p \leq 0,05$  apresentam diferença significativa. Os resultados da aplicação do PSSUQ são apresentados na Tabela 5. Além disso, nesse questionário, quanto menor o valor obtido, melhor a avaliação.

As médias dos indicadores já registram um melhor resultado para o JLoad, mostrando a preferência dos participantes por esta ferramenta. Além disso, em três das quatro métricas, os resultados apresentam diferença significativa.

Tabela 5 – Média das Métricas de Usabilidade

Métricas de Usabilidade	Pontuação Média (quanto menor, melhor)		t-test (valor de p)	Diferença Significativa
	JLoad	Eclipse		
<i>SysUse</i>	2,52	3,31	0,101548	Não
<i>InfoQual</i>	1,94	3,11	0,030102	Sim
<i>IntQual</i>	1,71	3,09	0,015533	Sim
<i>Média Geral</i>	2,14	3,16	0,032804	Sim

#### 6.2.1.1 Utilidade do sistema (SysUse)

Pelas médias para o JLoad = 2,52 e Eclipse= 3,31, observa-se que o JLoad foi melhor avaliado pelos participantes. As duas ferramentas podem ser consideradas úteis para os participantes, porém, com  $t(14)=1,752507$ ,  $p=0,101548$ , os resultados apontam que não existe uma diferença significativa em relação à utilidade das ferramentas.

Os resultados são apresentados utilizando o formato APA (*American Psychological Association*). No resultado reportado no parágrafo anterior, por exemplo, o  $t$  inicial denota o teste estatístico utilizado (teste  $t$  pareado); o valor em parênteses, logo em seguida, denota o número de graus de liberdade; a seguir é apresentado o *valor de t* obtido no teste; e, por fim, é mostrado o nível de significância, ou *valor de p*. Para existir diferença significativa, o nível de significância deve ser menor que 0,05.

#### 6.2.1.2 Qualidade da informação (InfoQual)

Com as médias para JLoad=1,94 e Eclipse=3,11, observa-se que um dos pontos fortes do JLoad está na qualidade da informação. Este ponto está relacionado à comunicação do usuário com sistema, sendo essencial para o aprendizado dos usuários surdos. Os resultados apontam diferença significativa já que  $t(14)=2,413135$ ,  $p=0,030102$ , sendo possível concluir que os participantes avaliaram superioridade na qualidade da informação do JLoad em relação ao Eclipse.

#### 6.2.1.3 Qualidade da interface (IntQual)

Com as médias para JLoad=1,71 e Eclipse=3,16, observa-se que a interface do JLoad foi melhor avaliada pelos participantes, mostrando também diferença significativa em relação ao Eclipse,  $t(14)=2,753643$ ,  $p=0,015533$ . Os resultados apontam que, para usuários surdos, a interface do JLoad mostrou-se mais eficiente em relação ao Eclipse.

#### 6.2.1.4 Média Geral

Os resultados apontam que os participantes avaliaram melhor o JLoad em diversos aspectos, dentre eles alguns importantes para interação com sistema, principalmente levando em consideração o perfil de usuário surdo. Para eles o cuidado com a comunicação durante a construção de um sistema é fundamental para que possam executar suas tarefas.

A média geral da avaliação de usabilidade com médias de JLoad=2,14 e Eclipse=3,16, apontam a preferência dos participantes para o uso do JLoad. Aplicado o t-student, com  $t(14)=2,368213$ ,  $p=0,032804$ , verificou-se que também há uma diferença estatística significativa, com isto, a proposta do JLoad para o ensino e aprendizado de programação para alunos surdos torna-se viável e uma forte prerrogativa para que surdos tenham melhor desempenho no aprendizado de uma linguagem de programação.

#### 6.2.2 Análise Qualitativa

Os resultados obtidos na análise quantitativa dos dados de usabilidade foram confirmados ao analisar as opiniões dos participantes.

A seguir, são apresentados os depoimentos de dois participantes. De acordo com eles, o JLoad proporciona uma interface fácil de utilizar que possibilita um aprendizado rápido. É possível verificar no depoimento características que sinalizam um melhor acesso a informação, a apresentando com uma melhor qualidade para o surdo. Além disso, um ponto negativo da ferramenta Eclipse é a alternância de telas para realização da oficina, visto que o mesmo não disponibiliza recursos de acessibilidade para os surdos.

*“[...] a ferramenta JLoad é perfeita, ela possibilita o aprendizado rápido, por conta da presença do tutor, da opção de gravar vídeos. O Eclipse tem suas qualidades, porém o JLoad é melhor, porque para o surdo é preciso acessibilidade e nessa ferramenta é possível encontrar. No Eclipse não existe a opção de vídeo chamada com o tutor, já no JLoad tem, por isso é mais atrativo para surdos. No meu ponto de vista o JLoad é melhor para uso do que o Eclipse.”*

Participante 3, 19 anos, sexo masculino, nenhuma experiência em programação.

*“[...] hoje nós utilizamos o eclipse, e fui me lembrando o que tinha estudado na aula passada e fui fazer a oficina, mas quando comecei acabei esquecendo algumas coisas. Na verdade é*

*bem complicado utilizar o eclipse, pois caso eu tenha dúvidas é preciso estar entrando em outra tela para poder me comunicar com o tutor; assim fica mais difícil realizar a atividade rápido. O JLoad é simples, pois nele há acessibilidade, as opções são bem localizadas, as informações são fáceis. No eclipse é mais trabalhoso, pois é preciso abrir mais de uma tela para usá-lo.”*

Participante 5, 19 anos, sexo masculino, nenhuma experiência em programação.

As avaliações sugerem que o JLoad apresentou uma melhor usabilidade para o público surdo em relação a ferramenta Eclipse, facilitando a realização das atividades de codificação e permitindo um melhor aprendizado.

### 6.3 MOTIVAÇÃO

Cada participante, ao final da oficina, respondeu o questionário SIMS (FORTIER; VALLERAND; GUAY, 1995) para avaliar o nível de motivação na realização daquela atividade. Todas as perguntas foram traduzidas para Libras para melhorar a compreensão dos participantes, que eram todos deficientes auditivos. Ao participante era feito o seguinte questionamento:

*Por que você está atualmente envolvido nesta atividade?*

Em seguida, catorze respostas para esse questionamento eram oferecidas ao participante como pode ser visto na Seção 2.9.1. Para cada uma delas, o participante expressava o seu nível de concordância utilizando a escala a seguir: *1: não concordo; 2: concordo muito pouco; 3: concordo pouco; 4: concordo moderadamente; 5: concordo o suficiente; 6: concordo muito; e 7: concordo totalmente.*

#### 6.3.1 Análise Quantitativa

A Tabela 6 mostra a média das pontuações de motivação obtidas pelos participantes no JLoad e no Eclipse.

Os dados foram submetidos ao teste de normalidade Shapiro-Wilk (SHAPIRO; WILK, 1965). Como os dados referentes às pontuações de motivação geral dos participantes não são satisfazem uma distribuição normal, eles foram submetidos ao teste não-paramétrico Wilcoxon Signed-Rank. Todos as demais pontuações foram submetidas ao teste paramétrico t-test pareado. Pela Tabela 6, podemos verificar que o resultado favorece o JLoad em todas as

Tabela 6 – Média das Métricas de Motivação

Métricas de Motivação	Pontuação Média		Validação	Teste Estatístico	Diferença Significativa
	JLoad	Eclipse			
<i>Motivação Geral</i>	5,58	5,07	Maior melhor	W=38	Não
<i>Amotivação</i>	3,68	3,73	Menor melhor	p=0,915	Não
<i>Regulação Externa</i>	5,89	5,42	Maior melhor	p=0,380	Não
<i>Regulação Identificada</i>	6,11	5,56	Maior melhor	p=0,283	Não
<i>Motivação Intrínseca</i>	6,22	5,25	Maior melhor	p=0,090	Não

métricas analisadas, contudo não foi identificada diferença significativa pelos testes estatísticos. A seguir detalhamos um pouco melhor os resultados.

#### 6.3.1.1 Motivação Geral

Com as médias para JLoad=5,58 e Eclipse=5,07, o desempenho na motivação geral é bastante similar com desempenho levemente melhor para o JLoad. Visto o tamanho da amostra de 14 (descontando os empates), consideramos o valor de W do teste *Wilcoxon Signed-Rank*. Como o resultado obtido foi **W = 38, p > ,05, (valor crítico de W = 21)** o resultado **não** apresenta diferença significativa mesmo o JLoad apresentando uma melhor média de motivação geral.

O teste *Wilcoxon Signed-Rank* nos fornece dois valores: o valor Z e o valor W. O primeiro é utilizado quando o tamanho da amostra é superior a 20. O segundo valor é usado para amostras menores. Esse valor é comparado com um valor crítico que varia de acordo com o nível de significância. No resultado reportado no parágrafo anterior, o valor crítico de W para o nível de significância de 0,05 é 21. Para existir diferença significativa, o valor de W deve ser menor que o valor crítico.

#### 6.3.1.2 Amotivação

A métrica de amotivação (desmotivação) deve ser vista ao contrário. Aqui, quanto menor, melhor. A amotivação média das ferramentas foi bem parecida (JLoad=3,68 e Eclipse=3,73), com leve vantagem para o JLoad.

Contudo, o resultado do teste estatístico paramétrico t-test, **t(14)=0,108587, p=0,915071**, mostra que **não** há diferença significativa.

### 6.3.1.3 Regulação Externa

Com médias JLoad=5,89 e Eclipse=5,42, a média da métrica também favorece o JLoad, mas o valor obtido  $t(14)=-0,906105$ ,  $p=0,380211$  indica **ausência** de diferença significativa.

### 6.3.1.4 Regulação Identificada

O JLoad apresenta melhor resultado ao analisar a média das métrica de Regulação Identificada (JLoad=6,11 e Eclipse=5,56). Já os valores  $t(14)=-1,115865$ ,  $p=0,283267$  obtidos no teste estatístico permitem concluir que **não** existe diferença significativa.

### 6.3.1.5 Motivação intrínseca

A média das pontuações de motivação intrínseca também apresenta um melhor resultado para o JLoad (JLoad=6,22 e Eclipse=5,25). Dentro de todas as métricas, essa é a que apresenta um melhor resultado estatístico. Contudo, com  $t(14)=-1,820134$ ,  $p=0,09018$ , **não** existe diferença significativa.

## 6.3.2 Análise Qualitativa

A seguir, apresenta-se o depoimento do Participante 6, 19 anos, sexo masculino, nenhuma experiência em programação. De acordo com o mesmo, o JLoad proporciona motivação para o aprendizado na área de programação. É possível verificar no depoimento características que demonstram que a ferramenta é adequada para o surdo:

*“[...] Foi minha primeira experiência com programação, então não tinha discernimento de como utilizar, como fazer os códigos, a linguagem também, e com esse experimento pude ter a noção e fico muito feliz por ter aprendido. Ter contato com essa ferramenta me deixou animado, pois vejo que pessoas lutam pela igualdade dos surdos com a comunidade, uma tecnologia inovadora e acessível. Quando utilizei o Jload, achei muito importante principalmente para o público surdo, uma ferramenta clara, explicativa e eu tive como saber bem rápido se o código estava certo ou errado, então fui praticando até conseguir fazer toda a atividade. Como foi minha primeira experiência com programação, achei um pouco difícil mas fui praticando e vendo os vídeos até conseguir realizar toda a atividade.*”

*Gostei muito da ferramenta e tenho vontade de aprender mais.”*

Participante 6, 19 anos, sexo masculino, nenhuma experiência em programação.

#### 6.4 QUESTIONÁRIO COMPARATIVO DAS FERRAMENTAS JLOAD E ECLIPSE

Cada participante, ao término do experimento, respondeu a um questionário na escala likert. O sujeito respondia 1 caso *discordasse fortemente* com a assertiva e respondia 5 caso *concordasse fortemente*.

##### 6.4.1 Análise Quantitativa

A Tabela 7 apresenta as médias das respostas dos participantes para cada ferramenta juntamente com os dados da análise quantitativa.

Tabela 7 – Média das Respostas do Questionário Comparativo

Pergunta	Pontuação Média		W-Value	Valor Crítico	Diferença Significativa
	JLoad	Eclipse			
<i>A ferramenta é fácil de usar</i>	4,533	2,8	2,5	12	Forte
<i>Me comuniquei facilmente com o tutor</i>	4,733	2,4	0	12	Forte
<i>Terminei a oficina mais rápido na ferramenta</i>	4,466	2,666	0	5	Forte
<i>A ferramenta me deixou cansado ou entediado</i>	1,733	2,2	7	2	Não
<i>Eu estaria motivado a fazer outra oficina na ferramenta</i>	4,866	3,6	0	1	Forte
<i>A ferramenta tem recursos para os surdos</i>	5	1,266	0	12	Forte

As respostas das assertivas foram submetidas ao teste de normalidade Shapiro-Wilk (SHAPIRO; WILK, 1965) e, em seguida, ao teste não paramétrico Wilcoxon Signed-Rank visto que não foi possível supor que os dados seguissem uma distribuição normal.

##### 6.4.1.1 A ferramenta é fácil de usar

Quanto a facilidade de usar as ferramentas, o JLoad apresentou média 4,533 superior ao Eclipse, 2,8. O resultado obtido pelo teste estatístico foi **W=2.5, p < 0,01 (valor crítico de W = 12)**. Indicando uma forte diferença significativa.

#### 6.4.1.2 Me comuniquei facilmente com o tutor

Em relação a facilidade de comunicação com o tutor, novamente o JLoad apresentou melhor resultado na média das respostas (JLoad=4,733 e Eclipse=2,4). O resultado do teste estatístico foi **W = 0, p < 0,01 (valor crítico de W = 12)**. Portanto, o resultado apresenta forte diferença significativa.

#### 6.4.1.3 Terminei a oficina mais rapidamente na ferramenta

Quanto às respostas sobre a velocidade de término da oficina, as médias obtidas foram JLoad=4,466 e Eclipse=2,666, ou seja, mais um resultado favorecendo o JLoad. Nos testes estatísticos o resultado foi **W = 0, p < 0,01 (valor crítico de W = 5)**, indicando, portanto, uma forte diferença significativa.

#### 6.4.1.4 A ferramenta me deixou cansado ou entediado

Devido ao número de empates nas respostas sobre se a ferramenta deixou os usuários cansados ou entediados, a amostra foi reduzida para N = 7. O resultado **W = 7, p > 0,05 (valor crítico de W = 2)** indica que a diferença estatística não é significativa. Contudo, as médias das respostas obtidas mostra um melhor resultado para o JLoad com média 1,733 contra 2,2 do Eclipse. Nessa pergunta em particular, quanto menor a pontuação, melhor o desempenho.

#### 6.4.1.5 Estaria motivado a fazer outra oficina na ferramenta

Sobre as respostas da assertiva de motivação para realizar outra oficina usando a ferramenta, o resultado foi **W = 0, p < 0,01 (valor crítico de W = 1)**. Portanto, o resultado apresenta forte diferença significativa. O que ratifica os valores médios obtidos (JLoad=4,866 e Eclipse=3,6) e vantagem do JLoad.

#### 6.4.1.6 A ferramenta tem recursos para surdos

Sobre a existência de recursos para surdos na ferramenta, o JLoad apresenta o valor 5 como média das respostas, ou seja, todos os participantes concordam fortemente à respeito da acessibilidade oferecida na ferramenta. O Eclipse apresentou média de 1,266. O valor obtido no teste estatístico foi **W = 0, p < 0,01 (valor crítico de W = 12)**. Portanto, o resultado também apresenta forte diferença significativa.

Em resumo, o JLoad obteve um melhor desempenho na média de todos os pontos do questionário, além disso, os testes estatísticos ratificam esse resultado. Confirmando as seguintes características do JLoad: melhor usabilidade; melhor interação tutor-aluno; término mais rápido das oficinas; menor cansaço para os surdos; uma motivação maior para realização de outra oficina; e a existência de recursos de acessibilidade.

#### 6.4.2 Análise Qualitativa

A seguir, são apresentados depoimentos onde os participantes comparam o uso das duas ferramentas. Eles ressaltam principalmente a presença de recursos de acessibilidade; maior facilidade de uso; e um melhor contato com o tutor.

*“ [...] Boa Tarde, novamente estou aqui, mas hoje para aprender como utilizar a ferramenta eclipse. É bem similar ao JLoad, porém para usar o Eclipse necessita instalar no computador, para se procurar algo dentro da ferramenta como pastas, ou para pesquisar algum texto, precisa conhecer um pouco de inglês. No Eclipse tem uma ferramenta que nos ajuda a compilar, o JLoad tem a mesma funcionalidade, mas para quem é surdo há uma parte a qual explica à atividade, há também o vídeo em Libras e um chat para se comunicar com o tutor, caso tenha dúvidas com o código ou como executar. Para suprir a necessidade de uma pessoa com deficiência o software mais fácil é o JLoad, pois caso a pessoa queira fazer o código em casa e tenha dúvidas, ela não precisará entrar no site da DELL para tentar resolver o problema. A principal razão das pessoas que desenvolve é tentar resolver o problema da maneira mais fácil.”*

Participante 1, 23 anos, sexo masculino, média experiência em programação.

*“[...] Na minha opinião o Eclipse tem um grau de dificuldade maior e não é acessível para o surdo. O JLoad é mais acessível para a utilização da pessoa surda e com ele tive mais facilidade para a criação dos códigos.”*

Participante 2, 29 anos, sexo feminino, pouca experiência em programação.

*“[...] a ferramenta JLoad é perfeita, ela possibilita o aprendizado rápido, por conta da presença do tutor, da opção de gravar vídeos. O Eclipse tem suas qualidades, porém o JLoad é melhor, porque para o surdo é preciso acessibilidade e nessa ferramenta é*

*possível encontrar. No Eclipse não existe a opção de vídeo chamada com o tutor, já no JLoad tem por isso é mais atrativo para surdos. No meu ponto de vista o JLoad é melhor para uso do que o Eclipse.”*

Participante 3, 19 anos, sexo masculino, nenhuma experiência em programação.

*“[...] meu sinal na comunidade surda é esse. O Eclipse é uma ferramenta complicada, nela não há a presença de um tutor, é preciso sair da página que está e ir ao correio interno, caso você tenha alguma dúvida, para poder tirar com o tutor, através de um e-mail, depois voltar para a plataforma. Então precisa está fazendo esse fluxo durante todo o teste. No Eclipse não tem acessibilidade para o surdo, já no JLOAD é possível realizar o teste com mais simplicidade, pois tem vídeos em Libras para auxiliar..Obrigado.”*

Participante 4, 18 anos, sexo masculino, nenhuma experiência em programação.

## 6.5 TEMPOS, AUXÍLIOS E NOTAS

Dentro do experimento foram observadas algumas outras métricas para comparar o desempenho dos participantes na utilização das duas ferramentas, dentre elas: tempo que o sujeito de pesquisa necessitava para finalizar as oficinas; o número de solicitações de auxílio ao tutor a respeito de dúvidas sobre o conteúdo da oficina; o número de requisições de ajuda quanto a dúvidas sobre o uso das ferramentas; a quantidade de solicitações ao Intérprete de Libras; e, por fim, a nota do participante na oficina. Câmeras de vídeo capturaram as interações dos participantes. Para coleta dos tempos e auxílios, esses vídeos foram analisados utilizando-se da técnica de análise situada para determinar os tempo de início e término da oficina de cada participante, bem como, o número de auxílios solicitados.

### 6.5.1 Análise Quantitativa

A Tabela 8 resume os dados referentes às métricas em questão. São apresentadas as médias dos resultados de cada participante, bem como os valores obtidos nos testes estatísticos. Todos os dados dos participantes foram submetidos ao teste de normalidade Shapiro-Wilk (SHAPIRO; WILK, 1965). Com isso, a normalidade dos dados pode ser suposta nas métricas de tempo, auxílio a ferramenta, auxílio ao intérprete e notas. Para essas métricas foi utilizado o teste estatístico paramétrico t-test pareado. Como a normalidade dos dados foi rejeitada para a

Tabela 8 – Quadro Resumo da Análise Quantitativa das Métricas

Métrica	Média		Teste Estatístico	Diferença Significativa
	JLoad	Eclipse		
<i>Tempo (min)</i>	44min 40s	64min 28s	p=0,0336	Sim
<i>Auxílio - Conteúdo</i>	6,333	3,666	W=27	Não
<i>Auxílio - Ferramenta</i>	1	5,133	p=0,0005	Forte
<i>Auxílio - Intérprete</i>	2,2	3,8	p=0,1901	Não
<i>Notas</i>	9,2	8,153	p=0.0281	Sim

métrica de auxílio a conteúdo, utilizamos o teste não paramétrico Wilcoxon Signed-Rank para esses dados.

#### 6.5.1.1 Tempo

A partir da Tabela 8, podemos observar que, em média, o sujeito de pesquisa que realizou a atividade no JLoad conseguiu concluir a oficina em menor tempo (44min 40s) do que aquele que utilizou o Eclipse (64min 28s). O resultado do teste estatístico foi  $t(14)=2,355649$ ,  $p=0,0336$ . O que indica diferença significativa na métrica de tempo entre as ferramentas. Com isso podemos concluir que o JLoad permite que os alunos realizem a oficina em menor tempo, e consequentemente, fiquem mais motivados para realizar a próxima atividade.

#### 6.5.1.2 Auxílios a conteúdo

Quando um aluno realiza uma oficina, ele provavelmente pode ter dúvida sobre o conteúdo. Perguntas do tipo como a atividade pode ser feita; se o que foi realizado está indo pelo caminho certo; ou porque seu código deu errado. Nessas situações, surge a necessidade da interação e proximidade com o tutor que auxilia o aluno nas suas dúvidas fazendo com que ele sinta interesse em ir até o final. Nesse quesito, em média, o sujeito de pesquisa que utilizou o JLoad solicitou mais ajuda ao tutor. O JLoad apresentou uma média de 6,333 auxílios contra 3,666 do Eclipse. Podemos concluir que a ferramenta possibilita uma melhor presença social do tutor. Essa maior interação faz com que o aluno se sinta mais motivado para fazer questionamentos.

Contudo, o resultado  $W = 27$ ,  $p > 0,05$  (valor crítico de  $W = 17$ ) não permite afirmar que a diferença entre as ferramentas apresenta diferença significativa.

### 6.5.1.3 Auxílios a Ferramenta

Os participantes do experimento quando tinham dúvida sobre o uso das ferramentas podiam pedir auxílio sobre como utilizar certa funcionalidade por exemplo. O JLoad apresentou, em média, uma dúvida por participante contra 5,133 do Eclipse. Esse resultado é confirmado pelo teste estatístico  $t(14)=4,467821$ ,  $p=0,000531$  que nos mostra uma forte diferença significativa visto que  $p < 0,01$ . Podemos concluir que a utilização do o JLoad é mais intuitiva e o aluno tem um melhor resultado já que não tem grandes dúvidas sobre o uso da ferramenta.

### 6.5.1.4 Auxílios Intérprete

Todos os participantes do experimento eram deficientes auditivos, logo, a maioria deles possuem a Libras como primeira língua. Ao realizar suas atividades os mesmos poderiam solicitar auxílio do intérprete de Libras caso tivessem dúvidas sobre a oficina, ferramenta, conteúdo, ou o próprio experimento em si. Uma equipe de intérpretes estava a disposição para responder tais dúvidas. Com o JLoad, os participantes precisaram de menos colaboração do intérprete o que nos posiciona como uma ferramenta onde o aluno surdo tem uma maior autonomia no desenvolvimento do seu aprendizado. A média de solicitações no JLoad foi 2,2 e a do Eclipse 3,8. O resultado estatístico  $t(14)=1,376818$ ,  $p=0,190188$ , contudo, não apresenta diferença significativa.

### 6.5.1.5 Notas

Para fim de mensuração do processo de aprendizado ao realizar a oficina prática, os códigos criados foram todos corrigidos por um único avaliador.

A correção realizada foi às cegas (blinded), ou seja, o avaliador não tinha conhecimento do participante autor dos códigos e, além disso, ele também não tinha informações se os códigos que estava corrigindo tinham sido criados utilizando a ferramenta Eclipse ou a ferramenta JLoad.

Na média, o JLoad possibilitou uma melhor nota aos alunos (média de 9,2 contra 8,153 no Eclipse), o resultado dos testes estatísticos  $t(14)=-2,447275$ ,  $p=0,028191$  só ratifica esse resultado indicando diferença significativa.

### 6.5.2 Análise Qualitativa

São mostrados, a seguir, mais alguns depoimentos dos participantes. Em todos eles, os sujeitos reforçam sua preferência pelo JLoad para aprendizado do aluno surdo.

É possível identificar nos depoimentos uma maior presença do tutor quando ao utilizar o JLoad, onde os participantes mencionam a facilidade de pedir auxílio. É preciso ressaltar que, ao utilizar o Eclipse, os sujeitos também podiam solicitar auxílios ao tutor, mas esses eram realizados através do correio interno do AVA, ou seja, um fluxo mais complicado com troca de diferentes telas.

Podemos concluir que o JLoad possibilita uma maior presença social do tutor, dessa maneira, o aluno não se sente desamparado no seu processo de aprendizagem.

*“[...] É a primeira vez que participo aqui do curso e também do experimento, percebi que a ferramenta JLoad possui acessibilidade pois tem o vídeo em Libras e o texto em Português. Ele proporciona um ensino de como realizar as atividades, eu achei bem interessante e acho que pode melhorar; aliás eu acredito que no futuro vai estar bem melhor e será uma ótima ferramenta. A atividade que eu realizei, compreendi o necessário para realizar. Mas se me interessar para aprender realmente, será bem mais fácil.”*

Participante 5, 19 anos, sexo masculino, nenhuma experiência em programação.

*“[...] Hoje é meu segundo dia de experimento, comecei com o JLoad que é uma ferramenta boa, e hoje fiz utilizando o eclipse. Particularmente, eu gostei de utilizar, tem suas dificuldades, mas fui testando. O JLoad é muito mais simples, nele há opção que nos ajudam a utilizar, visualmente, e também tem a presença do tutor. No eclipse é mais complicado compreender como realizar a atividade. Fazendo uma comparação entre essas duas ferramenta o JLoad se torna melhor do que o Eclipse, porque no JLoad nós temos auxílio e se torna fácil, pois para o aluno surdo é preciso ter um sistema mais visual para que ele possa aprender mais rápido. Por isso o JLoad é o sistema ideal, pois nele há acessibilidade para os alunos surdos.”*

Participante 6, 19 anos, sexo masculino, nenhuma experiência em programação.

*“[...] Eu havia utilizado o eclipse e o achei complicado. Tentei enviar mensagem para o tutor, mas demorava muito a responder, tive que parar durante duas vezes para revisar o conteúdo para tentar finalizar a atividade. O eclipse realmente é bem difícil mas no JLoad me senti mais segura a utilizar, foi bem fácil aprender e realmente é bem mais acessível para os alunos surdos.”*

Participante 7, 21 anos, sexo feminino, nenhuma experiência em programação.

*“[...] o experimento que realizei semana passada teve um grau de dificuldade maior. Precisei pesquisar, treinar, para conseguir fazer o que se pedia. Nesse agora, o JLoad , foi um pouco mais fácil, pois, teve o auxílio do tutor e intérprete, caso tivesse alguma dúvida no código se este não funcionasse, ou outras dúvidas, teria a disposição esse auxílio. Se torna mais simples, pois sou surdo. O JLoad é a ferramenta mais fácil, pois no Eclipse há textos em português, a gramática e o surdo não consegue compreender, dentre outras dúvidas. O JLoad é mais fácil pois temos o auxílio do intérprete. Obrigado.”*

Participante 8, 16 anos, sexo masculino, nenhuma experiência com programação.

## 6.6 CONCLUSÃO DOS RESULTADOS

Através desse estudo, foi possível desenvolver o OA JLoad, cujo desafio e objetivo sempre foi ser uma ferramenta para o aprendizado de surdos na área de programação Java. Em sua concepção, um design participativo foi realizado com o público alvo. A ferramenta foi desenvolvida baseada nos critérios internacionais de acessibilidade. Ela foi validada com um grupo de 15 surdos, e, ao final, foi bem aceita por parte dos participantes do experimento, alcançando os objetivos definidos.

O JLoad apresentou melhores resultados quanto à Usabilidade e Motivação. Nas métricas de Usabilidade, quanto menor o valor, melhor. A usabilidade geral média apresentada pelo JLoad foi 2,14 contra 3,16 do Eclipse. O JLoad também apresenta um melhor desempenho na motivação geral média com 5,58 contra 5,07 do Eclipse.

Na realidade, em todas as medições realizadas, o JLoad obteve, na média, um resultado melhor que o obtido pelo Eclipse: (1) Métricas de Usabilidade: SysUse, InfoQual, IntQual,

Usabilidade Geral; (2) Métricas de Motivação: Motivação Geral, Amotivação, Regulação Externa, Regulação Identificada, Motivação Intrínseca; (3) tempos de término; (4) auxílios; (5) notas; e (6) todos os itens do questionário comparativo.

Em grande parte dessas medições, esse resultado foi ratificado por uma diferença estatística significativa obtida através da submissão das pontuações a testes estatísticos. São elas: InfoQual; IntQual; Usabilidade Geral; cinco dos seis itens do questionário comparativo; tempos; auxílios a ferramenta; e notas.

O JLoad fornece uma melhor usabilidade, permite uma maior motivação aos participantes, seus usuários conseguem finalizar todas as tarefas, em média, cerca de 20 minutos mais rápido e obtêm melhores notas (um ponto acima das notas obtidas ao utilizar o Eclipse, em média). Além disso, a presença social do tutor é maior, permitindo um maior engajamento dos alunos para retirada de dúvidas: em média, o aluno utilizando o JLoad solicitou cerca de 70% mais auxílios de conteúdo ao tutor que aquele que utilizava o Eclipse. Tudo isso foi confirmado através dos depoimentos realizados pelos participantes que atestam, dentre outras qualidades, que o uso do JLoad permite uma maior motivação no aprendizado da linguagem de programação Java.

No próximo capítulo são apresentadas as considerações finais referentes a pesquisa do JLoad.

## 7 CONCLUSÕES E TRABALHOS FUTUROS

Os desafios na educação de PCDs são grandes. As abordagens tradicionais geralmente não se aplicam e requisitos específicos devem ser considerados. Isso ocorre visto que diferentes estímulos devem ser considerados dependendo da deficiência. Por exemplo, surdos não podem fazer uso da audição, contudo fazem bastante uso da visão, além disso, possuem uma língua própria. Portanto, abordagens de ensino que possuam mais elementos visuais e fazem uso de recursos que permitem ao surdo utilizar sua própria língua serão mais eficientes no aprendizado.

Com isso em mente e no contexto do ensino de programação em Java, foi apresentado o JLoad. Uma ferramenta concebida para diminuir a evasão nas aulas iniciais e possibilitar um melhor aprendizado, aumentando a motivação dos alunos através de recursos acessíveis, aprendizado progressivo e interação com o tutor. Na fase de concepção foi realizado design participativo onde os surdos foram integrados em diversas fases do processo de desenvolvimento, partindo desde a coleta inicial dos requisitos, passando pelo design das telas através de protótipos de baixa fidelidade e chegando até os testes de validação da ferramenta. O JLoad foi baseado nas seguintes teorias: aprendizado ativo, zona de desenvolvimento proximal de Vygotsky (1989), teoria da aquisição de linguagem de Krashen (1982), Teoria do fluxo de Csikszentmihalyi (1991), motivação para aprendizagem e presença social (SHORT; WILLIAMS; CHRISTIE, 1976).

Baseado na teoria da aquisição da linguagem, de Krashen (1982), a ferramenta elaborada tem o objetivo de: (1) minimizar a curva de aprendizagem; (2) oferecer a possibilidade de um acompanhamento contínuo; (3) aumentar a auto-confiança e motivação para aprendizagem do estudante; e (4) diminuir o nível de ansiedade. Conceitos de aprendizagem situada influenciaram o design da interação tutor/aluno, onde um espaço de trabalho compartilhado também é utilizado. Com base na teoria de fluxo de Csikszentmihalyi (1991), o JLoad é proposto juntamente com um programa de curso apropriado para ajustar as demandas das oficinas de acordo com as competências dos alunos para maximizar a experiência do usuário, mantendo-os na zona de fluxo. O JLoad também provê uma maior presença social do tutor, facilitando a comunicação online e a interação entre tutor e aluno através do chat que disponibiliza recursos de acessibilidade como o uso de Libras para envio de mensagens. Demandas de outros atores (intérpretes e tutores) também são atendidas de maneira a facilitar o acompanhamento dos alunos.

Além disso, o JLoad possui uma IDE própria permitindo a codificação, compilação e execução dos códigos no próprio Objeto de Aprendizagem e ele também é integrado a um AVA

acessível, onde os conteúdos dos cursos são disponibilizados para os alunos, isso é fundamental pois evita que os surdos fiquem mudando de contexto, ou seja, mudando de tela, saindo do AVA e indo para IDE. Essa mudança de contexto é muito negativa para o esforço cognitivo. Tudo desenvolvido para otimizar o aprendizado do aluno surdo no desenvolvimento dos seus primeiros programas.

Dentre as limitações do JLoad podemos citar que ele não trabalha com elementos gráficos; não é acessível para outras deficiências como cegos e tetraplégicos; e mesmo sendo responsivo não apresenta uma versão própria para *mobile*;

A ferramenta foi validada e avaliada por um grupo de surdos, ao mesmo tempo que foi comparada com o Eclipse em diversos pontos: métricas de usabilidade e motivação, tempos para término da oficina, número de auxílios solicitados, notas dos alunos e, por fim, um questionário comparativo respondido pelos participantes. Em todas as métricas, o JLoad apresenta, em média, um melhor resultado e em grande parte delas esse resultado é ratificado por diferença significativa aferida a partir de testes estatísticos.

Todas as fases de testes e experimento foram realizadas e concluídas com sucesso, o JLoad sai da fase experimental e integra mais um dos OAs do AVA Dell Accessible Learning, sendo utilizado por alunos do curso de Java básico, envolvendo surdos e pessoas sem deficiência, a partir do final de 2016.

Como trabalhos futuros, a ferramenta pode ser evoluida para atender outras deficiências, bem como pode ser criada uma versão para dispositivos móveis. A mesma pode futuramente dar suporte para o aprendizado de outras linguagens de programação. O JLoad também pode evoluir para se tornar um espaço de trabalho colaborativo para diferentes públicos como ouvintes e surdos.

Por fim, a presente pesquisa permitiu a publicação de dois artigos em conferências:

- *Introducing the Jload: A Java Learning Object to Assist the Deaf* (SILVA *et al.*, 2014); e
- *IT Education Strategies for the Deaf* (OLIVEIRA *et al.*, 2016).

## REFERÊNCIAS

- ALMEIDA, M. E. B. d. Tecnologia e educação a distância: abordagens e contribuições dos ambientes digitais e interativos de aprendizagem. **Revista Brasileira de Educação a Distância**, p. 6, 2011.
- ALVES, F. d. S. Ensino de física para pessoas surdas: o processo educacional do surdo no ensino médio e suas relações no ambiente escolar. Universidade Estadual Paulista (UNESP), 2012.
- AMABILE, T. M.; HILL, K. G.; HENNESSEY, B. A.; TIGHE, E. M. The work preference inventory: assessing intrinsic and extrinsic motivational orientations. **Journal of personality and social psychology**, American Psychological Association, v. 66, n. 5, p. 950, 1994.
- AMES, C.; ARCHER, J. Achievement goals in the classroom: Students' learning strategies and motivation processes. **Journal of educational psychology**, American Psychological Association, v. 80, n. 3, p. 260, 1988.
- BARROS, E. M. **Ações Coletivas, Identidade e Mobilizações políticas: Movimento Social Surdo e a luta por reconhecimento**. Dissertação (Mestrado) — Instituto de Filosofia e Ciências Humanas. Programa de Pós Graduação em Sociologia da Universidade Estadual de Campinas., 2015.
- BEHARES, L. E. Novas correntes na educação do surdo: dos enfoques clínicos aos culturais. **Santa Maria: UFSM**, p. 1–22, 2000.
- BERNHARD, J. Activity based physics education-some examples of innovative approaches at some universities and colleges in usa. **CUP-day. Linkoping University**, 1999.
- BORUCHOVITCH, E.; COSTA, E. d.; BORUCHOVITCH, E.; BZUNECK, J. O impacto da ansiedade no rendimento escolar e na motivação de alunos. **A motivação do aluno. Contribuições da psicologia contemporânea**, Vozes Petrópolis, p. 134–147, 2001.
- BRANDAO, L. O.; BRANDAO, A. A. F.; RIBEIRO, R. S. iVProg – uma ferramenta de programação visual para o ensino de algoritmos. In: **Proc. Congresso Brasileiro de Informatica na Educacao (CBIE) - Workshop on Software for Computers in Education**. Rio de Janeiro, Brazil: [s.n.], 2012. Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/1668/1430>>.
- CASPERSEN, M. E.; KOLLING, M. Stream: A first programming process. **ACM Transactions on Computing Education (TOCE)**, ACM, v. 9, n. 1, p. 4, 2009.
- CASTORINA, J. A.; FERREIRO, E.; LERNER, D.; OLIVEIRA, M. K. de; SCHILLING, C. **Piaget-Vygotsky: novas contribuições para o debate**. São Paulo, SP: Editora Ática, 2006.
- CENSO, E. B. **BR.(2010). Organização Associação Brasileira de Educação a Distância**. São Paulo, SP: Pearson Education do Brasil, 2014.
- CHAVES, E. **Legendagem para Surdos e ensurdecidos: um estudo baseado em corpus da segmentação nas legendas de filmes brasileiros em DVD**. Dissertação (Mestrado) — Programa de Pós-Graduação em Linguística Aplicada da Universidade Estadual do Ceará., 2012.
- CHICKERING, A. W.; GAMSON, Z. F. Seven principles for good practice in undergraduate education. **AAHE bulletin**, ERIC, v. 3, p. 7, 1987.

CONSORTIUM, W. W. W. *et al.* Web content accessibility guidelines (wcag) 2.0. World Wide Web Consortium, 2008.

CSIKSZENTMIHALYI, M. **Flow: The psychology of optimal experience**. New York: Harper Perennial, 1991.

CSIKSZENTMIHALYI, M.; NAKAMURA, J. The dynamics of intrinsic motivation: A study of adolescents. In: **Flow and the Foundations of Positive Psychology**. New York, EUA: Springer, 2014. p. 175–197.

DIAS, C. **Usabilidade na WEB: criando portais mais acessíveis**. Rio de Janeiro: Alta Books, 2007. ISBN 9788576081401. Disponível em: <<https://books.google.com.br/books?id=CohauAAACAAJ>>.

DOU - BRASIL. **Regulamenta a profissão de Tradutor e Intérprete da Língua Brasileira de Sinais**. 2010. Lei Brasileiro No. 8.213 - 01 de Setembro de 2010. Diário Oficial da Republica Federativa do Brasil - Brasília, DF - Brasil.

DRIGAS, A. S.; KOUREMENOS, D.; KOUREMENOS, S.; VRETTAROS, J. An e-learning system for the deaf people. In: **Proc. 6th International Conference on Information Technology Based Higher Education and Training (ITHET 2005)**. Santo Domingo, Dominican Republic: [s.n.], 2005. p. T2C/17–T2C/21.

DUQUE, R.; BRAVO, C. Analyzing work productivity and program quality in collaborative programming. In: IEEE. **2008 The Third International Conference on Software Engineering Advances**. Sliema, Malta, 2008. p. 270–276.

ECLIPSE, I. F. **Eclipse - The Eclipse Foundation open source community website**. 2016. Disponível em: <<https://eclipse.org/>>.

FABRE, M.-C. J.; TAMUSIUNAS, F.; TAROUÇO, L. M. R. Reusabilidade de objetos educacionais. **RENTE**, v. 1, n. 1, 2003.

FORTIER, M. S.; VALLERAND, R. J.; GUAY, F. Academic motivation and school performance: Toward a structural model. **Contemporary educational psychology**, Elsevier, v. 20, n. 3, p. 257–274, 1995.

GAMBOA, V.; VALADAS, S. T.; PAIXÃO, O. Validação da versão portuguesa da situational motivation scale (sims) em contextos académicos. In: UNIVERSIDADE DO MINHO. **Atas do XII Congresso Galego-Português de Psicopedagogia, 11-13 de setembro de 2013**. [S.l.], 2013. p. 4868–4882.

GOLDMAN, M.; LITTLE, G.; MILLER, R. C. Collabode: collaborative coding in the browser. In: ACM. **Proceedings of the 4th international workshop on Cooperative and human aspects of software engineering**. Waikiki, Honolulu, HI, USA, 2011. p. 65–68.

GOMES, E. R. **Objetos inteligentes de aprendizagem: uma abordagem baseada em agentes para objetos de aprendizagem**. Dissertação (Mestrado) — Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre., 2012.

GUAY, F.; VALLERAND, R. J.; BLANCHARD, C. On the assessment of situational intrinsic and extrinsic motivation: The situational motivation scale (sims). **Motivation and emotion**, Springer, v. 24, n. 3, p. 175–213, 2000.

HARACKIEWICZ, J. M.; ELLIOT, A. J. Achievement goals and intrinsic motivation. **Journal of personality and social psychology**, American Psychological Association, v. 65, n. 5, p. 904, 1993.

IBGE. **Censo 2010**. 2010. Instituto Brasileiro de Geografia e Estatística. Disponível em: <<http://www.censo2010.ibge.gov.br/>>.

KARSENTY, L. Cooperative work and shared visual context: An empirical study of comprehension problems in side-by-side and remote help dialogues. **Human-Computer Interaction**, Taylor & Francis, v. 14, n. 3, p. 283–315, 1999.

KRASHEN, S. **Principles and Practice in Second Language Acquisition**. First. Oxford, UK: Oxford Pergamon, 1982.

KRASNER, G. E.; POPE, S. T. *et al.* A description of the model-view-controller user interface paradigm in the smalltalk-80 system. **Journal of object oriented programming**, v. 1, n. 3, p. 26–49, 1988.

KRAUT, R. E.; GERGLE, D.; FUSSELL, S. R. The use of visual information in shared visual spaces: Informing the development of virtual co-presence. In: ACM. **Proceedings of the 2002 ACM conference on Computer supported cooperative work**. [S.l.], 2002. p. 31–40.

LAVE, J.; WENGER, E. **Situated learning: Legitimate peripheral participation**. New York: Cambridge University Press, 1991.

LEWIS, J. R. Psychometric evaluation of the post-study system usability questionnaire: The pssuq. In: SAGE PUBLICATIONS. **Proceedings of the Human Factors and Ergonomics Society Annual Meeting**. Atlanta, Georgia, USA, 1992. v. 36, n. 16, p. 1259–1260.

LEWIS, J. R. Multipoint scales: Mean and median differences and observed significance levels. **International Journal of Human-Computer Interaction**, Taylor & Francis, v. 5, n. 4, p. 383–392, 1993.

LEWIS, J. R. Ibm computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. **International Journal of Human-Computer Interaction**, Taylor & Francis, v. 7, n. 1, p. 57–78, 1995.

LOWMAN, J. **Active learning: Beyond the classroom**. 2000.

LTS, C. Ieee learning technology standards committee (ltsc): Ieee p1484. 12 learning object metadata working group (home page). 2000.

MACIEL, I. M. Educação a distancia. ambiente virtual: construindo significados. **Boletim Técnico do Senac**, v. 28, n. 3, p. 38–45, 2002.

MACKENZIE, I. S. **Human-Computer Interaction: An Empirical Research Perspective**. Amsterdam: Elsevier (Morgan Kaufmann), 2013. ISBN 978-0-12-405865-1. Disponível em: <<http://www.sciencedirect.com/science/book/9780124058651>>.

MANDERLINK, G.; HARACKIEWICZ, J. M. Proximal versus distal goal setting and intrinsic motivation. **Journal of Personality and Social Psychology**, American Psychological Association, v. 47, n. 4, p. 918, 1984.

MARTÍN-ALBO, J.; NÚÑEZ, J. L.; NAVARRO, J. G. Validation of the spanish version of the situational motivation scale (emsi) in the educational context. **The Spanish journal of psychology**, Cambridge Univ Press, v. 12, n. 02, p. 799–807, 2009.

MARTINS, M. d. L. O. O papel da usabilidade no ensino a distância mediado por computador. **Belo Horizonte**, 2004.

MCCARTHY, J. C.; MILES, V. C.; MONK, A. F.; HARRISON, M. D.; DIX, A. J.; WRIGHT, P. C. Text-based on-line conferencing: a conceptual and empirical analysis using a minimal prototype. **Human-Computer Interaction**, L. Erlbaum Associates Inc., v. 8, n. 2, p. 147–183, 1993.

MCKEACHIE, W. J. **Teaching and learning in the college classroom: A review of the research literature**. Michigan, USA: University of Michigan Press, 1986. v. 86.

MEIRA, L.; LERMAN, S. Zones of proximal development as fields for communication and dialogue. **Challenges and strategies for studying human development in cultural contexts**, p. 199–219, 2009.

NIELSEN, J.; BUDI, R. **Mobile Usability**. San Francisco, USA: Pearson Education, 2013. ISBN 9780133122176.

OLIVEIRA, F. C. D. M. B.; FREITAS, A. T. D.; QUEIROZ, B. D. S.; ARAUJO, T. A. C. D.; SILVA, L. C.; SOARES, É. F. It education strategies for the deaf. In: ACM. **18th International Conference on Enterprise Information Systems**. Rome, Italy, 2016. p. 473–482.

QUADROS, R. M. de; KARNOPP, L. B. **Língua de sinais brasileira: estudos lingüísticos**. Porto Alegre, RS: Artmed, 2007.

RYAN, R. M.; DECI, E. L. A self-determination theory approach to psychotherapy: The motivational basis for effective change. **Canadian Psychology/Psychologie canadienne**, Educational Publishing Foundation, v. 49, n. 3, p. 186, 2008.

SANTOS, R.; MAGALHAES, C. V. C.; NETO, J. S. C.; JÚNIOR, S. S. L. P. Proglib: Uma linguagem de programação baseada na escrita de libras. In: **Proc. XXII Simposio Brasileiro de Informatica na Educacao and XVII Workshop de Informatica na Escola (SBIE-WIE)**. Aracaju, Brazil: [s.n.], 2011. p. 1533–1542. Disponível em: <<http://www.br-ie.org/pub/index.php/wie/article/view/1984/1743>>.

SÉRGIO, J. d. S. R. L. **Presença social, percepções, relações, interpessoais e grau de satisfação com a curso de E-Learning**. Dissertação (Mestrado) — Universidade Aberta, 2007.

SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). **Biometrika**, JSTOR, v. 52, n. 3/4, p. 591–611, 1965.

SHORT, J.; WILLIAMS, E.; CHRISTIE, B. **The social psychology of telecommunications**. John Wiley and Sons Ltd, 1976.

SILVA, L. C.; OLIVEIRA, F. C. de M.; OLIVEIRA, A. C. de; FREITAS, A. T. de. Introducing the jload: A java learning object to assist the deaf. In: IEEE. **Advanced Learning Technologies (ICALT), 2014 IEEE 14th International Conference on**. Athens, Greece, 2014. p. 579–583.

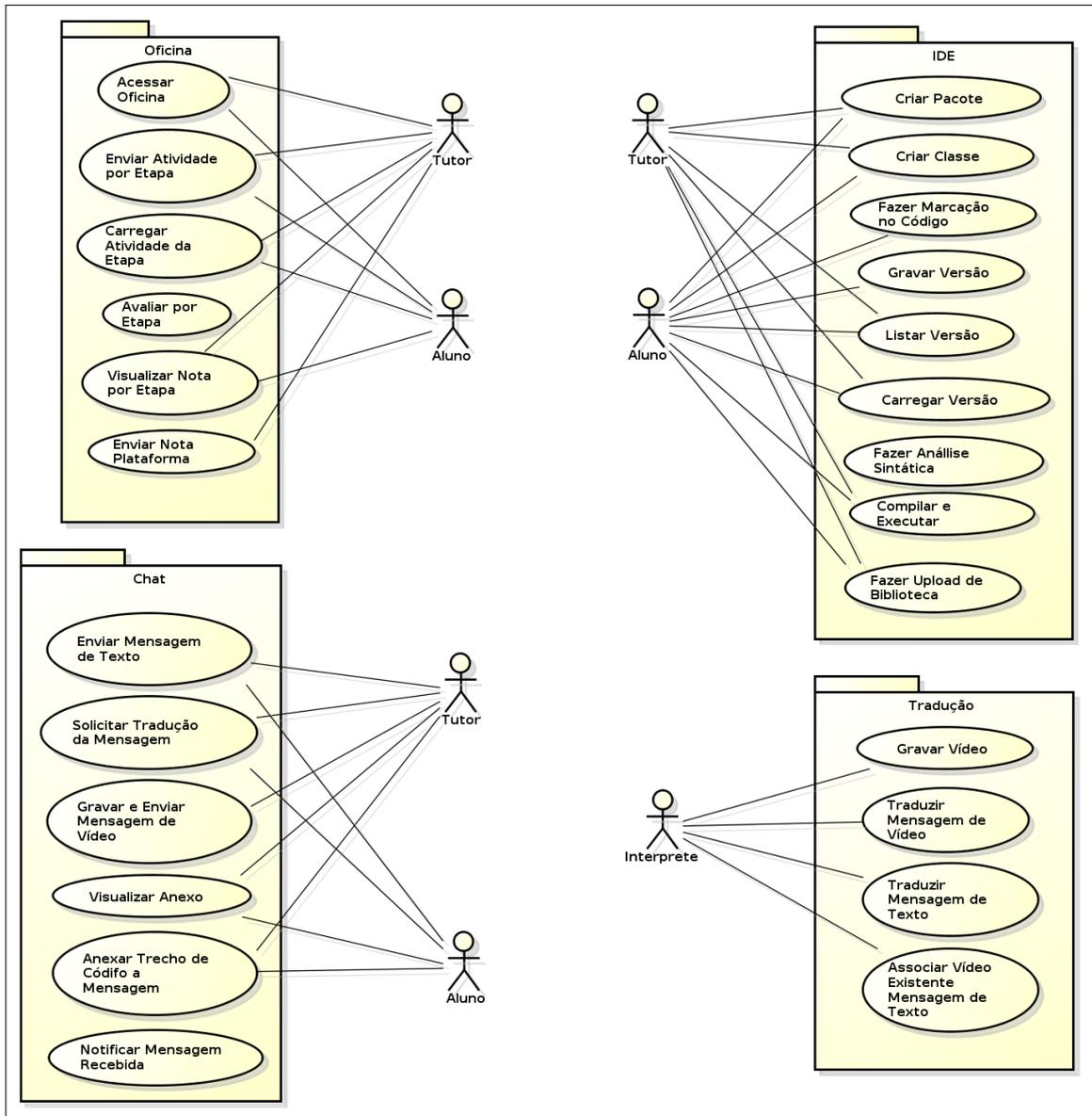
SISTO, F. F.; SISTO, F.; BORUCHOVITCH, E.; FINI, L.; BRENELLI, R.; MARTINELLI, S. Dificuldades de aprendizagem. **Dificuldades de aprendizagem no contexto psicopedagógico**, Vozes Petrópolis eRJ, p. 19–39, 2001.

- STROBEL, K. História da educação de surdos. **Florianópolis: Universidade Federal de Santa Catarina**, 2009. Material da disciplina História da Educação dos Surdos.
- SUCHMAN, L. A. **Plans and situated actions: The problem of human-machine communication**. New York, NY, USA: Cambridge university press, 1987. ISBN 0-521-33137-4.
- TAILLE, Y. de L.; OLIVEIRA, M. de; DANTAS, H. **Piaget, Vygotsky, Wallon: teorias psicogenéticas em discussão**. Rio de Janeiro, RJ: Summus, 1992. ISBN 9788532304124. Disponível em: <<https://books.google.com.br/books?id=29awWGSPyFYC>>.
- THOMA, A. d. S. Comunidades, cultura, identidades e movimento surdo. **Apostila do Curso de LIBRAS I**, 2009.
- VALLERAND, R. J. Toward a hierarchical model of intrinsic and extrinsic motivation. **Advances in experimental social psychology**, Elsevier, v. 29, p. 271–360, 1997.
- VALLERAND, R. J.; PELLETIER, L. G.; KOESTNER, R. Reflections on self-determination theory. **Canadian Psychology/Psychologie canadienne**, Educational Publishing Foundation, v. 49, n. 3, p. 257, 2008.
- VYGOTSKY, L. S. **Thought and Language**. Cambridge, MA: MIT Press (reprinted in 1986), 1934.
- VYGOTSKY, L. S. Concrete human psychology. **Soviet psychology**, Taylor & Francis, v. 27, n. 2, p. 53–77, 1989.
- WERTSCH, J. V.; ZANÓN, J.; CORTÉS, M. **Vygotsky y la formación social de la mente**. Barcelona, ESP: Paidós Barcelona, 1988.
- WHITTAKER, S.; GEELHOED, E.; ROBINSON, E. Shared workspaces: how do they work and when are they useful? **International Journal of Man-Machine Studies**, Elsevier, v. 39, n. 5, p. 813–842, 1993.
- ZOGBI, P. **Conheça o setor que tem mais vagas que profissionais no Brasil**. 2016. Disponível em: <<http://www.infomoney.com.br/carreira/emprego/noticia/4447259/conheca-setor-que-tem-mais-vagas-que-profissionais-brasil>>.

## **APÊNDICES**

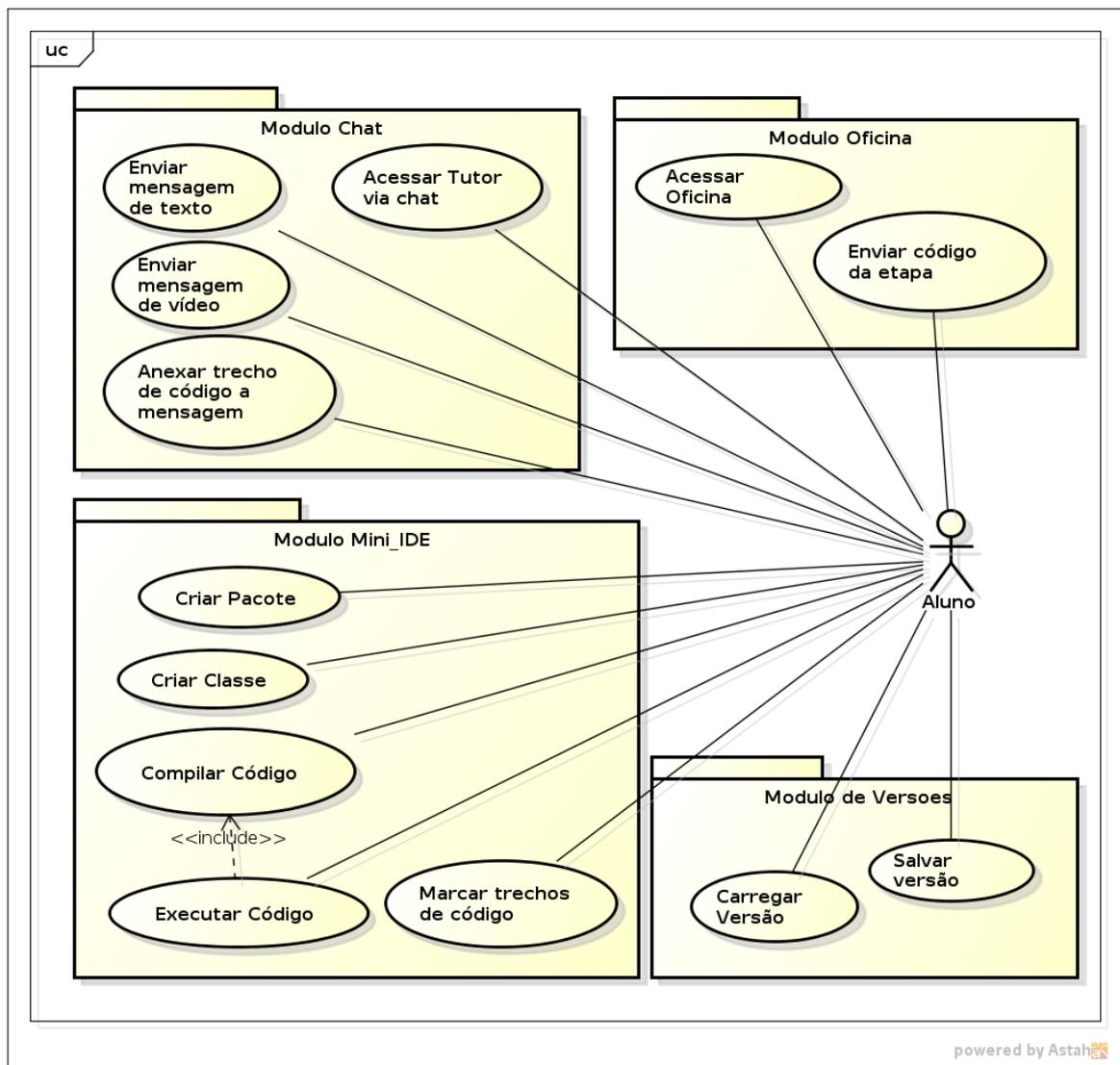
APÊNDICE A – Diagramas de Caso de Uso do JLoad

Figura 28 – Diagrama de Caso de Uso dos Atores do JLoad



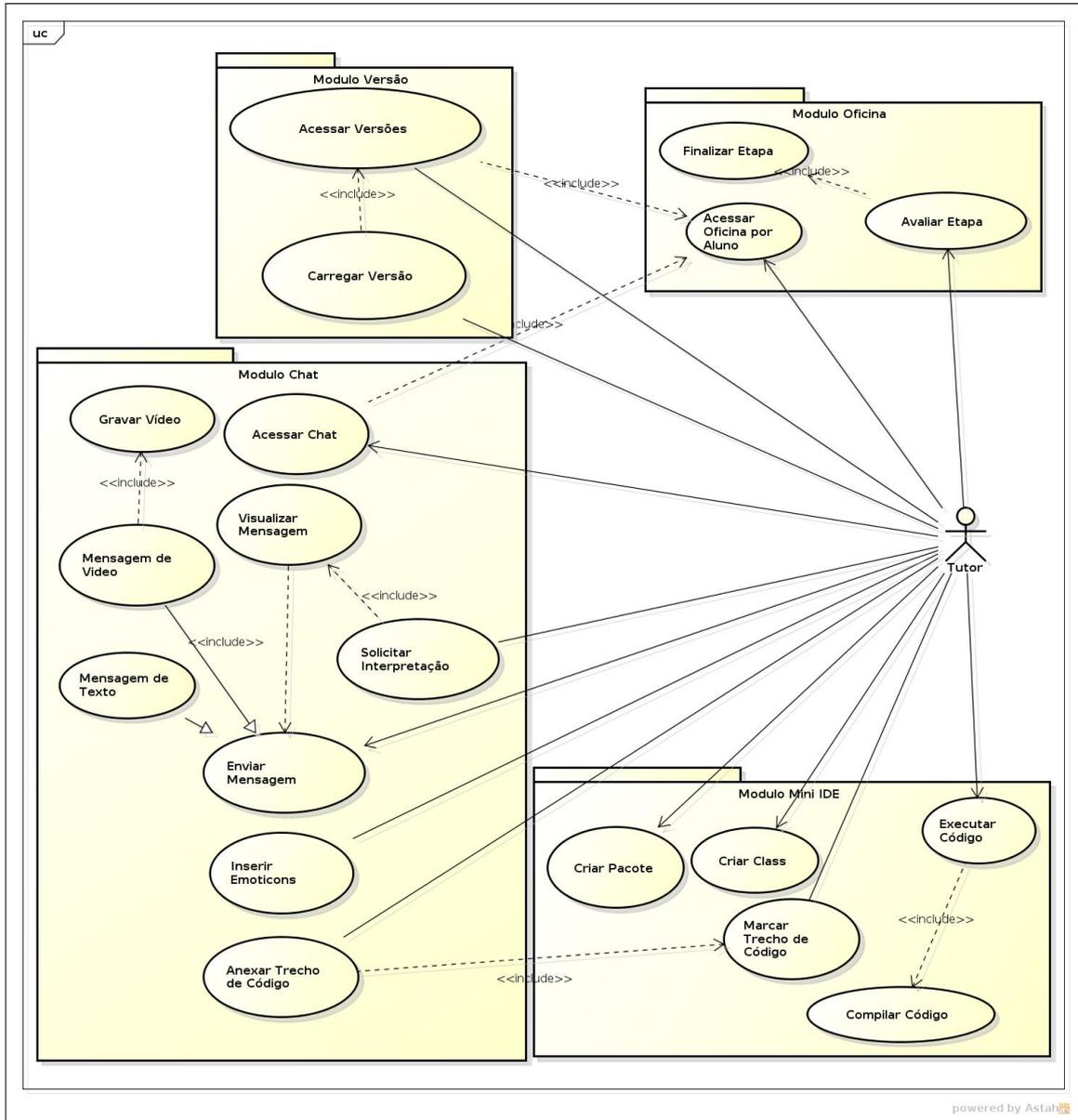
Fonte: A autora.

Figura 29 – Diagrama de Caso de Uso do Aluno



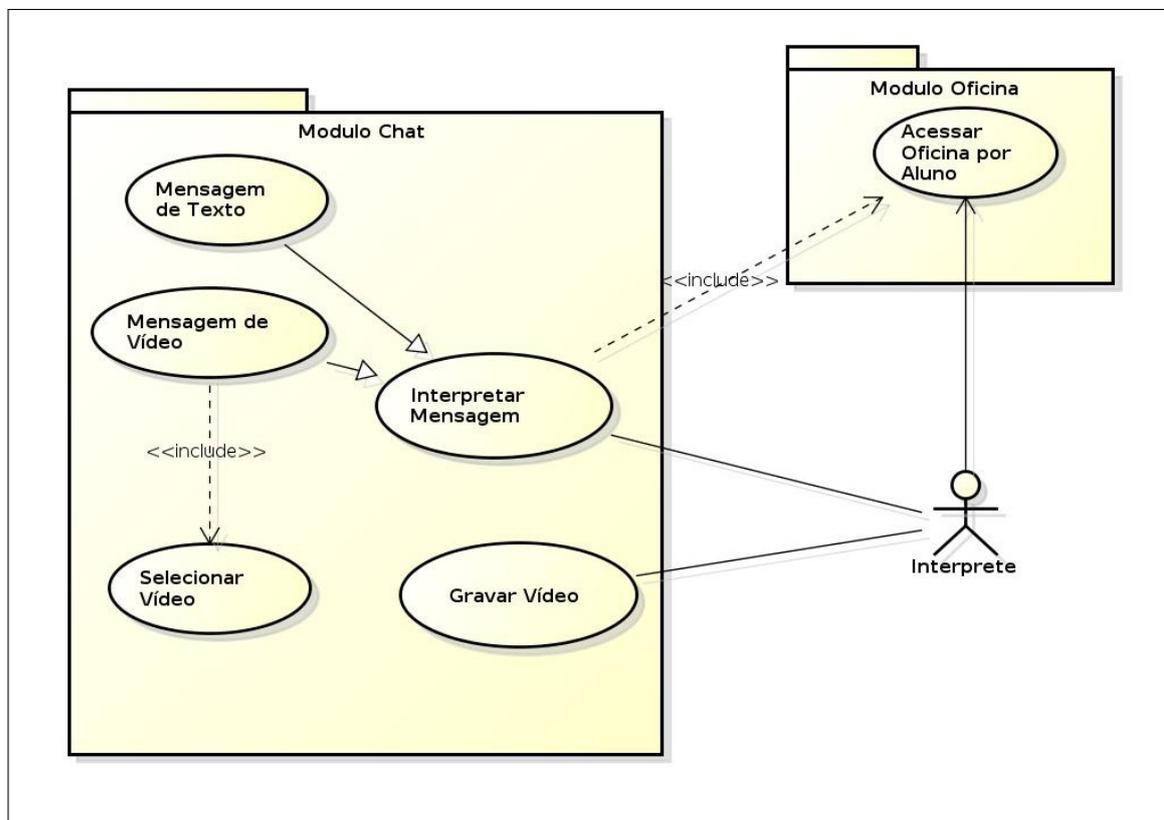
Fonte: A autora.

Figura 30 – Diagrama de Caso de Uso do Tutor



Fonte: A autora.

Figura 31 – Diagrama de Caso de Uso do Intérprete



Fonte: A autora.

## APÊNDICE B – OFICINAS DESENVOLVIDAS NO EXPERIMENTO

### B.1 OFICINA ECLIPSE

Nesta aula você viu como fazer para mostrar uma mensagem na tela utilizando a linguagem Java. Vamos praticar um pouco e criar um código Java que mostre uma mensagem na tela.

#### Etapa 1

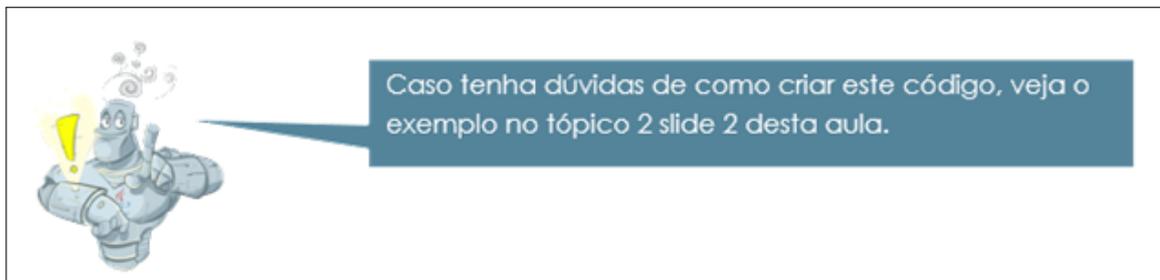
Vamos criar um código que mostre no console a seguinte mensagem:



```
<terminated> HelloWorld (3) [Java Application] C:\Program Files\Java\jre7\bin\javaw.exe (14/11/2013 12:26:32)
 Bem vindo!
 Seu nome aqui
```

#### Especificações:

- Crie uma classe chamada **Exemplo01** (Não esqueça de marcar a opção de criar o método main automaticamente).
- Crie dois comandos **System.out.println()** para mostrar cada mensagem.



## B.2 OFICINA JLOAD

Nesta aula você viu como fazer para mostrar uma mensagem na tela utilizando a linguagem Java. Vamos praticar um pouco e criar um código Java que mostre uma mensagem na tela.

### **Etapa 1**

- a) Crie um pacote chamado minhasclasses no PROJETO\_JLOAD.

### **Etapa 2**

- a) Crie uma classe chamada MinhaClasse dentro do pacote minhasclasses.
- b) Crie o método main dentro da classe que você criou.

### **Etapa 3**

- a) Crie o comando `System.out.println()` para mostrar no console a frase abaixo abaixo: “Aula 01 de Java”
- b) Utiliza o caractere `+` para unir ( concatenar) a frase mostrado na etapa 03 com a frase abaixo:  
“Meu primeiro código Java”  
Exemplo: “frase 1” + “frase 2”.
- c) Mostre no console as frases concatenadas utilizando o comando `System.out.println()`.

### **Etapa 4**

- a) Execute o código e verifique se as mensagens são mostradas no console.

Pronto! Chegamos no fim desta atividade! Agora, clique no botão ENVIAR para enviar a atividade para o Tutor.

**ANEXOS**

## ANEXO A – Questionário PSSUQ em inglês (LEWIS, 1995)

### **Post-Study System Usability Questionnaire Items**

1. Overall, I am satisfied with how easy it is to use this system
2. It was simple to use this system
3. I could effectively complete the tasks and scenarios using this system
4. I was able to complete the tasks and scenarios quickly using this system
5. I was able to efficiently complete the tasks and scenarios using this system
6. I felt comfortable using this system
7. It was easy to learn to use this system
8. I believe I could become productive quickly using this system
9. The system gave error messages that clearly told me how to fix problems
10. Whenever I made a mistake using the system, I could recover easily and quickly
11. The information (such as on-line help, on-screen messages, and other documentation) provided with this system was clear
12. It was easy to find the information I needed
13. The information provided for the system was easy to understand
14. The information was effective in helping me complete the tasks and scenarios
15. The organization of information on the system screens was clear
16. The interface of this system was pleasant
17. I liked using the interface of this system
18. This system has all the functions and capabilities I expect it to have
19. Overall, I am satisfied with this system