



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO AMAZONAS
CAMPUS MANAUS CENTRO
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO
DE SISTEMAS**

RENATO PEREIRA FERREIRA

**UMLIBRAS WEB: FERRAMENTA COMPUTACIONAL PARA CRIAÇÃO DE
DIAGRAMAS DE SEQUÊNCIA COM ACESSIBILIDADE EM LIBRAS**

**MANAUS – AM
2022**

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
DO AMAZONAS
CAMPUS MANAUS CENTRO
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO
DE SISTEMAS**

RENATO PEREIRA FERREIRA

**UMLIBRAS WEB: FERRAMENTA COMPUTACIONAL PARA CRIAÇÃO DE
DIAGRAMAS DE SEQUÊNCIA COM ACESSIBILIDADE EM LIBRAS**

Trabalho de conclusão de curso apresentado à banca examinadora do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas – IFAM, Campus Manaus Centro, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. M.Sc. Sergio Augusto C. Bezerra

**MANAUS – AM
2022**

Biblioteca do IFAM – Campus Manaus Centro

F383u Ferreira, Renato Pereira.

Umlibras web: ferramenta computacional para criação de diagramas de sequência com acessibilidade em libras / Renato Pereira Ferreira. – Manaus, 2022.

62 p. : il. color.

Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistema) – Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, *Campus Manaus Centro*, 2022.

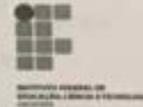
Orientador: Prof. Me. Sergio Augusto Coelho Bezerra.

1. Desenvolvimento de sistema. 2. Educação inclusiva. 3. Tecnologia assistiva. 4. Acessibilidade. 5. Libras. I. Bezerra, Sergio Augusto Coelho. (Orient.) II. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas III. Título.

CDD 005.3



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO MÉDIA E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA - AM
DEPARTAMENTO ACADÊMICO DE INFORMAÇÃO E COMUNICAÇÃO
CURSO DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

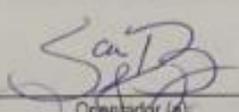


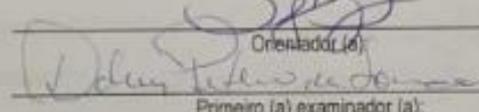
TERMO DE APROVAÇÃO

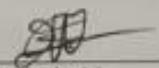
A monografia, que tem como título: UMLIBRAS WEB: FERRAMENTA COMPUTACIONAL PARA CRIAÇÃO DE DIAGRAMAS DE SEQUÊNCIA COM ACESSIBILIDADE EM LIBRAS foi submetida à defesa pública, sob a avaliação de banca examinadora, como parte dos requisitos necessários para a obtenção do título de graduação do curso superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

AUTOR (A): RENATO PEREIRA FERREIRA

Monografia aprovada em: 15 / 12 / 2022


Orientador(a):


Primeiro (a) examinador (a):


Segundo (a) examinador (a):

AGRADECIMENTOS

A Deus por tudo que tem proporcionado em minha vida.

Aos meus pais Maria do Rosário e Raimundo Ferreira por me ensinarem somente o que eu precisava aprender.

A minha namorada, Arilene Ferreira, por todo o incentivo e apoio.

Ao meu orientador, Prof. M.Sc. Sergio Augusto C. Bezerra, pelo tema sugerido e por sua excepcional orientação.

Ao Instituto Federal de Educação, Ciência e Tecnologia do Amazonas IFAM - Campus Manaus Centro, aos professores, à coordenação do curso TADS e a todos os colaboradores.

RESUMO

Inserido no contexto da educação inclusiva e tecnologia assistiva, este trabalho desenvolveu uma ferramenta computacional para criação de diagramas de sequência da *Unified Modeling Language* (UML), com recursos de acessibilidade em Libras, que subsidiará o ensino-aprendizagem do discente, com ou sem deficiência auditiva, na modelagem de sistemas computacionais. O docente terá um recurso didático-pedagógico para auxiliá-lo na exposição do conteúdo relacionado ao diagrama de sequência, favorecendo a aprendizagem do discente com deficiência auditiva e dessa forma promovendo a educação inclusiva. Atualmente é grande a escassez de ferramentas adequadas ao contexto e às necessidades do aluno com deficiência auditiva. Sob esta perspectiva, a ferramenta desenvolvida busca dar suporte ao professor nas atividades propostas, criando condições para a aprendizagem desses alunos. Denominada UMLibras, a ferramenta foi construída utilizando como estrutura base a biblioteca *JavaScript mxGraph* e a *Application Programming Interface* (API) VLibras, visando auxiliar na modelagem do diagrama de sequência por meio do uso de vídeos, ilustrações e o idioma Libras. Dessa forma, espera-se contribuir com o ensino-aprendizagem do discente com ou sem deficiência auditiva, tornando mais fácil seu acesso à informação e ao conhecimento proposto.

Palavras-chave: Educação inclusiva, tecnologia assistiva, deficiência auditiva, acessibilidade, Libras.

ABSTRACT

Inserted in the context of inclusive education and assistive technology, this work developed a computational tool for creating sequence diagrams of the Unified Modeling Language (UML), with accessibility resources in Libras, which will subsidize the teaching-learning of students, with or without disabilities hearing, in the modeling of computational systems. The professor will have a didactic-pedagogical resource to help him in exposing the content related to the sequence diagram, favoring the learning of students with hearing impairment and thus promoting inclusive education. Currently, there is a great shortage of tools suitable for the context and needs of students with hearing impairments. From this perspective, the developed tool seeks to support the teacher in the proposed activities, creating conditions for the learning of these students. Named UMLibras, the tool was built using the mxGraph JavaScript library and the VLibras Application Programming Interface (API) as a base structure, aiming to assist in modeling the sequence diagram through the use of videos, illustrations and the Libras language. In this way, it is expected to contribute to the teaching-learning of students with or without hearing impairment, making their access to information and proposed knowledge easier.

Keywords: Inclusive education, assistive technology, hearing impairment, accessibility, Libras.

LISTA DE FIGURAS

Figura 1 - Modelo cascata de ciclo de vida de desenvolvimento de software	17
Figura 2 - Diagrama de Sequência para ver informações de pacientes.	23
Figura 3 - Dispositivos com VLibras.....	24
Figura 4 - Personagem Hugo.....	24
Figura 5 - Tradutor virtual Rybená	25
Figura 6 - Logo da Ava	26
Figura 7 - Jornal Primeira Mão.....	26
Figura 8 - Arquitetura da biblioteca mxGraph	29
Figura 9 - UMLibras desktop.....	30
Figura 10 - Diagrama de caso de uso.....	35
Figura 11 - Modelo entidade-relacionamento da ferramenta UMLibras	37
Figura 12 - Modelo lógico (forma textual) da ferramenta UMLibras	38
Figura 13 - Diagrama Lógico da ferramenta UMLibras	39
Figura 14 - Diagrama de classes	40
Figura 15 - Protótipo da tela de login.....	41
Figura 16 - Protótipo da tela de cadastro de usuário	41
Figura 17 - Protótipo da tela principal da ferramenta UMLibras.....	42
Figura 18 - Protótipo da tela de gerenciamento de mídias.....	42
Figura 19 - Código para carregar a biblioteca mxGraph	43
Figura 20 - Código para a construção de um gráfico	43
Figura 21 - Código para inserir vértices e arestas	44
Figura 22 - Código para criar um modelo de gráfico.....	45
Figura 23 - Código para modificar estilos	45
Figura 24 - Código para adicionar folha de estilo.....	46
Figura 25 - Código de um objeto JavaScript.....	46
Figura 26 - Código para converter um objeto em XML	47
Figura 27 - Resultado do objeto convertido em XML	47
Figura 28 - Exemplo de arquivo PHP.....	48
Figura 29 - Código para integrar a API Vlibras à um website	48
Figura 30 - Código para criar as tabelas do banco de dados da UMLibras.	49

Figura 31 - Diagrama de sequência de atendimento a um cliente	50
Figura 32 - Tela de login da ferramenta UMLibras	51
Figura 33 - Tela de cadastro de usuário	51
Figura 34 - Tela principal com um diagrama de sequência modelado...	52
Figura 35 - Paleta diagrama de sequência	52
Figura 36 - Menu e barra de ferramenta horizontal	53
Figura 37 - Janela exportar imagem	53
Figura 38 - Vídeo de introdução	54
Figura 39 - Janela de introdução	54
Figura 40 - Palheta Diagrama.....	55
Figura 41 - Propriedades de um elemento selecionado.....	55
Figura 42 - Diagrama de sequência.....	56
Figura 43 - Tela de gerenciamento de mídias	56

LISTA DE TABELAS

Tabela 1 - Ferramentas UML disponíveis no mercado	31
Tabela 2 - Lista dos requisitos funcionais da ferramenta.....	33
Tabela 3 - Lista dos requisitos não funcionais da ferramenta.....	33
Tabela 4 - Caso de uso criar diagrama.....	35
Tabela 5 - Caso de uso abrir VLibras	36
Tabela 6 - Caso de uso exibir informações.....	36
Tabela 7 - Caso de uso salvar diagrama	61
Tabela 8 - Caso de uso abrir diagrama.....	61
Tabela 9 - Caso de uso exportar diagrama.....	62
Tabela 10 - Caso de uso exibir introdução	62

LISTA DE SIGLAS

API	Application Programming Interface
ASF	Apache Software Foundation
BD	Banco de Dados
CSS	Cascading Style Sheets
DER	Diagrama Entidade-Relacionamento
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IBGE	Instituto Brasileiro de Geografia e Estatística
LIBRAS	Língua Brasileira de Sinais
RF	Requisito Funcional
RNF	Requisito Não Funcional
SGBD	Sistema Gerenciador de Banco de Dados
SQL	Structured Query Language
UC	Use case
UML	Unified Modeling Language
W3C	World Wide Web Consortium
XML	Extensible Markup Language

SUMÁRIO

1	INTRODUÇÃO	14
1.1	CONTEXTUALIZAÇÃO E JUSTIFICATIVA	14
1.2	OBJETIVOS	16
1.2.1	<i>Objetivo geral</i>	16
1.2.2	<i>Objetivos específicos</i>	16
1.3	METODOLOGIA.....	16
1.4	ORGANIZAÇÃO DA MONOGRAFIA	18
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	ACESSIBILIDADE.....	20
2.2	LIBRAS.....	21
2.3	TECNOLOGIA UML	21
2.4	DIAGRAMA DE SEQUÊNCIA	22
2.5	TECNOLOGIAS ASSISTIVAS.....	23
2.5.1	<i>VLibras</i>	24
2.5.2	<i>Hand Talk</i>	24
2.5.3	<i>Transcrição Instantânea</i>	24
2.5.4	<i>Prodeaf</i>	25
2.5.5	<i>WebLibras</i>	25
2.5.6	<i>Rybená</i>	25
2.5.7	<i>& Ava</i>	26
2.5.8	<i>Primeira Mão</i>	26
3	TECNOLOGIAS UTILIZADAS	27
3.1	SERVIDOR APACHE	27
3.2	MYSQL	27
3.3	HTML	27
3.4	CSS.....	28
3.5	JAVASCRIPT	28
3.6	VLBRAS.....	28
3.7	MXGRAPH	29
4	TRABALHOS RELACIONADOS	30
4.1	UMLBRAS VERSÃO DESKTOP.....	30

4.2	FERRAMENTAS SEMELHANTES.....	31
5	FERRAMENTA DESENVOLVIDA	32
5.1	LEVANTAMENTO E ANÁLISE DE REQUISITOS.....	32
5.1.1	<i>Requisitos funcionais</i>	<i>32</i>
5.1.2	<i>Requisitos não-funcionais</i>	<i>33</i>
5.2	PROJETO	34
5.2.1	<i>Estrutura básica</i>	<i>34</i>
5.2.2	<i>Diagrama de caso de uso</i>	<i>34</i>
5.2.3	<i>Descrição dos casos de uso</i>	<i>35</i>
5.2.4	<i>Modelo conceitual do banco de dados.....</i>	<i>37</i>
5.2.5	<i>Modelo lógico do banco de dados.....</i>	<i>38</i>
5.2.6	<i>Diagrama de classes.....</i>	<i>39</i>
5.2.7	<i>Interface</i>	<i>40</i>
5.3	CODIFICAÇÃO	43
5.3.1	<i>Biblioteca mxGraph.....</i>	<i>43</i>
5.3.2	<i>Gerar gráfico</i>	<i>43</i>
5.3.3	<i>Vértices e arestas.....</i>	<i>44</i>
5.3.4	<i>Modelo</i>	<i>45</i>
5.3.5	<i>Folha de estilo.....</i>	<i>45</i>
5.3.6	<i>CSS.....</i>	<i>46</i>
5.3.7	<i>Gerar XML do gráfico</i>	<i>46</i>
5.3.8	<i>Arquivos</i>	<i>47</i>
5.3.9	<i>Integração VLibras</i>	<i>48</i>
5.3.10	<i>Banco de Dados.....</i>	<i>48</i>
5.4	TESTE.....	49
5.5	IMPLEMENTAÇÃO.....	50
5.5.1	<i>Tela de login.....</i>	<i>50</i>
5.5.2	<i>Tela principal.....</i>	<i>52</i>
6	CONSIDERAÇÕES FINAIS	57
6.1	TRABALHOS FUTUROS	57
7	REFERÊNCIAS.....	58
8	APÊNDICE A	61

1 INTRODUÇÃO

1.1 Contextualização e justificativa

Atualmente a educação inclusiva cria um ensino igualitário que consegue atender o grupo das pessoas com necessidades especiais.

Conforme Lei nº 13.146, de 6 de julho de 2015 (Lei Brasileira de Inclusão da Pessoa com Deficiência):

Art. 2º Considera-se pessoa com deficiência aquela que tem impedimento de longo prazo de natureza física, mental, intelectual ou sensorial, o qual, em interação com uma ou mais barreiras, pode obstruir sua participação plena e efetiva na sociedade em igualdade de condições com as demais pessoas (BRASIL, 2015).

A educação inclusiva permite que indivíduos com necessidades especiais possam acompanhar o ensino do mesmo modo que os seus colegas sem qualquer prejuízo ao desempenho acadêmico (ORTEGA, 2019).

Atrelada a educação inclusiva estão as tecnologias assistivas, sendo consideradas quaisquer ferramentas utilizadas com a finalidade de proporcionar acessibilidade, maior independência e autonomia à pessoa com necessidade especial, inclusive servindo de apoio ao professor.

As pessoas com deficiência auditiva ou surdez têm a acessibilidade dificultada pela barreira da comunicação, sendo esta fundamental para o desenvolvimento social e intelectual do indivíduo.

No Brasil, segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), 5% da população brasileira é formada por pessoas surdas, o que corresponde a mais de 10 milhões de pessoas. Desse total, mais de 2,7 milhões apresentam deficiência auditiva severa (IBGE, 2010).

Nesse contexto as tecnologias assistivas são de fundamental importância na inclusão e na diminuição da barreira entre surdos e ouvintes.

Um exemplo de tecnologia assistiva é a suíte VLibras, um conjunto de ferramentas computacionais, de código aberto, que traduz conteúdos digitais - texto, áudio e vídeo - para Libras (BRASIL, 2022).

Libras é a sigla da Língua Brasileira de Sinais, uma língua de modalidade gestual-visual onde é possível se comunicar através de gestos, expressões faciais e corporais. A Libras é muito utilizada na comunicação com pessoas surdas, sendo, portanto, uma importante ferramenta de inclusão social (CRISTIANO, 2017).

Atualmente, existem muitas tecnologias assistivas que auxiliam a comunidade surda. Como exemplos de aplicativos constam VLibras, Rybená, Hand Talk, ProDeaf, WebLibras, &Ava, Primeira Mão e Transcrição Instantânea, dos quais a maioria traduz textos e áudios para Libras ou converter português escrito para voz, funcionando em computadores ou dispositivos móveis. O Whats app também é outro aplicativo bastante utilizado pela comunidade surda, com ele é possível a emissão de mensagens, viabilizando a comunicação para surdos que entendem a língua portuguesa.

Entretanto, tecnologias como a UML não dispõem de nenhuma ferramenta com acessibilidade, exceto a primeira versão criada da ferramenta UMLibras. Um trabalho pioneiro desenvolvido pelo aluno Antonio José Barbosa Fonseca do curso de Tecnologia em Análise e Desenvolvimento de Sistemas - TADS do Instituto Federal de Educação, Ciência e Tecnologia do Amazonas - IFAM, Campus Manaus Centro, sob orientação e responsabilidade do professor M.Sc. Sergio Augusto C. Bezerra, que foi o idealizador da ferramenta UMLibras. Essa versão foi desenvolvida para a plataforma desktop e contemplou o módulo que auxilia a elaboração do Diagrama de Caso de Uso da UML. Na perspectiva que a ferramenta computacional UMLibras possibilitasse a construção dos demais diagramas da UML, então o professor Sergio Bezerra fez uma conscientização e um chamamento para que outros alunos pudessem colaborar com o desenvolvimento dos demais módulos, inclusive mudando da plataforma desktop para Web.

Nesse sentido, este trabalho colabora com o desenvolvimento do módulo Web da ferramenta UMLibras voltado à construção de diagramas de sequência, a fim de subsidiar o professor quanto ao ensino e o aprendizado por parte do discente com ou sem deficiência auditiva.

1.2 Objetivos

1.2.1 Objetivo geral

Desenvolver uma ferramenta computacional para auxiliar docentes no ensino e facilitar a aprendizagem do discente com ou sem deficiência auditiva, na criação de diagramas de sequência da *Unified Modeling Language* (UML) com recursos de acessibilidade em Libras.

1.2.2 Objetivos específicos

Com o propósito de alcançar o objetivo geral deste trabalho, foram estabelecidos os objetivos específicos listados a seguir:

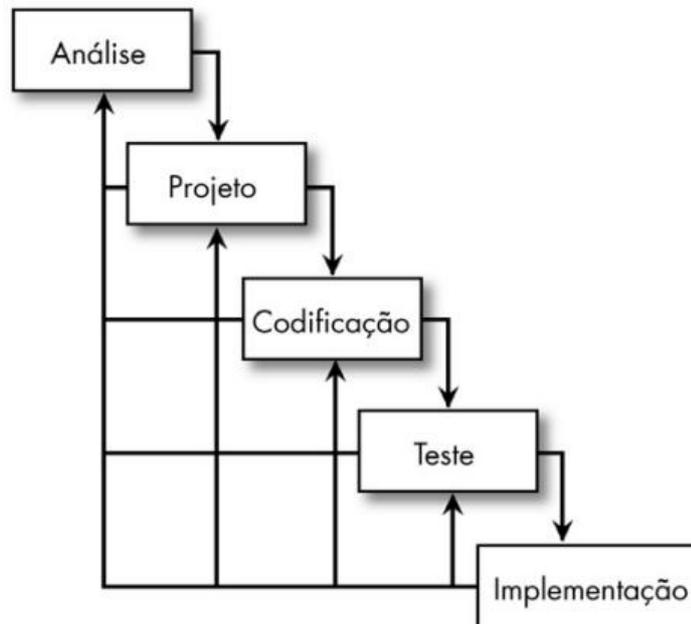
- Tornar acessível para pessoas surdas o processo de criação de diagrama de sequência da UML;
- Integrar a tecnologia assistiva VLibras na utilização da ferramenta;
- Produzir vídeos com intérpretes em Libras específicos para subsidiar o aprendizado das funcionalidades do diagrama de sequência;
- Incorporar à ferramenta imagens autoexplicativas e *links* para que abram GIFs ou vídeos em Libras;

1.3 Metodologia

Um processo de desenvolvimento de software é representado por um modelo, enquanto o modelo é operacionalizado por meio de uma metodologia. Existem diversos modelos de processo de desenvolvimento de *software*, e cada modelo pode ter mais do que uma metodologia que o operacionaliza. A metodologia estabelece basicamente a sequência das atividades e como elas se relacionam entre si, identificando o momento em que os métodos e as ferramentas serão utilizados (AUDY; PRIKLADNICKI, 2008).

A Figura 1 apresenta o modelo de desenvolvimento de software denominado modelo em cascata ou modelo clássico, onde o mesmo é dividido em várias fases distintas que seguem um padrão lógico e sequencial. Cada fase precisa ser concluída para passar à próxima fase e podem ocorrer retornos para a fase anterior.

Figura 1 - Modelo cascata de ciclo de vida de desenvolvimento de software.



Fonte:(AUDY; PRIKLADNICKI, 2008)

Os ciclos de vida especificam algumas atividades que devem ser executadas, assim como a sua ordem. Sua função básica é diminuir os problemas encontrados no processo como um todo (AUDY; PRIKLADNICKI, 2008). Como o nome cascata sugere, cada etapa do projeto deságua na próxima, formando a ideia de uma cachoeira.

Na primeira etapa, chamada análise, os requisitos necessários para a ferramenta foram coletados e anotados em um documento de especificação, abstraindo o que a ferramenta deveria fazer. A partir deste documento de requisitos foi possível planejar todas as outras fases do ciclo de vida.

Na segunda etapa foram gerados os modelos e a lógica de negócio aplicada à ferramenta. Nesta fase, um conjunto de informações baseadas nas necessidades do usuário orientou como as regras de negócio foram implementadas.

As regras de negócio são instruções que definem ou restringem comportamentos, mostrando como os procedimentos devem ser seguidos.

Na terceira etapa, codificação, foi escrito o código-fonte, implementado todos os modelos, requisitos e regras de negócio que foram definidos nas fases anteriores.

Na etapa de testes, foram localizados defeitos dentro da ferramenta. Nesta fase, ocorreu o retorno para a fase anterior e até mesmo para as fases iniciais, para a correção de problemas, passando por novos testes até a ferramenta ficar pronta.

Na última fase a ferramenta ficou pronta para implantação em um ambiente de produção, onde os usuários podem acessar, analisar e identificar se os requisitos definidos no início do ciclo foram atendidos.

1.4 Organização da Monografia

O presente trabalho encontra-se organizado em 8 capítulos, descritos abaixo:

O capítulo 1 introduz o projeto através de sua contextualização, apresentação do problema e proposta de solução. Mostra o objetivo geral, específicos e metodologia adotada no desenvolvimento do trabalho.

O capítulo 2 trata sobre os temas necessários para o desenvolvimento do trabalho. São apresentados o conceito de acessibilidade e Libras, as legislações relacionadas à comunidade surda, algumas ferramentas de tecnologia assistiva e um modelo de diagrama da UML.

O capítulo 3 descreve cada tecnologia utilizada para desenvolver a ferramenta UMLibras Web.

O capítulo 4 expõe os pontos positivo e negativo do trabalho referente à primeira versão da ferramenta UMLibras e uma tabela comparativa entre as ferramentas UML disponíveis no mercado.

O capítulo 5 discorre sobre a construção da ferramenta UMLibras, mostrando seus requisitos, estrutura, componentes, diagramas e funcionalidades.

O capítulo 6 descreve as considerações finais, onde são apresentados os resultados obtidos no projeto.

O capítulo 7 apresenta todas as referências bibliográficas utilizadas no trabalho.

Por fim, o capítulo 8 exibe o apêndice A, com os casos de uso complementares do capítulo 5.

2 FUNDAMENTAÇÃO TEÓRICA

A tecnologia pode proporcionar às pessoas com as mais variadas deficiências, maior independência e aquisição de competências que as tornem aptas a adquirir conhecimento como os demais alunos, fazendo necessário buscar constante inovação nesse aspecto educacional, podendo diminuir a exclusão e mostrar ao mundo que não são apenas padrões físicos que devem ser levados em conta, mais sim éticos, morais e intelectuais, considerando a tecnologia na educação inclusiva, um apoio ao professor nessa modalidade de educação (FREITAS; FERREIRA, 2018).

2.1 Acessibilidade

Conforme o Art. 3, inciso I da Lei nº 13.146, de 6 de julho de 2015:

Acessibilidade: possibilidade e condição de alcance para utilização, com segurança e autonomia, de espaços, mobiliários, equipamentos urbanos, edificações, transportes, informação e comunicação, inclusive seus sistemas e tecnologias [...] (BRASIL, 2015).

A acessibilidade para surdos é um grande desafio, principalmente na área da educação. Pensar a Educação, nos espaços escolares ou não escolares, como uma instância fundamental para o processo de desenvolvimento social dos indivíduos demonstra a grande necessidade de ressignificar os sistemas e as políticas educacionais para que o atendimento aos alunos com deficiência se constitua de modo qualitativo (ALMEIDA, 2015).

É necessário tornar computadores e dispositivos móveis acessíveis, facilitando o acesso ao conteúdo dessas tecnologias e reduzindo barreiras de comunicação e acesso à informação.

2.2 Libras

De acordo com a Lei nº 10.436, de 24 de abril de 2002, em seu parágrafo único:

Entende-se como Língua Brasileira de Sinais - Libras a forma de comunicação e expressão, em que o sistema linguístico de natureza visual-motora, com estrutura gramatical própria, constituem um sistema linguístico de transmissão de ideias e fatos, oriundos de comunidades de pessoas surdas do Brasil (BRASIL, 2002).

A Língua Brasileira de Sinais é fundamental para a comunicação da comunidade surda. Através dessa língua é possível reduzir as barreiras de comunicação entre os surdos e os ouvintes, ampliando a autonomia e promovendo a acessibilidade.

No Brasil, após o reconhecimento da Libras como língua natural oriunda da comunidade surda por meio da Lei n.º 10.436, de 24 de abril de 2002, e regulamentada por meio do Decreto n.º 5.626, de 22 de dezembro de 2005, percebe-se os caminhos trilhados para efetivação de uma educação que respeite as singularidades da comunidade surda (ALMEIDA, 2015).

Libras possui gramática, sintaxe, morfologia etc. E para sinalizar uma mensagem é necessário seguir 5 parâmetros básicos: configuração de mãos, pontos de articulação (indicam onde você está sinalizando, sobre a testa ou na frente do corpo), orientação (indica a direção tomada pela mão na realização de determinado sinal), movimento (se para executar o sinal é necessário movimento ou é estático) e expressão do rosto e do corpo.

2.3 Tecnologia UML

Modelagem de sistema é o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva diferente do sistema. A modelagem de sistema geralmente representa o sistema com algum tipo de notação gráfica, que, atualmente, quase

sempre é baseada em notações de UML (linguagem de modelagem unificada, do inglês *Unified Modeling Language*) (SOMMERVILLE, 2011).

A UML surgiu da união de três métodos de modelagem: o método de Booch, o método OMT (Object Modeling Technique) de Jacobson, e o método OOSE (Object-Oriented Software Engineering) de Rumbaugh. Estes eram, até meados da década de 1990, os métodos de modelagem orientada a objetos mais populares entre os profissionais da área de desenvolvimento de software (GUEDES, 2011).

Para que o entendimento de um modelo se faça de forma inequívoca, é necessário adotar para sua especificação uma linguagem cujos símbolos tenham semânticas (significados) e regras de combinação definidos precisamente. E ainda, para que o modelo seja suficientemente completo, a linguagem deve ser expressiva, de forma a descrever concisamente todos os aspectos necessários da realidade modelada (PEREIRA, 2011).

Dentre os diversos modelos, a UML dispõe de um modelo que representa a interação entre os objetos de um sistema computacional, denominado diagrama de sequência, que permite mostrar o relacionamento entre os objetos para a realização de suas funções.

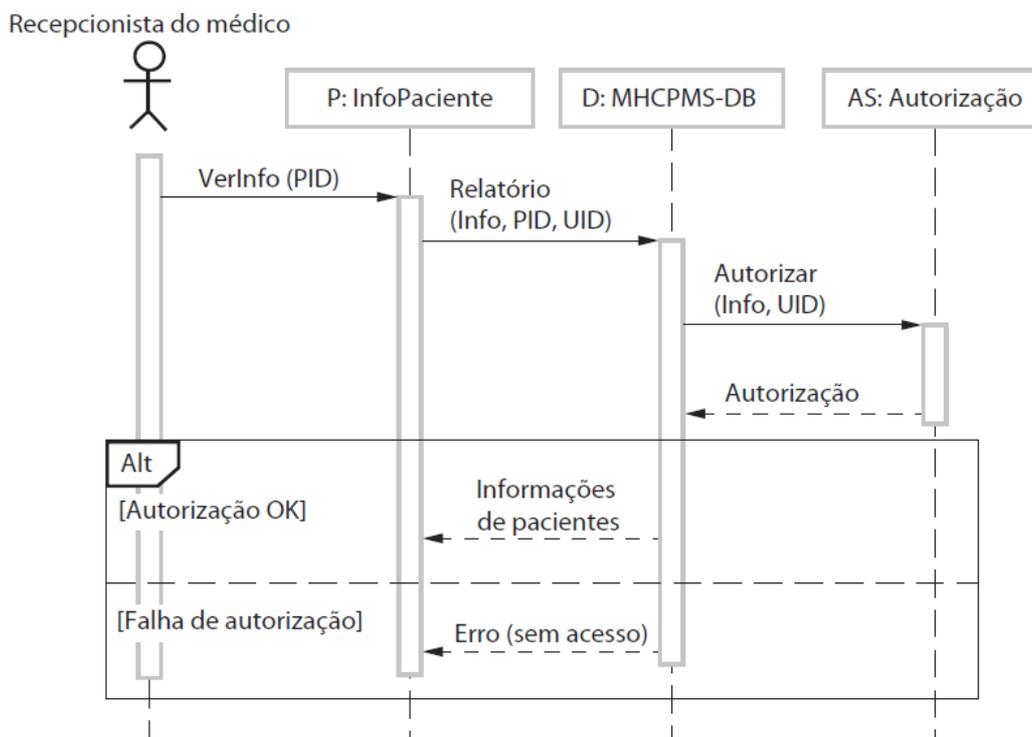
2.4 Diagrama de sequência

Os diagramas de sequência (DSs) permitem especificar colaborações, descrevendo a sequência de mensagens passadas de objeto a objeto necessária para a realização de um determinado procedimento, seja ele um processo de negócio ou uma funcionalidade em um sistema. Essa colaboração segue uma visão temporal, constituída de duas dimensões: a dimensão horizontal, representada pelos objetos que irão interagir e a dimensão vertical que representa o tempo (PEREIRA, 2011).

A Figura 2 ilustra um exemplo de diagrama de sequência que demonstra seus conceitos básicos. Esse diagrama modela as interações envolvidas em um caso de uso 'Ver informações de pacientes', em que a recepcionista do médico pode ver algumas informações sobre o paciente. O diagrama mostra um grupo

de objetos representados por linhas de vida e as mensagens que eles trocam entre si, descrevendo o fluxo de um processo.

Figura 2 - Diagrama de Sequência para ver informações de pacientes.



Fonte: (SOMMERVILLE, 2011).

Os objetos e atores envolvidos estão listados na parte superior do diagrama, com uma linha tracejada verticalmente a partir deles. Interações entre objetos são indicadas por setas anotadas. O retângulo na linha tracejada indica a linha da vida do objeto em questão (ou seja, o tempo em que a instância do objeto está envolvida no processamento). Deve-se ler a sequência de interações de cima para baixo. As anotações sobre as setas indicam as chamadas para os objetos, seus parâmetros e os valores de retorno (SOMMERVILLE, 2011).

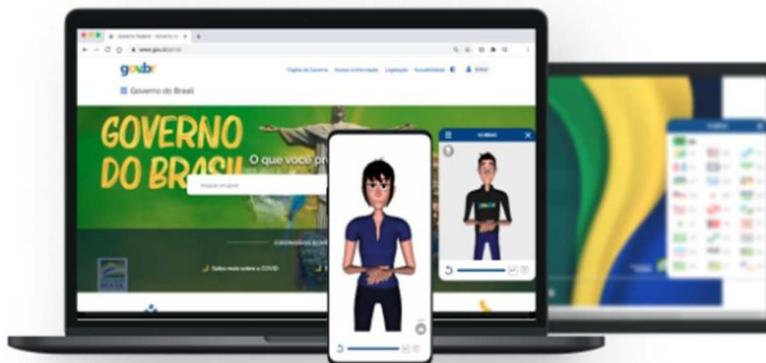
2.5 Tecnologias assistivas

Tecnologia Assistiva é o termo usado para identificar todo o arsenal de Recursos e Serviços que contribuem para proporcionar ou ampliar habilidades funcionais de pessoas com deficiência e consequentemente promover Vida Independente e Inclusão (ASSISTIVA, 2022). Nesse contexto, a seguir são listadas algumas ferramentas dentro do escopo de Tecnologia Assistiva.

2.5.1 VLibras

Um exemplo de tecnologia assistiva é a suíte VLibras, Figura 3, que é um aplicativo que traduz da língua Portuguesa para Libras desenvolvida para diversas plataformas como celular e computador.

Figura 3 - Dispositivos com VLibras.



Fonte: (MDN, 2022)

2.5.2 Hand Talk

Esse aplicativo transforma as imagens e textos em linguagens de sinais. Os idealizadores criaram um personagem chamado Hugo, Figura 4, que funciona como uma interface que traduz textos em Português para Libras. O software converte as mensagens SMS e se o surdo fotografar imagens com legendas também vai poder obter a tradução instantaneamente.

Figura 4 - Personagem Hugo.



Fonte: (HandTalk, 2022)

2.5.3 Transcrição Instantânea

Este aplicativo do Google transcreve áudios em mais de 70 idiomas. O objetivo do aplicativo é tornar as conversas cotidianas e os sons próximos mais

acessíveis para pessoas surdas ou com deficiência auditiva. A ferramenta também torna capaz a captação de sons como “porta batendo”, “pessoa espirrando”, entre outros.

2.5.4 Prodeaf

É um aplicativo capaz de traduzir textos e voz para Libras. Seu principal objetivo é a comunicação entre as pessoas com necessidades especiais. A pessoa pronuncia uma frase ou digita uma e um avatar realiza a tradução para a língua de sinais. O aplicativo é totalmente gratuito, e pode ser utilizado nos aparelhos com Android.

2.5.5 WebLibras

É um tradutor gratuito da ProDeaf que usa um personagem 3D para apresentar as interpretações em Libras em sites na web. O WebLibras permite que as traduções sejam revisadas manualmente, garantindo 100% de equivalência semântica, e permite aos surdos plena compreensão do conteúdo em sua língua primária.

2.5.6 Rybená

Funciona como um simples tradutor, quando a pessoa coloca o texto que deseja e este é convertido em LIBRAS, aparecendo uma animação conforme Figura 5 e traduz o texto inserido na caixa de mensagem.

Figura 5 - Tradutor virtual Rybená.



Fonte: (ICTS 2022)

2.5.7 & Ava

AVA é um aplicativo que realiza a transcrição de áudio para textos. A ideia é possibilitar que pessoas surdas ou com deficiência auditiva possam participar das conversas de pessoas ouvintes, acompanhando as falas de cada um por meio dos textos sem precisar fazer leitura labial.

O AVA também permite que as pessoas se cadastrem e que a voz de cada uma seja reconhecida nas conversas em grupo, que são transcritas para o aplicativo que funciona como um chat. A logomarca é mostrada na Figura 6.

Figura 6 - Logo da Ava.



Fonte: (AVA, 2022)

2.5.8 Primeira Mão

Produzido pela TV INES, do Instituto Nacional de Educação de Surdos, é outro aplicativo inovador que consiste em transmitir informações via um jornal semanal bilíngue em Português e Libras, mostrado na Figura 7, compartilhando as principais notícias do Brasil e do mundo, onde deste modo, garante que a informação chegará até o público surdo. Disponível nas lojas de aplicativo Android e IOS. Basta buscar pelo nome “Primeira Mão”.

Figura 7 - Jornal Primeira Mão.



Fonte: (INES, 2022)

3 TECNOLOGIAS UTILIZADAS

Neste capítulo serão apresentadas as tecnologias que foram utilizadas nas fases de desenvolvimento da ferramenta UMLibras Web.

3.1 Servidor Apache

O Apache HTTP Server Project é um esforço colaborativo de desenvolvimento de *software* destinado a criar uma implementação de código-fonte robusto, de nível comercial, com recursos e disponível gratuitamente. O projeto é gerenciado em conjunto por um grupo de voluntários localizados em todo o mundo, usando a Internet e a Web para comunicar, planejar e desenvolver o servidor e sua documentação relacionada. Este projeto faz parte da Apache Software Foundation. (APACHE, 2022)

Centenas de usuários contribuíram com ideias, código e documentação para o projeto.

A Licença Apache é uma licença de *software* livre de autoria da Apache Software Foundation (ASF).

3.2 MySQL

O *software* MySQL oferece um servidor de banco de dados SQL (Structured Query Language) muito rápido, *multithread*, multiusuário e robusto.

O MySQL Server destina-se a sistemas de produção de carga pesada e de missão crítica, bem como para incorporação em *software* implantado em massa (MYSQL, 2022). Além disso, possui licença General Public License (GPL) e Licença comercial.

3.3 HTML

HyperText Markup Language (Linguagem de Marcação de Hipertexto) é uma linguagem de marcação que define a estrutura do seu conteúdo.

O HTML consiste em uma série de elementos, que são usados para incluir ou envolver diferentes partes do conteúdo de uma página para fazê-lo aparecer ou agir de uma determinada maneira. As *tags* anexas podem criar um *hiperlink* de uma palavra ou imagem para outro lugar, bem como podem colocar palavras em itálico, podem aumentar ou diminuir a fonte e assim por diante.

3.4 CSS

Cascading Style Sheets ou Folhas de Estilo em Cascata é uma linguagem de estilo usada para descrever a apresentação de um documento escrito em HTML ou em XML (incluindo várias linguagens em XML como SVG, MathML ou XHTML). O CSS descreve como elementos são mostrados na tela, no papel, na fala ou em outras mídias (MDN, 2022).

3.5 Javascript

É uma linguagem de programação leve, interpretada ou compilada *just-in-time* com funções de primeira classe, baseada em protótipo, multiparadigma, *single-threaded*, dinâmica, suportando estilos orientados a objetos, imperativos e declarativos (por exemplo, programação funcional) (MDN, 2022).

Conhecida como a linguagem de *script* para páginas da Web, permite que muitos ambientes não navegadores também a utilizem, como Node.js, Apache CouchDB e Adobe Acrobat.

3.6 VLibras

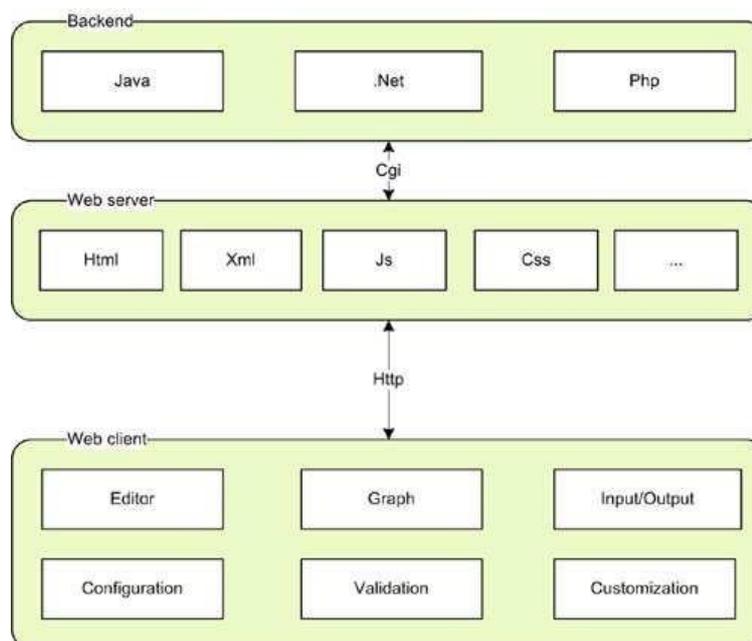
A suíte VLibras é um conjunto de ferramentas gratuitas e de código aberto que traduz conteúdos digitais (texto, áudio e vídeo) de Português para Libras, tornando computadores, celulares e plataformas Web mais acessíveis para as pessoas surdas (BRASIL, 2022).

3.7 MxGraph

O mxGraph é uma biblioteca de diagramação JavaScript que fornece todas as funcionalidades necessárias para desenhar e associar um contexto a um diagrama, permitindo a criação de modelos gráficos que são executados nativamente em qualquer navegador suportado por seu fornecedor (MXGRAPH, 2020).

Essa biblioteca compreende principalmente um arquivo JavaScript que contém toda a funcionalidade do mxGraph. Ele é carregado em uma página Web em uma seção JavaScript e é executado em um contêiner HTML no navegador, conforme Figura 8. Por ser uma arquitetura simples, requer somente um servidor web capaz de servir páginas HTML e um navegador Web habilitado para JavaScript.

Figura 8 - Arquitetura da biblioteca mxGraph.



Fonte: (MXGRAPH, 2020).

As principais vantagens desta tecnologia são:

- Nenhum *plug-in* de terceiros é necessário para o seu funcionamento;
- As tecnologias envolvidas são abertas e padronizadas;

4 TRABALHOS RELACIONADOS

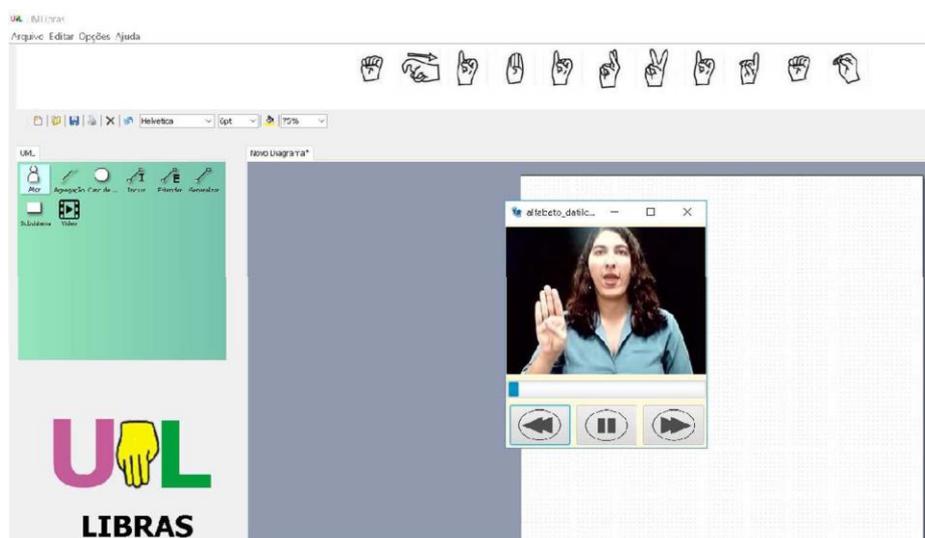
Neste capítulo será apresentado o trabalho referente à primeira versão da ferramenta UMLibras, sob orientação do professor M.Sc. Sergio Augusto C. Bezerra, na qual foi desenvolvido pelo aluno Antonio José Barbosa Fonseca do curso de Tecnologia em Análise e Desenvolvimento de Sistemas – TADS, ambos pertencentes ao Instituto Federal de Educação, Ciência e Tecnologia do Amazonas - IFAM.

4.1 UMLibras versão desktop

Desenvolvida para a plataforma desktop e utilizando a linguagem de programação Java, a primeira versão da ferramenta UMLibras permite a criação de diagramas de caso de uso da UML.

A ferramenta, mostrada na Figura 9, utiliza um algoritmo para converter palavras em um grupo de imagens realizando a datilologia em Libras, proporcionando a pessoas com deficiência auditiva acessibilidade e interatividade na utilização para o desenvolvimento do referido diagrama. Além disso, também permite associar vídeos aos objetos, sejam eles em Libras ou não, bem como sua execução.

Figura 9 - UMLibras desktop.



Fonte: (FONSECA, 2018)

A única limitação da ferramenta é permitir somente a criação do diagrama de caso de uso, porém, por se tratar de um projeto inicial, ainda prevê a criação de módulos adicionais que contemplem os demais diagramas da UML.

4.2 Ferramentas semelhantes

A Tabela 1 traz algumas características das principais ferramentas de modelagem disponíveis no mercado, fazendo uma comparação com a ferramenta UMLibras web e desktop, onde todas diferem destas por não oferecerem acessibilidade.

A primeira coluna apresenta o nome das ferramentas, a coluna seguinte quais ferramentas desenvolvem o diagrama de sequência, na coluna três suas licenças de uso e na última coluna se atendem ou não ao requisito de acessibilidade.

Tabela 1 – Ferramentas UML disponíveis no mercado.

Nome	Diagrama de Seqência	Tipo de Licença	Oferece acessibilidade?
Astah	Sim	<i>Paga/Community</i>	Não
UML Diagram Creator	Sim	<i>Paga/Free</i>	Não
Umbrello	Sim	<i>OpenSource</i>	Não
Argo UML	Sim	<i>OpenSource</i>	Não
Modelio	Sim	<i>OpenSource</i>	Não
Yuml	Sim	<i>Paga/Free</i>	Não
Dia	Sim	GPL	Não
Poseidon	Sim	<i>Paga/Community</i>	Não
Diagrams	Sim	<i>Free</i>	Não
UMLibras desktop	Não	<i>Free</i>	Sim
UMLibras web	Sim	<i>Free</i>	Sim

Fonte: (Próprio autor, 2022).

5 FERRAMENTA DESENVOLVIDA

Neste capítulo serão apresentados os requisitos funcionais, não funcionais, arquitetura, componentes, diagrama de caso de uso, diagrama de classes e funcionalidades da ferramenta UMLibras.

Na construção da ferramenta foi utilizado como base a estrutura de ciclo de vida de desenvolvimento denominada modelo em cascata, onde a evolução das atividades segue uma sequência lógica de etapas ao longo do projeto.

Uma das primeiras fases de um processo de desenvolvimento de software consiste no Levantamento de Requisitos. As outras etapas, sugeridas por muitos autores, são: Análise de Requisitos, Projeto, que se constitui na principal fase da modelagem, Codificação, Testes e Implantação (GUEDES, 2011).

5.1 Levantamento e Análise de requisitos

No modelo em cascata, os requisitos do sistema são coletados no início do projeto e, em seguida, são criadas fases sequenciais para implementar esses requisitos.

Nessa etapa foi apresentado o problema e através do levantamento de requisitos foram definidas as necessidades dos usuários da ferramenta. Para coletar os requisitos, foi realizada pesquisa em livros, entrevistas com alguns usuários e comparação com ferramentas semelhantes no mercado.

A seguir serão descritos os requisitos funcionais e não-funcionais pesquisados, respectivamente, listados na Tabela 2 e Tabela 3.

5.1.1 Requisitos funcionais

Os requisitos funcionais descrevem os comportamentos do produto. Exemplos incluem ações, processos, dados e interações que o produto deve executar (PMI, 2017).

Tabela 2 - Lista dos requisitos funcionais da ferramenta.

Requisito	Descrição
RF01	A ferramenta deve permitir criar, salvar, abrir e exportar diagramas;
RF02	A ferramenta deve exibir links para vídeos ou gifs, ilustrações e notificações em Libras;
RF03	A ferramenta deve permitir manter usuários.
RF04	A ferramenta deve permitir abrir o intérprete virtual VLibras.
RF07	A ferramenta deve exibir vídeos introdutórios em libras.

Fonte: (Próprio autor, 2022).

5.1.2 Requisitos não-funcionais

Os requisitos não-funcionais complementam os requisitos funcionais e descrevem as condições ou qualidades ambientais requeridas para que o produto seja eficaz. Exemplos incluem: confiabilidade, proteção, desempenho, segurança, nível de serviço, suportabilidade, retenção/descarte, etc (PMI, 2017).

Tabela 3 - Lista dos requisitos não-funcionais da ferramenta.

Requisito	Descrição
RNF01	O sistema deve estar sempre disponível, caso ocorra alguma interrupção ele deverá ser restaurado o mais rápido possível;
RNF02	O sistema deve ser fácil de usar, além de possuir uma interface simples, intuitiva, moderna e amigável;
RNF03	Os tempos de resposta não poderão ultrapassar 5 segundos.

Fonte: (Próprio autor, 2022).

5.2 Projeto

Nesta etapa foi planejado as especificações e definido a estrutura da ferramenta desenvolvida, mostrando a articulação lógica entre seus componentes, detalhes de procedimentos e caracterização da interface.

5.2.1 Estrutura básica

Seguindo os padrões Web, a ferramenta possui duas estruturas básicas: *front-end* que é a parte visual da ferramenta e *back-end* que fica por trás da aplicação.

No *front-end* foram utilizadas as linguagens de programação HTML, CSS, e JavaScript para fazer as interações com o usuário e receber as informações necessárias ao funcionamento da ferramenta.

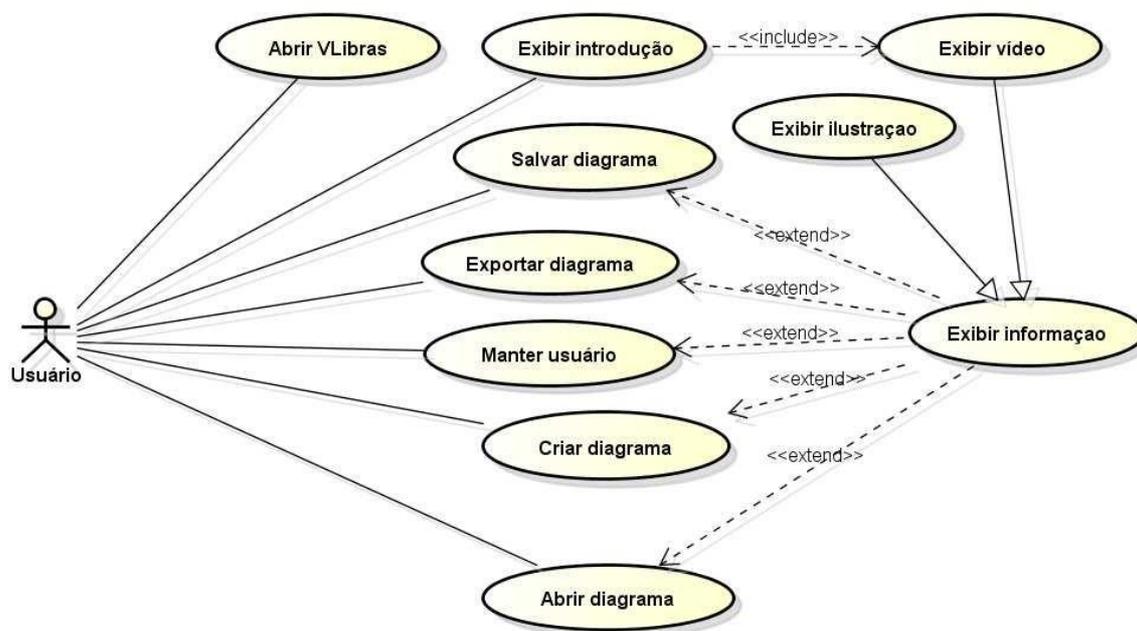
No *back-end* foi utilizado o Sistema de Gerenciamento de Banco de Dados (SGBD) MySQL e o servidor Web Apache, onde processa as informações recebidas e então devolve para o navegador os resultados dos processamentos.

Foi utilizado como base para o desenvolvimento da biblioteca JavaScript mxGraph, onde foram aplicadas as adaptações de acessibilidade e a integração com a Application Programming Interface (API) VLibras.

5.2.2 Diagrama de caso de uso

O diagrama de caso de uso demonstra as interações entre os usuários e as funcionalidades de um sistema. A Figura 10 ilustra o diagrama de caso de uso da ferramenta desenvolvida, onde é possível identificar o usuário do sistema representado por um boneco palito. Este usuário tem as opções de interação com o sistema representadas por balões onde, entre várias ações, ele pode criar um diagrama, salvar o diagrama criado, exibir vídeos com informações em libras e ativar o tradutor virtual VLibras.

Figura 10 - Diagrama de caso de uso.



Fonte: (Próprio autor, 2022).

5.2.3 Descrição dos casos de uso

As Tabelas 4, 5 e 6 descrevem de forma detalhada alguns casos de uso, os demais estão listados no Apêndice A.

Tabela 4 - Caso de uso: criar diagrama.

Item	Descrição
UC01	Permite ao usuário criar uma nova aba para um novo diagrama.
Ator	Usuário.
Pré-condição	Não existe.
Fluxo principal	P1 – o usuário clica no botão novo da barra de ferramentas. P2 – o sistema abre uma nova aba com a área de trabalho vazia para o usuário criar um novo diagrama.
Fluxo alternativo	P1 – o usuário clica no menu arquivo, opção novo. P2 – o sistema abre uma nova aba com a área de trabalho vazia para o usuário criar um novo diagrama.

Fonte: (Próprio autor, 2022).

Tabela 5 - Caso de uso: abrir VLibras.

Item	Descrição
UC02	Permite ao usuário abrir o intérprete virtual VLibras.
Ator	Usuário.
Pré-condição	Não existe.
Fluxo principal	<p>P1 – o usuário clica no botão suspenso com o ícone de libras.</p> <p>P2 – o sistema abre uma janela no canto inferior direito e carrega o intérprete virtual VLibras.</p> <p>P3 – o usuário deverá clicar em cima do texto que deseja a tradução pelo intérprete virtual.</p> <p>P4 – o intérprete virtual executa a tradução e aguarda novo click do usuário.</p> <p>P5 – o usuário pode clicar em outro texto ou fechar o intérprete clicando no X do canto superior direito da janela do intérprete virtual.</p>
Fluxo alternativo	Não existe.

Fonte: (Próprio autor, 2022).

Tabela 6 - Caso de uso: exibir informações.

Item	Descrição
UC03	Permite ao usuário visualizar um vídeo com informações em libras sobre determinada funcionalidade do sistema.
Ator	Usuário.
Pré-condição	Clicar no ícone indicando vídeo de ajuda.
Fluxo principal	<p>P1 – antes de usar uma funcionalidade o usuário poderá clicar no ícone de ajuda da funcionalidade para assistir um vídeo com informações em libras sobre a utilização da funcionalidade do sistema.</p> <p>P2 – o sistema exibe uma janela mostrando o vídeo com a informação de ajuda em libras.</p> <p>P3 – após o término do vídeo, o sistema permite fechar ou repetir o vídeo.</p>
Fluxo alternativo	Não existe.

Fonte: (Próprio autor, 2022).

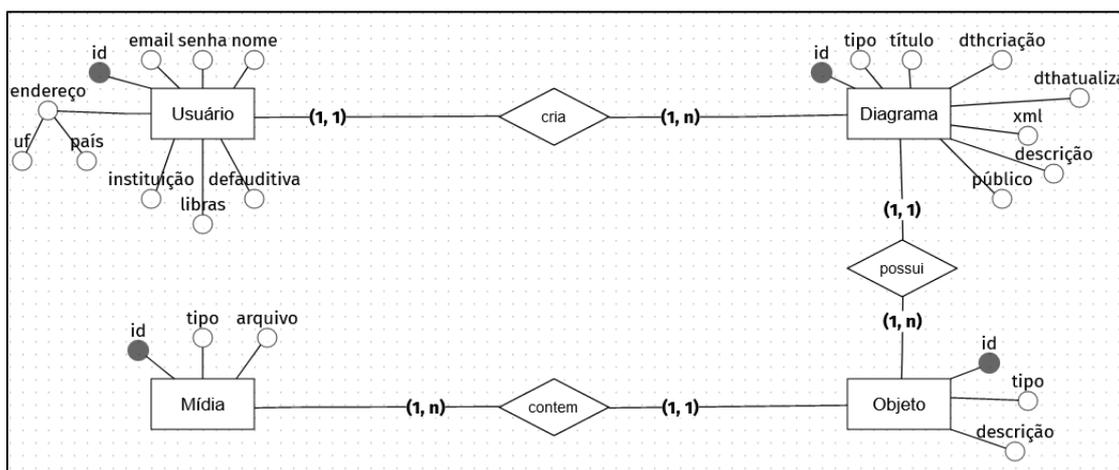
5.2.4 Modelo conceitual do banco de dados

Um modelo conceitual é uma descrição do banco de dados de forma independente de implementação em um sistema gerenciador de banco de dados - SGBD. O modelo conceitual registra que dados podem aparecer no banco de dados, mas não registra como estes dados estão armazenados a nível de SGBD. A técnica mais difundida de modelagem conceitual é a abordagem Entidade-Relacionamento (ER) (HEUSER, 2009), que se constitui em um diagrama que permite representar de forma gráfica os elementos que foram definidos para o Banco de Dados - BD.

Aplicado esta técnica na construção do banco de dados relacional, então foi possível descrever de forma simples, todos os atributos importantes necessários à construção do módulo da ferramenta, bem como suas entidades e seus relacionamentos.

A Figura 11 apresenta o modelo entidade-relacionamento do banco de dados do módulo da ferramenta UMLibras voltada à construção de diagramas de sequência.

Figura 11 - Modelo entidade-relacionamento da ferramenta UMLibras.



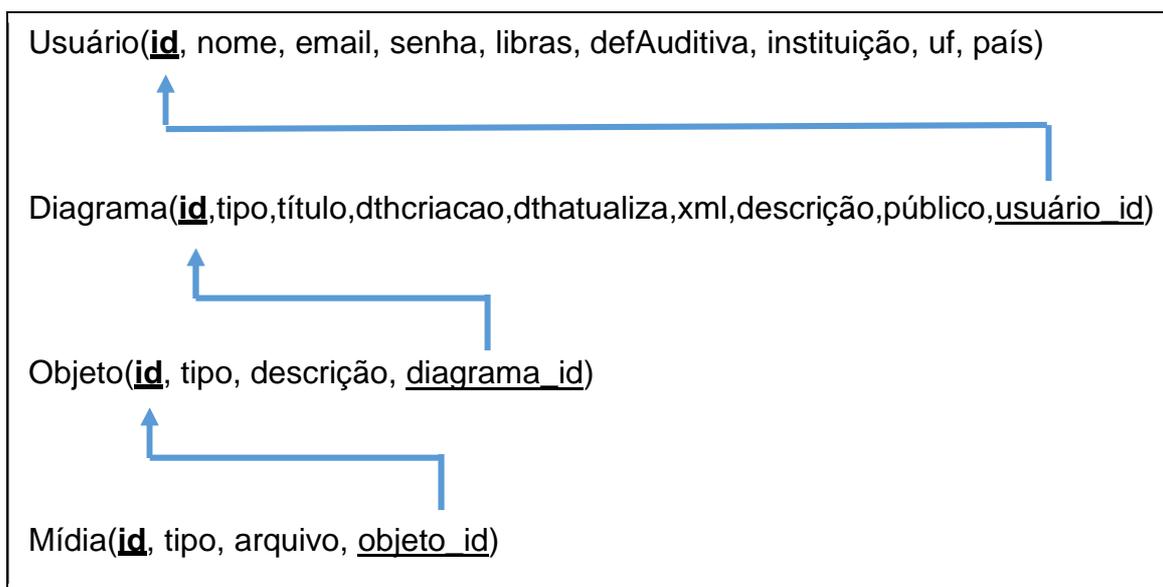
Fonte: (Próprio autor, 2022).

5.2.5 Modelo lógico do banco de dados

Um modelo lógico é uma descrição de um banco de dados no nível de abstração visto pelo usuário do SGBD (HEUSER, 2009).

As Figuras 12 e 13 apresentam o modelo lógico da ferramenta UMLibras, respectivamente, de forma textual, e em forma de tabelas.

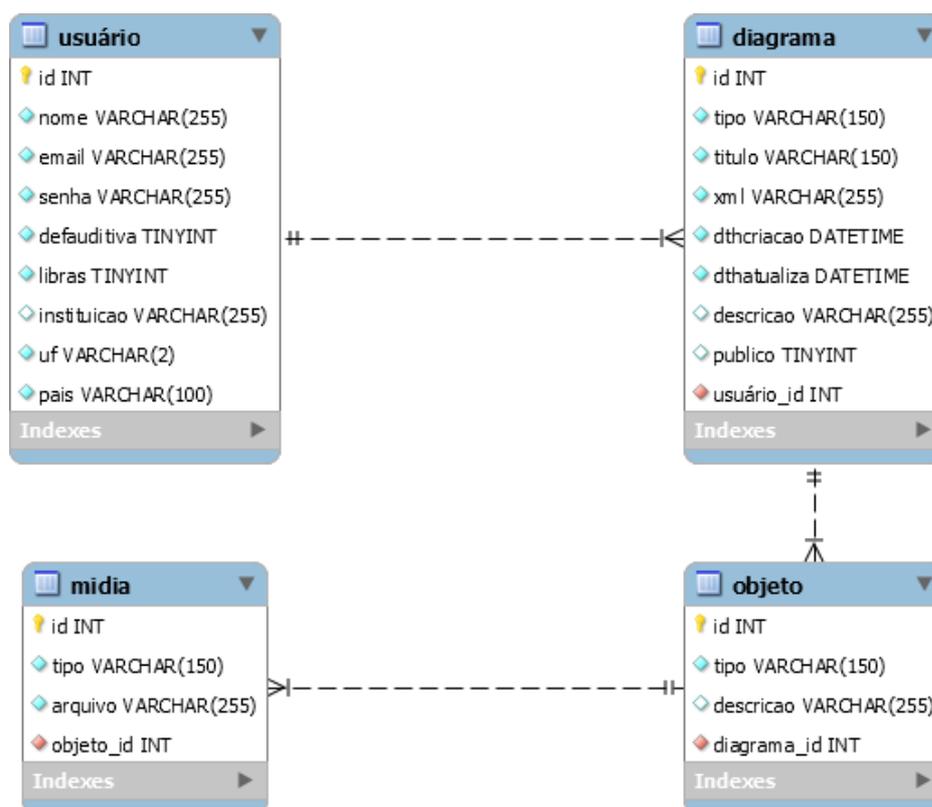
Figura 12 - Modelo lógico (forma textual) da ferramenta UMLibras.



Fonte: (Próprio autor, 2022).

Tanto a Figura 12 quanto a Figura 13 implementam os requisitos identificados na modelagem conceitual mostrada na Figura 11. Com base nesses modelos, foi definindo quais as tabelas o banco de dados iria conter e quais os nomes das colunas.

Figura 13 - Diagrama Lógico da ferramenta UMLibras.



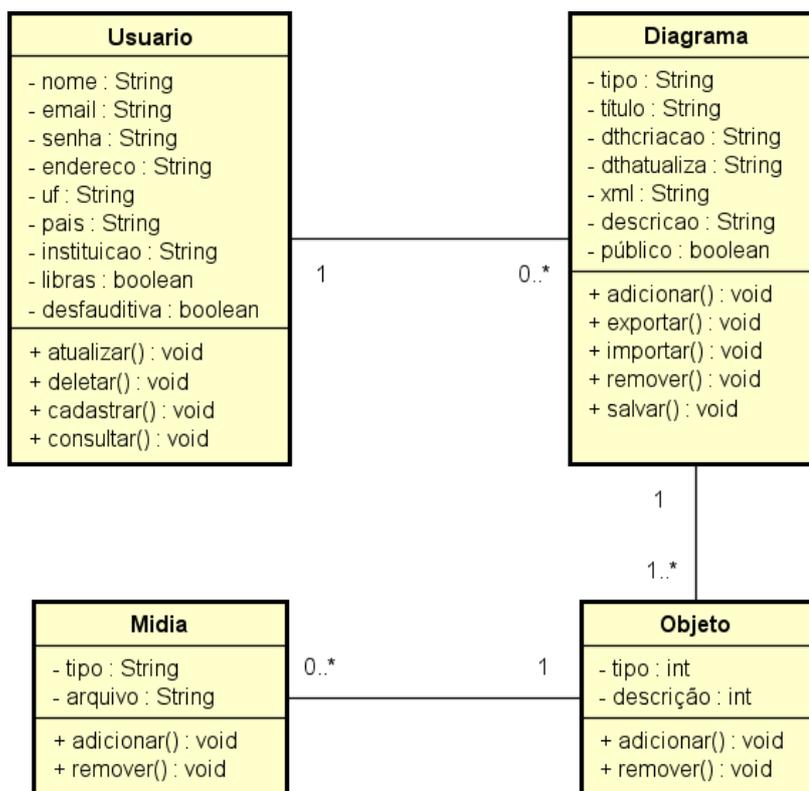
Fonte: (Próprio autor, 2022).

5.2.6 Diagrama de classes

Os diagramas de classe são usados no desenvolvimento de um modelo de sistema orientado a objetos para mostrar as classes de um sistema e as associações entre essas classes. Em poucas palavras, uma classe de objeto pode ser pensada como uma definição geral de um tipo de objeto do sistema. Uma associação é um *link* entre classes que indica algum relacionamento entre essas classes (SOMMERVILLE, 2011).

A classe usuário, além de seus atributos, contém os quatro métodos básicos, CRUD (Create, Read, Update e Delete), necessários para manter um usuário. O atributo xml é o mais importante da classe diagrama, pois contém todas as configurações e propriedades para formar um modelo.

Figura 14 - Diagrama de classes.



Fonte: (Próprio autor, 2022).

5.2.7 Interface

Essa seção demonstra os protótipos das telas da ferramenta UMLibras, onde foram desenhados utilizando a ferramenta de design de interface de usuário Balsamiq, disponível em balsamiq.com.

As Figuras de 15 a 18 especificam as telas de Login, Cadastro de usuário, Principal e Gerenciar mídias.

Figura 15 - Protótipo da tela de login.



UMLibras

https://umlibras.com

UMLibras

Cadastro

Nome

Email

Senha

Deficiência auditiva? Sim Não

Entende Libras? Sim Não

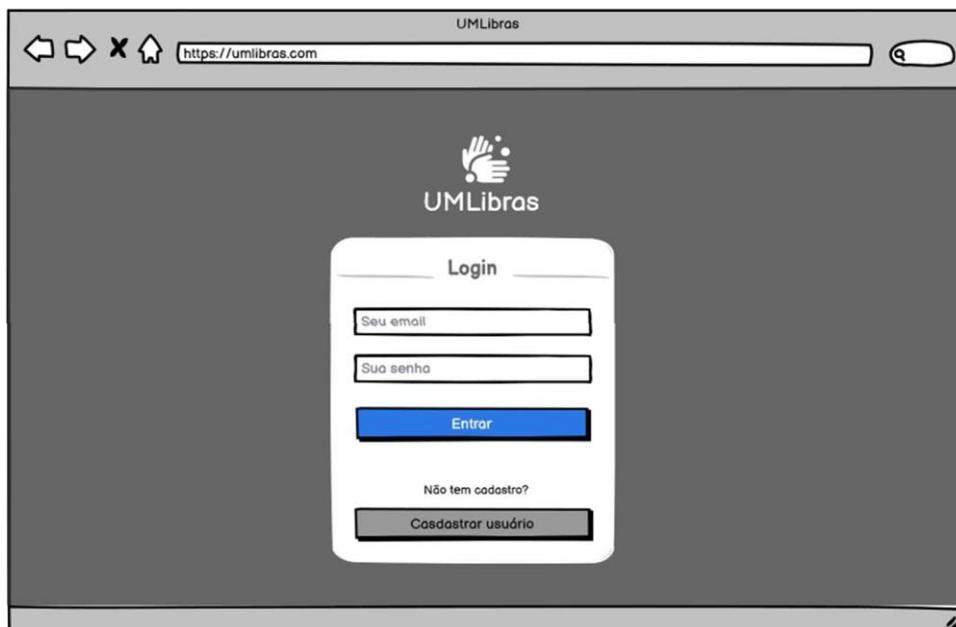
Intituição

UF País

Cadastrar

Fonte: (Próprio autor, 2022).

Figura 16 - Protótipo da tela de cadastro de usuário.



UMLibras

https://umlibras.com

UMLibras

Login

Seu email

Sua senha

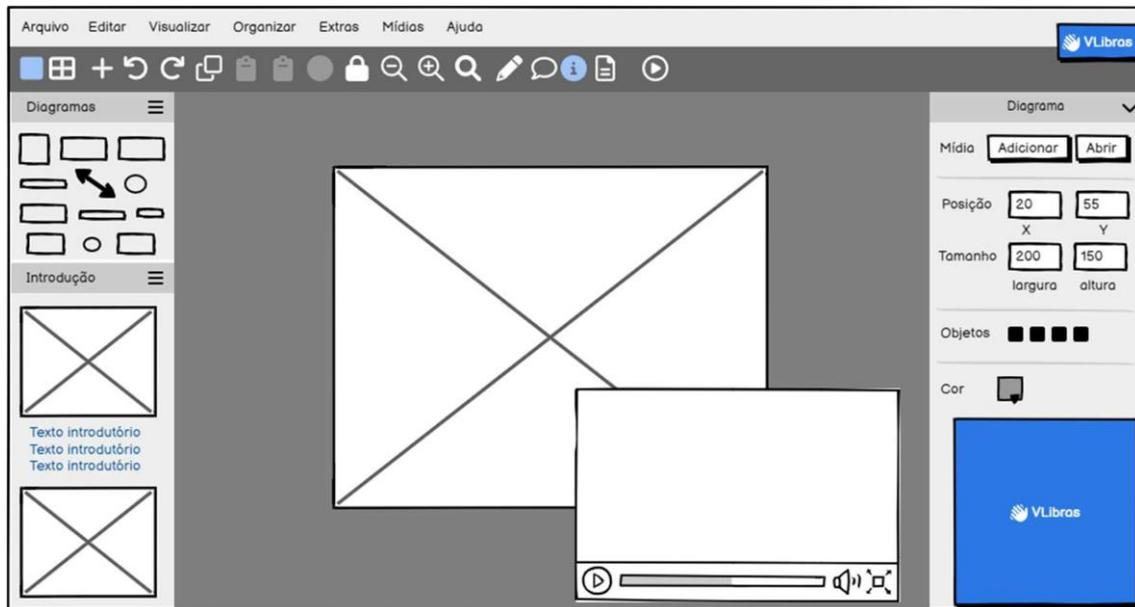
Entrar

Não tem cadastro?

Cadastrar usuário

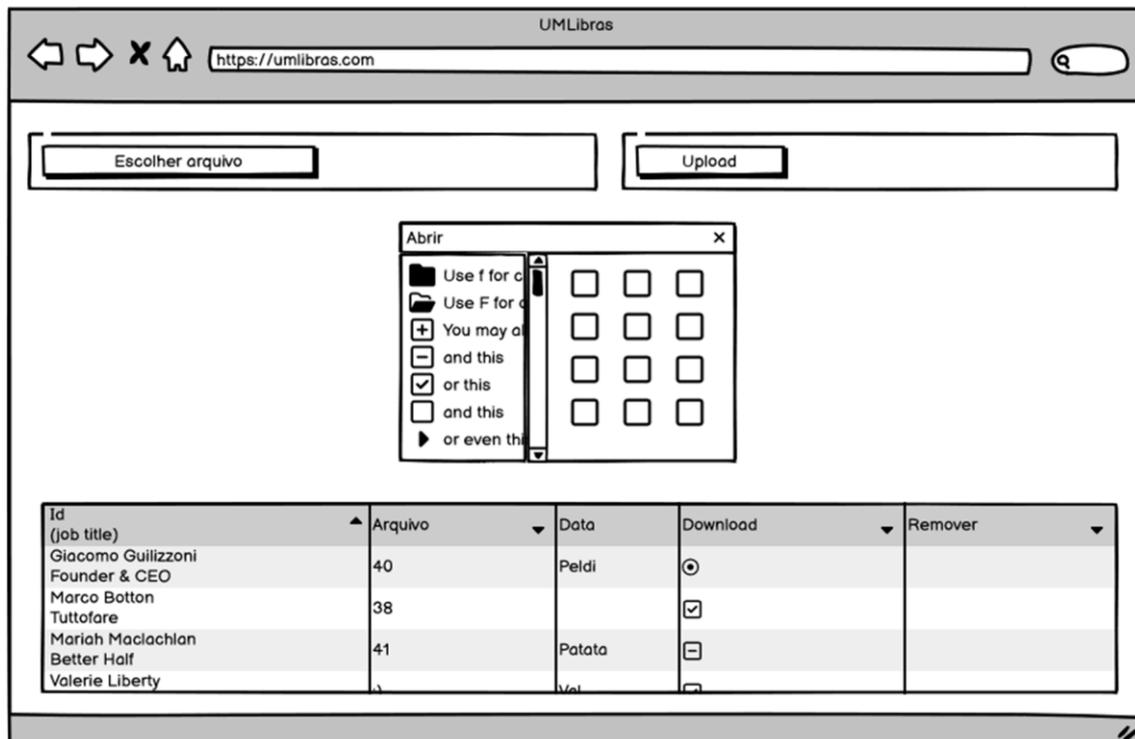
Fonte: (Próprio autor, 2022).

Figura 17 - Protótipo da tela principal da ferramenta UMLibras.



Fonte: (Próprio autor, 2022).

Figura 18 - Protótipo da tela de gerenciamento de mídias.



Fonte: (Próprio autor, 2022).

5.3 Codificação

A tarefa de implementação inclui todas as atividades de desenvolvimento do sistema propriamente dito, ou seja, que estão relacionadas com a concretização do modelo de desenho produzido na tarefa anterior. Os diversos componentes aplicativos são codificados e testados de forma isolada, garantindo assim a respectiva correção interna (SILVA; VIDEIRA, 2001).

5.3.1 Biblioteca mxGraph

A biblioteca é carregada usando o código da Figura 19. A variável `mxBasePath` é usada para definir o caminho de onde a biblioteca carrega seus recursos. Essa variável deve ser definida antes de carregar o código da biblioteca. O arquivo `mxClient.min.js` contém todo o código necessário em um único arquivo minificado.

Figura 19 - Código para carregar a biblioteca mxGraph.

```
1 <script type="text/javascript">
2   mxBasePath = 'javascript/src';
3 </script>
4 <script type="text/javascript" src="javascript/src/js/mxClient.js"></script>
```

Fonte: (MXGRAPH, 2020).

5.3.2 Gerar gráfico

O código da Figura 20, constrói um modelo de gráfico vazio e passa o contêiner e o modelo vazio para o construtor do gráfico.

Figura 20 - Código para a construção de um gráfico.

```
1 var model = new mxGraphModel();
2 var graph = new mxGraph(container, model);
```

Fonte: (MXGRAPH, 2020).

5.3.3 Vértices e arestas

As funções `beginUpdate()` e `endUpdate()` são usadas para inserir vértices e arestas, Figura 21. A `endUpdate()` deve sempre entrar em um bloco final para garantir que seja sempre executado se o `beginUpdate()` for executado. No entanto, `beginUpdate()` não deve fazer parte do bloco `try` para garantir que `endUpdate()` nunca seja executado se `beginUpdate()` falhar. Isso é necessário para que o modelo permaneça em estado consistente, ou seja, para cada chamada para `beginUpdate()` deve haver sempre exatamente uma chamada para `endUpdate()`.

A parte dentro do bloco `try` cria os vértices e arestas do gráfico. O pai padrão é obtido do gráfico e, normalmente, é o primeiro filho da célula raiz no modelo, que é criado automaticamente ao usar o modelo de gráfico sem argumentos.

Figura 21 - Código para inserir vértices e arestas.

```
1  var parent = graph.getDefaultParent();
2
3  model.beginUpdate();
4  try
5  {
6      var v1 = graph.insertVertex(parent, null, '', 20, 20, 80, 30);
7      var v2 = graph.insertVertex(parent, null, '', 200, 150, 80, 30);
8      var e1 = graph.insertEdge(parent, null, '', v1, v2);
9  }
10 finally
11 {
12     model.endUpdate();
13 }
```

Fonte: (MXGRAPH, 2020).

O uso de `beginUpdate()` e `endUpdate()` não apenas melhora o desempenho da exibição, mas também é usado para marcar os limites para alterações que podem ser desfeitas quando desfazer/refazer é usado.

5.3.4 Modelo

A função `mxCell()` define os elementos do modelo gráfico, que é implementado por `mxGraphModel()`.

O modelo gráfico tem as seguintes propriedades:

- O elemento raiz do gráfico contém as camadas.
- O pai de cada camada é o elemento raiz.
- Uma camada pode conter elementos do modelo de gráfico, ou seja, vértices, arestas e grupos.
- Os grupos podem conter elementos do modelo gráfico, recursivamente.

A Figura 22 mostra o código para criar um novo modelo de gráfico com uma célula raiz e uma camada padrão.

Figura 22 - Código para criar um modelo de gráfico.

```
1 var root = new mxCell();
2 root.insert(new mxCell());
3 var model = new mxGraphModel(root);
```

Fonte: (MXGRAPH, 2020).

5.3.5 Folha de estilo

A aparência das células em um gráfico é definida pela folha de estilo, que é uma instância de `mxStylesheet`. Um estilo é uma matriz de pares de chave e valor a serem usados com as células. As chaves são definidas em `mxConstants` e os valores podem ser strings e números, bem como objetos ou funções JavaScript.

A Figura 23 mostra o código para modificar os estilos padrão para vértices e arestas em um gráfico existente.

Figura 23 - Código para modificar estilos.

```
1 var vertexStyle = graph.getStylesheet().getDefaultVertexStyle();
2 vertexStyle[mxConstants.STYLE_ROUNDED] = true;
3
4 var edgeStyle = graph.getStylesheet().getDefaultEdgeStyle();
5 edgeStyle[mxConstants.STYLE_EDGE] = mxEdgeStyle.TopToBottom;
```

Fonte: (MXGRAPH, 2020).

5.3.6 CSS

A folha de estilo CSS contém as definições de estilo para vários elementos da interface do usuário, como menus pop-up. Folhas de estilo adicionais podem ser adicionadas usando `mxClient.link('stylesheet', filename)` ou por meio de uma tag de folha de estilo da seção UI na configuração do editor, Figura 24.

Figura 24 - Código para adicionar folha de estilo.

```
1 <mxEditor>
2 <ui>
3 <stylesheet name="examples/editors/css/process.css"/>
4 ...
```

Fonte: (MXGRAPH, 2020).

5.3.7 Gerar XML do gráfico

O esquema de codificação padrão mapeia todos os campos não-objeto para atributos de string e todos os campos de objeto para nós filhos, usando o nome do construtor do objeto como o nome do nó e o nome do campo para o valor como atributo. Esse esquema de codificação padrão pode ser substituído por codecs personalizados, que são registrados no `mxCodecRegistry`.

Por exemplo, a Figura 25 mostra o código de um objeto JavaScript. Para codificar este objeto e mostrar o XML resultante em uma nova janela, o código da Figura 26 é executado. Resultando no código XML da Figura 27.

Figura 25 - Código de um objeto JavaScript.

```
1 var object = new Object();
2 object.myBool = true;
3 object.myObject = new Object();
4 object.myObject.name = 'Test';
5 object.myArray = ['a', ['b', 'c'], 'd'];
```

Fonte: (MXGRAPH, 2020).

Figura 26 - Código para converter um objeto em XML.

```

1  var encoder = new mxCodec();
2  var node = encoder.encode(object);
3  mxUtils.popup(mxUtils.getXml(node));

```

Fonte: (MXGRAPH, 2020).

Figura 27 - Resultado do objeto convertido em XML.

```

1  <Objeto meuBool="1">
2  <Nome do objeto="Teste" as="meuObjeto"/>
3  <Array as="meuArray">
4  <adicionar valor="a"/>
5  <Array>
6  <adicionar valor="b"/>
7  <adicionar valor="c"/>
8  </Array>
9  <add value="d"/>
10 </Array>
11 </objeto>

```

Fonte: (MXGRAPH, 2020).

5.3.8 Arquivos

As funções salvar, abrir, readGraphModel e writeGraphModel implementam um mecanismo padrão para manipulação de arquivos no mxEditor.

A implementação padrão de mxEditor.save é chamada com um argumento para indicar se o salvamento foi acionado pelo usuário ou pelo sistema. Em seguida, ele usa a variável urlPost do objeto do editor para verificar se uma requisição post deve ser emitida. Se a variável for definida, o editor emitirá uma requisição post para a URL especificada, passando o XML como uma variável POST chamada xml.

A Figura 28 ilustra um exemplo de arquivo PHP localizado no mesmo diretório da página HTML. Se o nome do arquivo for server.php, a variável urlPost deve ser definida como server.php no editor. O arquivo PHP obterá o XML da solicitação POST e o gravará em um arquivo chamado diagram.xml.

Figura 28 - Exemplo de arquivo PHP.

```

1  <?php
2  $xml = $_HTTP_POST_VARS['xml'];
3  if ($xml != null) {
4      $fh=fopen("diagram.xml", "w");
5      fputs($fh, stripslashes($xml));
6      fclose($fh);
7  }
8  ?>

```

Fonte: (MXGRAPH, 2020).

5.3.9 Integração VLibras

A integração do aplicativo VLibras Widget com um website é feita com a inserção de alguns trechos de código antes do fechamento da tag <body> de uma página HTML Figura 29.

Figura 29 - Código para integrar a API Vlibras à um website.

```

1  <body> <!-- Início do corpo da página -->
2
3      ... <!-- Conteúdo da página -->
4
5  <div vw class="enabled">
6      <div vw-access-button class="active"></div>
7      <div vw-plugin-wrapper>
8          <div class="vw-plugin-top-wrapper"></div>
9      </div>
10 </div>
11 <script src="https://vlibras.gov.br/app/vlibras-plugin.js"></script>
12 <script>
13     new window.VLibras.Widget('https://vlibras.gov.br/app');
14 </script>
15 </body> <!-- Fim do corpo da página -->

```

Fonte: (BRASIL, 2022).

5.3.10 Banco de Dados

A Figura 30 ilustra o código de criação das tabelas do banco de dados da ferramenta UMLibras.

Figura 30 - Código para criar as tabelas do banco de dados da UMLibras.

```

1 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
2 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0;
3 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='ONLY_FULL_GROUP_BY,STRICT_TRANS_TABLES,
4 NO_ZERO_IN_DATE,NO_ZERO_DATE,ERROR_FOR_DIVISION_BY_ZERO,NO_ENGINE_SUBSTITUTION';
5 CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8;
6 USE `mydb`;
7
8 CREATE TABLE IF NOT EXISTS `mydb`.`usuário` (`id` INT NOT NULL,
9 `nome` VARCHAR(255) NOT NULL,`email` VARCHAR(255) NOT NULL,
10 `senha` VARCHAR(255) NOT NULL,`defAuditiva` TINYINT NOT NULL,
11 `libras` TINYINT NOT NULL,`instituicao` VARCHAR(255) NULL,
12 `uf` VARCHAR(2) NOT NULL,`pais` VARCHAR(100) NOT NULL,
13 PRIMARY KEY (`id`)) ENGINE = InnoDB;
14
15 CREATE TABLE IF NOT EXISTS `mydb`.`diagrama` (`id` INT NOT NULL,
16 `tipo` VARCHAR(150) NOT NULL,`titulo` VARCHAR(150) NOT NULL,
17 `xml` VARCHAR(255) NOT NULL,`dhcriacao` DATETIME NOT NULL,
18 `dhatualizacao` DATETIME NOT NULL,`descricao` VARCHAR(255) NULL,
19 `publico` TINYINT NULL,`usuário_id` INT NOT NULL,PRIMARY KEY (`id`),
20 INDEX `fk_diagrama_usuario_idx` (`usuário_id` ASC) VISIBLE,
21 CONSTRAINT `fk_diagrama_usuario` FOREIGN KEY (`usuário_id`)
22 REFERENCES `mydb`.`usuário` (`id`) ON DELETE CASCADE
23 ON UPDATE CASCADE) ENGINE = InnoDB;
24
25 CREATE TABLE IF NOT EXISTS `mydb`.`objeto` (`id` INT NOT NULL,
26 `tipo` VARCHAR(150) NOT NULL,`descricao` VARCHAR(255) NULL,
27 `diagrama_id` INT NOT NULL,PRIMARY KEY (`id`),
28 INDEX `fk_objeto_diagrama1_idx` (`diagrama_id` ASC) VISIBLE,
29 CONSTRAINT `fk_objeto_diagrama1` FOREIGN KEY (`diagrama_id`)
30 REFERENCES `mydb`.`diagrama` (`id`) ON DELETE CASCADE
31 ON UPDATE CASCADE) ENGINE = InnoDB;
32
33 CREATE TABLE IF NOT EXISTS `mydb`.`midia` (`id` INT NOT NULL,
34 `tipo` VARCHAR(150) NOT NULL,`arquivo` VARCHAR(255) NOT NULL,
35 `objeto_id` INT NOT NULL,PRIMARY KEY (`id`),
36 INDEX `fk_midia_objeto1_idx` (`objeto_id` ASC) VISIBLE,
37 CONSTRAINT `fk_midia_objeto1` FOREIGN KEY (`objeto_id`)
38 REFERENCES `mydb`.`objeto` (`id`) ON DELETE CASCADE
39 ON UPDATE CASCADE) ENGINE = InnoDB;
40
41 SET SQL_MODE=@OLD_SQL_MODE;
42 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
43 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
44

```

Fonte: (Próprio autor, 2022).

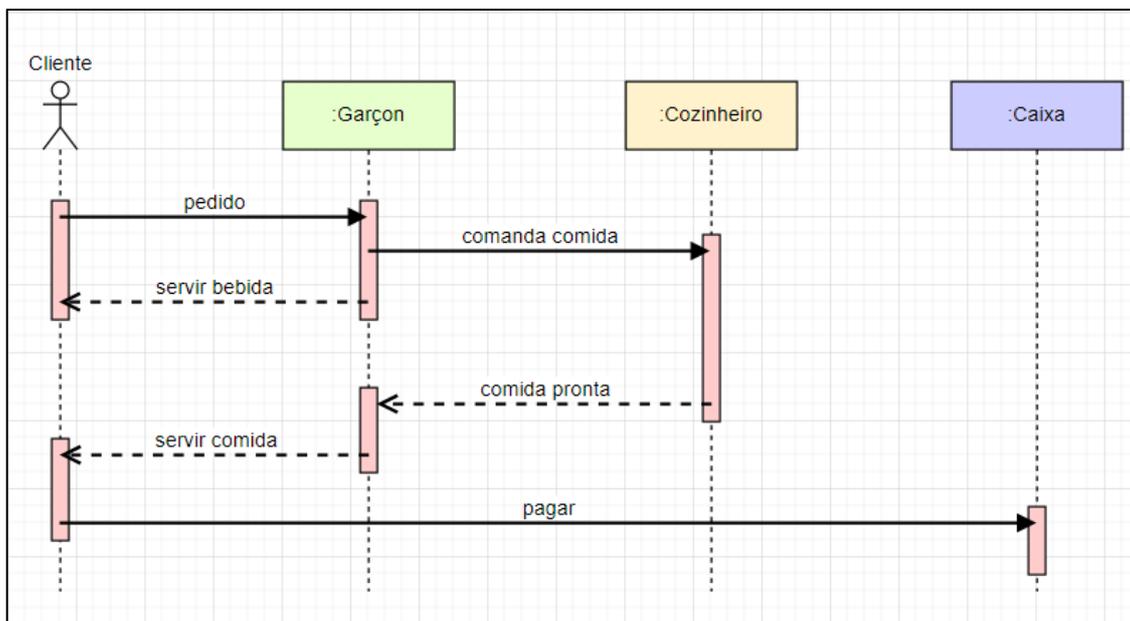
5.4 Teste

O teste é destinado a mostrar que um programa faz o que é proposto a fazer e para descobrir os defeitos do programa antes do uso. Quando se testa o software, o programa é executado usando dados fictícios. Os resultados do teste são verificados à procura de erros, anomalias ou informações sobre os atributos não funcionais do programa (SOMMERVILLE, 2011).

Com base nas especificações feitas na fase de projeto, foram realizados testes de validação, com modelos de diagramas, onde a ferramenta UMLibras

executou corretamente as modelagens de forma satisfatória, como exemplifica a Figura 31. Modelada na ferramenta UMLibras, ilustra a sequências de interações durante um processo de atendimento a um cliente em uma lanchonete.

Figura 31 – Diagrama de sequência de atendimento a um cliente.



Fonte: (Próprio autor, 2022).

5.5 Implementação

Essa seção demonstra o produto final da ferramenta UMLibras, abordando a operacionalidade e os resultados obtidos ao modelar diagramas de sequência.

5.5.1 Tela de login

Para acessar a ferramenta um utilizador necessita estar cadastrado com um usuário e senha. A Figura 32 ilustra a tela de login da ferramenta UMLibras.

Figura 32 - Tela de login da ferramenta UMLibras.



A tela de login da ferramenta UMLibras apresenta o logotipo 'U L LIBRAS' no topo, onde a letra 'L' é substituída por uma mão amarela. Abaixo do logotipo, há um formulário de login contendo dois campos de entrada: 'Seu usuário' e 'Sua senha'. Abaixo desses campos, há um botão azul com o texto 'Entrar'. Abaixo do botão 'Entrar', há o texto 'Não tem cadastro?' e um botão verde com o texto 'Cadastrar-se'.

Fonte: (Próprio autor, 2022).

Caso o usuário não possua cadastro no sistema, será necessário clicar no botão cadastrar-se e efetuar o preenchimento do formulário da tela seguinte Figura 33.

Figura 33 - Tela de cadastro de usuário.



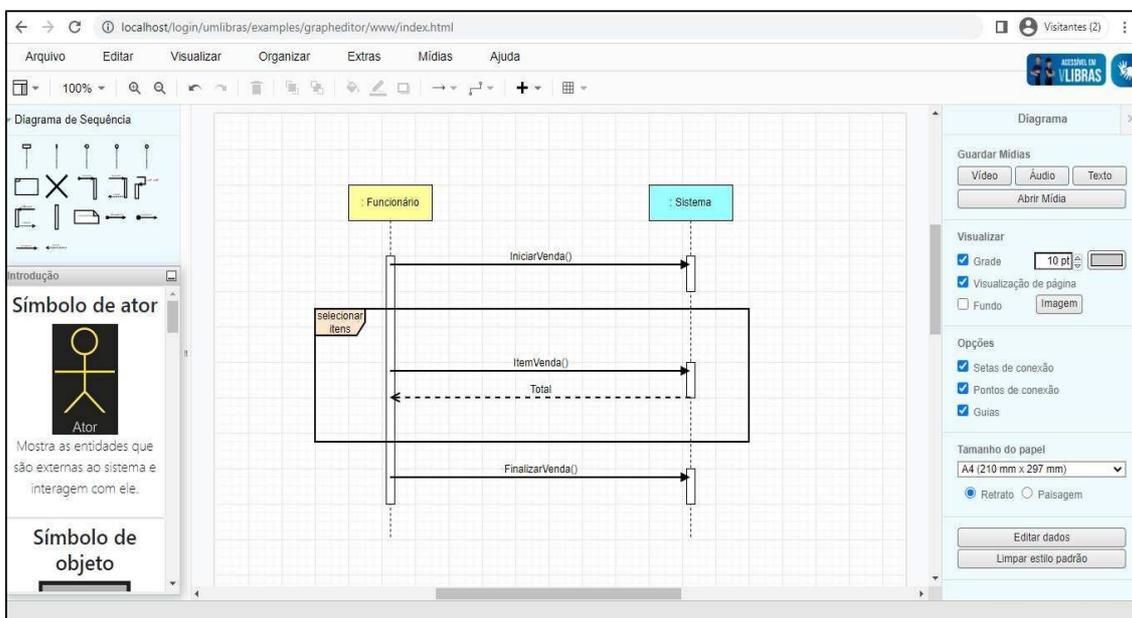
A tela de cadastro de usuário da ferramenta UMLibras apresenta o logotipo 'U L LIBRAS' no topo, onde a letra 'L' é substituída por uma mão amarela. Abaixo do logotipo, há um formulário de cadastro contendo os seguintes campos e opções: 'Nome', 'E-mail', 'Senha', 'Deficiência auditiva?' com opções de 'Sim' e 'Não', 'Entende Libras?' com opções de 'Sim' e 'Não', 'Instituição', 'UF' e 'País'. Abaixo dos campos, há um botão verde com o texto 'Cadastrar'.

Fonte: (Próprio autor, 2022).

5.5.2 Tela principal

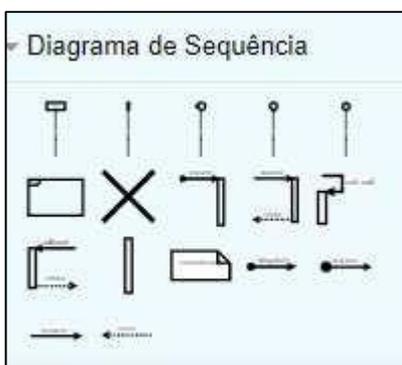
Na interface principal da ferramenta, Figura 34, o usuário poderá criar um novo diagrama utilizando os elementos da palheta denominada diagrama de sequência, como na Figura 35, ou ainda, abrir um diagrama existente, salvar, realizar a impressão, configurar, organizar e adicionar mídias através dos menus e botões das barras de ferramentas horizontais, como na Figura 36. Também é possível exportar ou importar o diagrama em formato XML e gerar arquivos de imagens nos formatos: PNG, JPEG, SVG, GIF e PDF, Figura 37.

Figura 34 - Tela principal com um diagrama de sequência modelado.



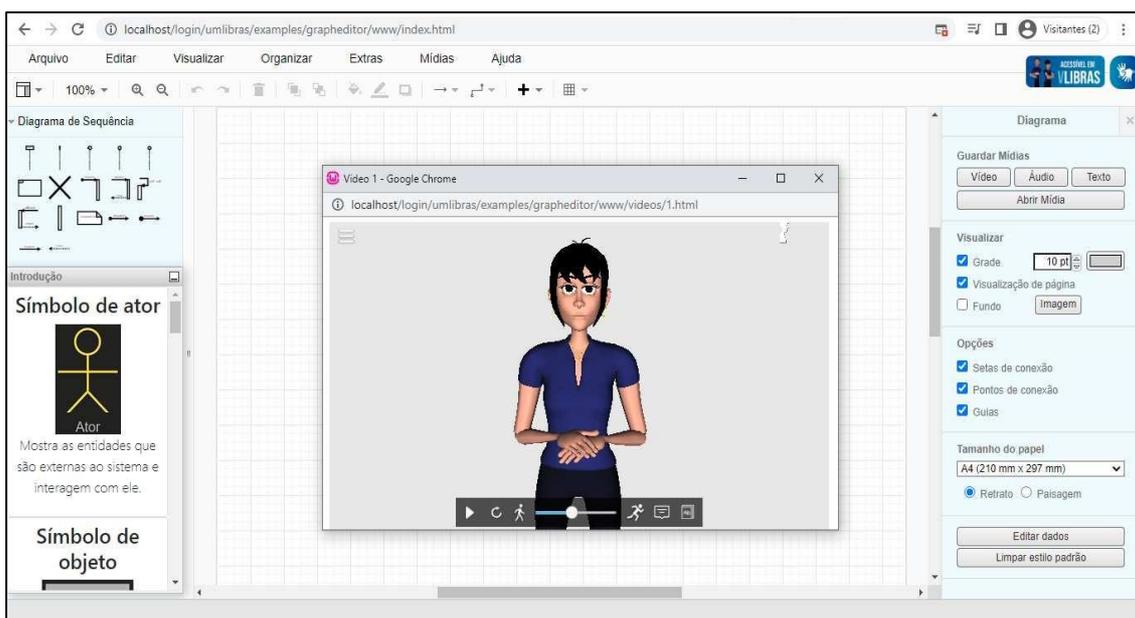
Fonte: (Próprio autor, 2022).

Figura 35 - Paleta diagrama de sequência.



Fonte: (Próprio autor, 2022).

Figura 38 - Vídeo de introdução.



Fonte: (Próprio autor, 2022).

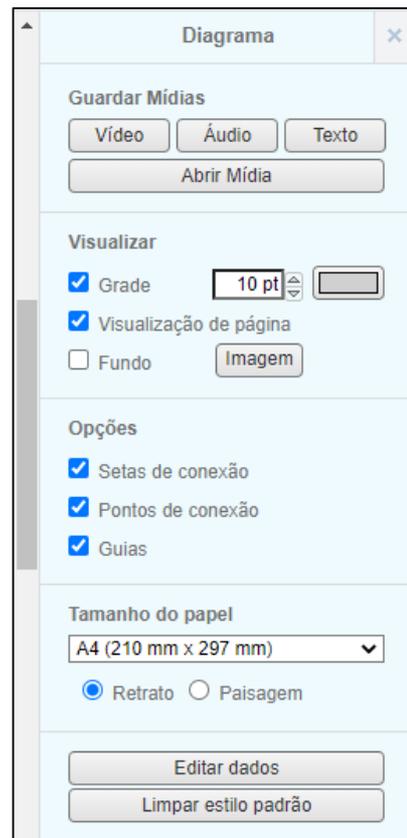
Figura 39 - Janela de introdução.



Fonte: (Próprio autor, 2022).

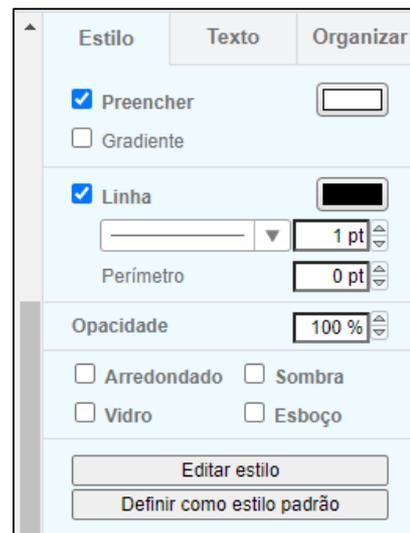
Na palheta a direita, denominada diagrama, Figura 40, é possível editar algumas configurações da área de trabalho. Clicando em algum elemento do diagrama, suas propriedades serão exibidas na mesma palheta, Figura 41. Também é possível a alteração do texto contido em um elemento através do duplo clique do mouse em cima do texto.

Figura 40 - Palheta Diagrama.



Fonte: (Próprio autor, 2022).

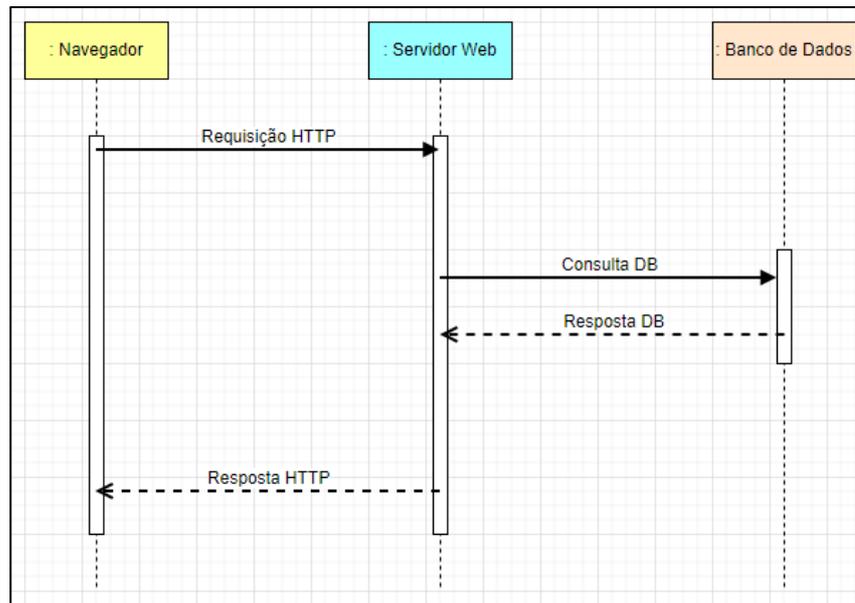
Figura 41 - Propriedades de um elemento selecionado.



Fonte: (Próprio autor, 2022).

A Figura 42 demonstra um exemplo de diagrama de sequência, modelado de forma satisfatória na ferramenta UMLibras, onde é possível compreender parte de um sistema web.

Figura 42 - Diagrama de sequência.



Fonte: (Próprio autor, 2022).

A Figura 43 apresenta a janela de gerenciamento de mídias, onde é possível armazenar vídeos, imagens e textos relacionados aos diagramas criados pelo usuário.

Figura 43 - Tela de gerenciamento de mídias.



Fonte: (Próprio autor, 2022).

6 CONSIDERAÇÕES FINAIS

O mercado disponibiliza uma considerável quantidade de ferramentas de tecnologia assistiva, ainda assim, não é suficiente para atender as demandas da comunidade surda. Ferramentas como VLibras, Rybená, Hand Talk, ProDeaf, WebLibras, &Ava, Primeira Mão e Transcrição Instantânea, possibilitam a tradução de textos e áudios para Libras, auxiliando em vários aspectos as pessoas com deficiência auditiva ou surdez.

Com tudo, ainda existiam lacunas com relação a ferramentas direcionadas à tecnologia UML. Todos os aplicativos disponíveis no mercado não atendem as necessidades de acessibilidade, dificultando sua utilização por uma pessoa com deficiência auditiva.

O presente trabalho desenvolveu o módulo de Diagrama de Sequência da UML integrado à ferramenta computacional UMLibras Web, idealizada pelo professor M.Sc. Sergio Augusto Coelho Bezerra do IFAM, e sob sua coordenação, alunos do Curso de Análise e Desenvolvimento de Sistemas do Departamento Acadêmico de Informação e Comunicação do Campus Manaus Centro vêm incrementando seu potencial para que outros diagramas da UML sejam contemplados.

A ferramenta desenvolvida atende aos principais requisitos de acessibilidade, apresentando uma interface simples e intuitiva, facilitando sua utilização e aprendizado da tecnologia UML quanto ao diagrama de sequência.

6.1 Trabalhos futuros

- Realizar testes de usabilidade da ferramenta com usuários que tenham ou não deficiência auditiva;
- Aplicar um questionário de avaliação com estes usuários, para verificar se a ferramenta está adequada às atividades propostas pelo professor e se favorece o aprendizado por parte do aluno.
- Desenvolvimento de outros módulos na ferramenta UMLibras para subsidiar os demais diagramas da UML.

7 REFERÊNCIAS

ALMEIDA, W. G. **Educação de surdos: formação, estratégias e prática docente**. Ilhéus: Editus, 2015.

APACHE. **The Apache HTTP Server Project**. Disponível em: <https://httpd.apache.org/ABOUT_APACHE.html>. Acesso em: 23 jun. 2022.

ASSISTIVA. **Tecnologia Assistiva**. Disponível em: <<https://www.assistiva.com.br/tassistiva.html>>. Acesso em: 2 jun. 2022.

AUDY, J.; PRIKLADNICKI, R. **Desenvolvimento distribuído de Software**. Rio de Janeiro: Elsevier, 2008.

AVA, & **ava - Legendas para todos**. Disponível em: <<https://pt.ava.me/>>. Acesso em: 22 jun. 2022.

BRASIL. **Lei nº 10.436, de 24 de abril de 2002**. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/2002/l10436.htm>. Acesso em: 2 maio. 2022.

BRASIL. **Lei nº 13.146, de 6 de julho de 2015**. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm>. Acesso em: 2 maio. 2022.

BRASIL. **Ministério da Economia. VLibras**. Disponível em: <<https://www.gov.br/governodigital/pt-br/vlibras>>. Acesso em: 15 maio. 2022.

CRISTIANO, A. **O que é Libras?** Disponível em: <<https://www.libras.com.br/o-que-e-libras>>. Acesso em: 1 abr. 2022.

FONSECA, A. J. B. **UMLibras: Ferramenta computacional para criação de diagramas de caso de uso com acessibilidade em libras**. Instituto Federal de

Educação, Ciência e Tecnologia do Amazonas (IFAM), 2018.

FREITAS, S. DE; FERREIRA, B. Q. N. **A IMPORTÂNCIA DA TECNOLOGIA NA EDUCAÇÃO ESPECIAL.** Disponível em: <<http://www.isciweb.com.br/revista/1035-a-importancia-da-tecnologia-na-educacao-especial>>. Acesso em: 21 jun. 2022.

GUEDES, G. T. A. **UML 2 - Uma abordagem prática.** 2. ed. São Paulo: Novatec, 2011.

HANDTALK, **HandTalk - Torne o site da sua empresa acessível.** Disponível em: <<https://handtalk.me/br/>>. Acesso em: 22 jun. 2022.

HEUSER, C. A. **Projeto de Banco de Dados.** 6. ed. Porto Alegre: Bookman, 2009.

IBGE. **Instituto Brasileiro de Geografia e Estatística - Censo 2010.** Disponível em: <<https://censo2010.ibge.gov.br/>>. Acesso em: 22 jun. 2022.

ICTS, G. **Rybená - Acessibilidade Web.** Disponível em: <<https://portal.rybena.com.br/site-rybena/>>. Acesso em: 9 dez. 2022.

INES, **Instituto Nacional de Educação de Surdos.** Disponível em: <<https://www.gov.br/ines/pt-br>>. Acesso em: 22 jun. 2022.

MDN. **CSS.** Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 14 jun. 2022.

MXGRAPH. **mxGraph 4.2.2.** Disponível em: <<https://jgraph.github.io/mxgraph/>>. Acesso em: 28 jun. 2022.

MYSQL. **MySQL 8.0 Reference Manual - 1 General Information.** Disponível em: <<https://dev.mysql.com/doc/refman/8.0/en/introduction.html>>. Acesso em: 14 jun. 2022.

ORTEGA, G. **Educação inclusiva: como a tecnologia ajuda nesse processo.**

Disponível em: <<https://escolasdisruptivas.com.br/metodologias-inovadoras/educacao-inclusiva-como-a-tecnologia-ajuda-nesse-processo/>>.

Acesso em: 8 maio. 2022.

PEREIRA, L. A. DE M. **Análise e Modelagem de Sistemas com a UML.** 1ª ed.

Rio de Janeiro: Luiz Antônio M. Pereira, 2011.

PMI. **GUIA PMBOK - Guia do conhecimento em gerenciamento de projetos.**

6. ed. Project Management Institute, 2017.

SILVA, A.; VIDEIRA, C. **UML, Metodologias e Ferramentas CASE.** Porto:

Centro Atlântico, 2001.

SOMMERVILLE, I. **Engenharia de Software.** 9. ed. São Paulo: Pearson 2011.

8 APÊNDICE A

Neste apêndice, são descritos os casos de usos que complementam os casos de uso demonstrados no item 4.4.

Tabela 7- Caso de uso salvar diagrama.

Item	Descrição
UC04	Permite ao usuário salvar um diagrama criado na área de trabalho.
Ator	Usuário.
Pré-condição	Não existe.
Fluxo principal	<p>P1 – o usuário clica no botão salvar da barra de ferramentas.</p> <p>P2 – o sistema abre uma janela solicitando que o usuário selecione o local onde deseja salvar o diagrama e após selecionar o local o usuário deverá clicar no botão salvar.</p> <p>P3 – o sistema exibe uma confirmação do salvamento do diagrama.</p>
Fluxo alternativo	<p>P1 – o usuário clica no menu arquivo opção salvar.</p> <p>P2 – o sistema abre uma janela solicitando que o usuário selecione o local onde deseja salvar o diagrama e após selecionar o local o usuário deverá clicar no botão salvar.</p> <p>P3 – o sistema exibe uma confirmação do salvamento do diagrama.</p>

Fonte: (Próprio autor, 2022).

Tabela 8 - Caso de uso abrir diagrama.

Item	Descrição
UC05	Permite ao usuário abrir um diagrama salvo anteriormente.
Ator	Usuário.
Pré-condição	Não existe.
Fluxo principal	<p>P1 – o usuário clica no botão abrir da barra de ferramentas.</p> <p>P2 – o sistema abre uma janela solicitando que o usuário selecione o local onde se encontra o diagrama que deseja abrir, após selecionar o arquivo do diagrama, o usuário deverá pressionar o botão abrir.</p> <p>P3 – o sistema abre uma nova aba com o diagrama selecionado pelo usuário no passo 2 anterior.</p>

Fluxo alternativo	<p>P1 – o usuário clica no menu arquivo opção abrir.</p> <p>P2 – o sistema abre uma janela solicitando que o usuário selecione o local onde se encontra o diagrama que deseja abrir, após selecionar o arquivo do diagrama, o usuário deverá pressionar o botão abrir.</p> <p>P3 – o sistema abre uma nova aba com o diagrama selecionado pelo usuário no passo 2 anterior.</p>
-------------------	---

Fonte: (Próprio autor, 2022).

Tabela 9 - Caso de uso exportar diagrama.

Item	Descrição
UC06	Permite ao usuário exportar um diagrama criado na área de trabalho.
Ator	Usuário.
Pré-condição	O usuário precisa ter algum diagrama criado na área de trabalho.
Fluxo principal	<p>P1 – o usuário clica no menu arquivo opção exportar.</p> <p>P2 – o sistema abre uma janela solicitando que o usuário selecione em qual tipo de arquivo deseja exportar o diagrama, as opções exibidas são: pdf, jpg ou png. Após selecionar o tipo, o usuário deverá clicar no botão exportar.</p> <p>P3 – o sistema exibe uma confirmação da exportação do diagrama.</p>
Fluxo alternativo	Nenhum

Fonte: (Próprio autor, 2022).

Tabela 10 - Caso de uso exibir introdução.

Item	Descrição
UC07	Permite ao usuário exibir um vídeo em libras introdutório sobre o sistema.
Ator	Usuário.
Pré-condição	Não existe.
Fluxo principal	<p>P1 – o usuário inicia o uso do sistema entrando na tela principal.</p> <p>P2 – o sistema exibe uma janela com vídeos de introdução.</p> <p>P2 – o usuário clica na ilustração para visualizar o vídeo de introdução.</p>
Fluxo alternativo	Nenhum

Fonte: (Próprio autor, 2022).



MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DO AMAZONAS

EMAIL Nº 350/2023 - PROT/CMC (11.01.03.01.08.08)

Nº do Protocolo: NÃO PROTOCOLADO

Manaus-AM, 15 de Fevereiro de 2023

Renato_Pereira_Ferreira_ENCAMINHA_TCC_E_FICHA_DE_AUTORIZACAO_150223.pdf

Total de páginas do documento original: 65

(Assinado digitalmente em 15/02/2023 17:42)

MOISES ISRAEL DE SOUZA ABREU

AUX EM ADMINISTRACAO

267959

Para verificar a autenticidade deste documento entre em <https://sig.ifam.edu.br/documentos/> informando seu número: **350**, ano: **2023**, tipo: **EMAIL**, data de Assinatura: **15/02/2023** e o código de verificação: **d1123d462e**