



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

DEMILSON PEREIRA DA COSTA

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

**MANAUS – AM
2023**

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

DEMILSON PEREIRA DA COSTA

**UMLIBRAS WEB: FERRAMENTA COMPUTACIONAL PARA CRIAÇÃO DE
DIAGRAMAS DE CASO DE USO 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
2023**

Biblioteca do IFAM – Campus Manaus Centro

C837u Costa, Demilson Pereira da.

Umlibras web: ferramenta computacional para criação de diagramas de caso de uso com acessibilidade em libras / Demilson Pereira da Costa. – Manaus, 2025.

71 p. : il. color.

Monografia (Tecnologia em Análise e Desenvolvimento de Sistema). – Instituto Federal de Educação, Ciência e Tecnologia do Amazonas, *Campus* Manaus Centro, 2025.

Orientador: Prof. Me. Sergio Augusto Coelho Bezerra.

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

CDD 005

Elaborada por Márcia Auzier CRB 11/597



**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**

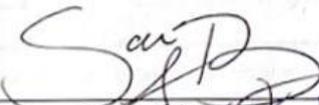


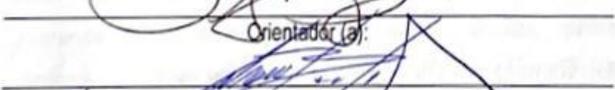
TERMO DE APROVAÇÃO

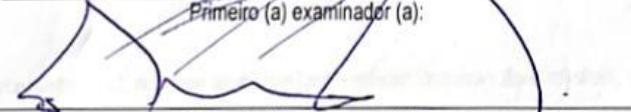
A monografia, que tem como título: UMLIBRAS WEB: FERRAMENTA COMPUTACIONAL PARA CRIAÇÃO DE DIAGRAMAS DE CASO DE USO COM ACESSIBILIDADE EM LIBRAS foi submetida à defesa pública, sob a avaliação de banca examinadora, como parte dos requisitos necessários para a obtenção do título de graduação do curso superior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

AUTOR (A): DEMILSON PEREIRA DA COSTA

Monografia aprovada em: 7 / 12 / 2023


Orientador (a):


Primeiro (a) examinador (a):


Segundo (a) examinador (a):

AGRADECIMENTOS

É com grande satisfação que expresso meus profundos agradecimentos a todos que contribuíram para a realização deste trabalho.

Primeiramente, agradeço a Deus, fonte de toda sabedoria e força, por guiar meus passos e iluminar meu caminho ao longo desta jornada acadêmica. Em segundo lugar, expresso minha gratidão aos meus pais, Deusdeth Almeida e Lucia Jeffes, pelos ótimos ensinamentos, amor incondicional e constante apoio. Suas palavras de encorajamento foram a luz que me guiou nos momentos desafiadores.

Em especial, dedico um agradecimento especial à minha esposa Glenda Oliveira, cujo apoio constante e incentivo foram pilares fundamentais para a conclusão deste trabalho. Sua compreensão, paciência e amor foram a inspiração que impulsionou meu empenho e dedicação.

Além disso, desejo estender meus agradecimentos ao meu orientador, Prof. M.Sc. Sergio Augusto C. Bezerra, pela orientação valiosa, apoio incansável e *insights* perspicazes.

Agradeço também à minha família, amigos e colegas pelo compartilhamento de conhecimentos, troca de ideias e apoio moral, tornando esta jornada mais rica e gratificante. Cada contribuição, por menor que fosse, foi fundamental para a conclusão deste trabalho e por isso dedico a todos vocês meus mais sinceros agradecimentos.

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

No contexto da educação inclusiva e tecnologia assistiva, este estudo se concentrou no desenvolvimento de uma ferramenta computacional para a criação de diagramas de Caso de Uso da Unified Modeling Language (UML) com recursos de acessibilidade em Língua Brasileira de Sinais (Libras) denominada UMLibras. A ferramenta visa apoiar docentes e discentes, com ou sem deficiência auditiva, quanto ao ensino-aprendizagem de modelagem de sistemas computacionais de diagramas. O modelo RAD (Rapid Application Development) foi adotado para o desenvolvimento da ferramenta. RAD é uma abordagem que se mostra particularmente adequada para projetos individuais que demandam desenvolvimento ágil e resultados rápidos, mantendo a qualidade do produto final. Utilizando a biblioteca JavaScript mxGraph e a API VLibras, a UMLibras foi projetada para auxiliar na modelagem do diagrama de Caso de Uso com associação de figuras e vídeos com intérpretes. Espera-se que a UMLibras contribua significativamente para o ensino-aprendizagem de tais modelos da UML, buscando oferecer suporte ao docente quanto ao ensino, bem como criar condições favoráveis para a aprendizagem desses discentes. Nesse sentido, o uso da UMLibras nas instituições de ensino pode promover, de maneira inclusiva e igualitária, tanto a educação quanto o aumento das ofertas de emprego.

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

ABSTRACT

In the context of inclusive education and assistive technology, this study focused on the development of a computational tool for creating Unified Modeling Language (UML) Use Case diagrams with accessibility features in Brazilian Sign Language (Libras), called UMLibras. The tool aims to support teachers and students, with or without hearing impairments, in the teaching and learning of modeling computer systems from diagrams. The RAD (Rapid Application Development) model was adopted for the development of the tool. RAD is an approach that is particularly suitable for individual projects that require agile development and fast results, while maintaining the quality of the final product. Using the JavaScript library mxGraph and the VLibras API, UMLibras was designed to assist in the modeling of the Use Case diagram with associated figures and videos with interpreters. It is expected that UMLibras will contribute significantly to the teaching and learning of such UML models, seeking to offer support to the teacher in teaching, as well as to create favorable conditions for the learning of these students. In this sense, the use of UMLibras in educational institutions can promote both education and the increase in job offers in an inclusive and egalitarian manner.

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

LISTA DE FIGURAS

FIGURA 1 - MODELO RAD (DESENVOLVIMENTO RÁPIDO DE APLICATIVOS).	18
FIGURA 2 - NO DIAGRAMA DE CASO DE USO, REPRESENTO A INTERAÇÃO DO ATOR 'USUÁRIO LOGADO' COM OS CASOS DE USO 'CRIAR', 'SALVAR' E 'EDITAR DIAGRAMA DE CASO DE USO'.	25
FIGURA 3 - DISPOSITIVOS COM VLIBRAS (VLIBRAS, 2024).	25
FIGURA 4 - PERSONAGEM HUGO.	26
FIGURA 5 - TRADUTOR VIRTUAL RYBENÁ.	27
FIGURA 6 - LOGO DA AVA.	27
FIGURA 7 - JORNAL PRIMEIRA MÃO.	28
FIGURA 8 - ARQUITETURA DA BIBLIOTECA MXGRAPH.	32
FIGURA 9 - UMLIBRAS DESKTOP.	34
FIGURA 10 - DIAGRAMA DE CASO DE USO.	40
FIGURA 11 - MODELO ENTIDADE-RELACIONAMENTO DA FERRAMENTA UMLIBRAS.	42
FIGURA 12 - MODELO LÓGICO (FORMA TEXTUAL) DA FERRAMENTA UMLIBRAS.	43
FIGURA 13 - DIAGRAMA LÓGICO DA FERRAMENTA UMLIBRAS.	44
FIGURA 14 - DIAGRAMA DE CLASSES.	45
FIGURA 15 - PROTÓTIPO DA TELA DE LOGIN.	45
FIGURA 16 - PROTÓTIPO DA TELA DE CADASTRO DE USUÁRIO.	46
FIGURA 17 - PROTÓTIPO DA TELA PRINCIPAL DA FERRAMENTA UMLIBRAS.	46
FIGURA 18 - PROTÓTIPO DA TELA DE GERENCIAMENTO DE MÍDIAS.	47
FIGURA 19 - CÓDIGO PARA CARREGAR A BIBLIOTECA MXGRAPH.	48
FIGURA 20 - CÓDIGO PARA A CONSTRUÇÃO DE UM GRÁFICO.	48
FIGURA 21 - CÓDIGO PARA INSERIR VÉRTICES E ARESTAS.	49
FIGURA 22 - CÓDIGO PARA CRIAR UM MODELO DE GRÁFICO.	49
FIGURA 23 - CÓDIGO PARA MODIFICAR ESTILOS.	50
FIGURA 24 - CÓDIGO PARA ADICIONAR FOLHA DE ESTILO.	50
FIGURA 25 - CÓDIGO DE UM OBJETO JAVASCRIPT.	51
FIGURA 26 - CÓDIGO PARA CONVERTER UM OBJETO EM XML.	51
FIGURA 27 - RESULTADO DO OBJETO CONVERTIDO EM XML.	51
FIGURA 28 - EXEMPLO DE ARQUIVO PHP.	52

FIGURA 29 - CÓDIGO PARA INTEGRAR A API VLIBRAS À UM WEBSITE.....	53
FIGURA 30 - CÓDIGO PARA CRIAR AS TABELAS DO BANCO DE DADOS DA UMLIBRAS.....	53
FIGURA 31 - TELA DE LOGIN DA FERRAMENTA UMLIBRAS.....	54
FIGURA 32 - TELA DE CADASTRO DE USUÁRIO.....	54
FIGURA 33 - TELA PRINCIPAL COM UM DIAGRAMA DE CASO DE USO E UM EXEMPLO PRÁTICO.....	55
FIGURA 34 - PALETA DIAGRAMA DE CASO DE USO.....	56
FIGURA 35 - MENU E BARRA DE FERRAMENTA HORIZONTAL.....	56
FIGURA 36 - VÍDEO DE INTRODUÇÃO.....	57
FIGURA 37 - JANELA DE INTRODUÇÃO.....	57
FIGURA 38 - PALHETA DIAGRAMA.....	58
FIGURA 39 - PROPRIEDADES DE UM ELEMENTO SELECIONADO.....	58
FIGURA 40 - DIAGRAMA DE CASO DE USO.....	59
FIGURA 41 - TELA DE GERENCIAMENTO DE MÍDIAS, IMPLEMENTAÇÃO FUTURA.....	59

LISTA DE TABELAS

TABELA 1 - FERRAMENTAS UML DISPONÍVEIS NO MERCADO.....	35
TABELA 2 - LISTA DOS REQUISITOS FUNCIONAIS DA FERRAMENTA.....	38
TABELA 3 - LISTA DOS REQUISITOS NÃO-FUNCIONAIS DA FERRAMENTA.....	38
TABELA 4 - CASO DE USO: CRIAR DIAGRAMA.....	40
TABELA 5 - CASO DE USO: ABRIR VLIBRAS.....	41
TABELA 6 - CASO DE USO: EXIBIR INFORMAÇÕES.....	41
TABELA 7 - CASO DE USO SALVAR DIAGRAMA.....	66
TABELA 8 - CASO DE USO ABRIR DIAGRAMA.....	66
TABELA 9 - CASO DE USO EXPORTAR DIAGRAMA.....	67
TABELA 10 - CASO DE USO EXIBIR INTRODUÇÃO.....	67

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
PcD	Pessoa com Deficiência
DCU	Diagrama de Caso de Uso
DCUs	Diagrama de Casos de Uso

SUMÁRIO

1	INTRODUÇÃO	14
1.1	CONTEXTUALIZAÇÃO E JUSTIFICATIVA	14
1.2	OBJETIVOS.....	17
1.2.1	<i>Objetivo geral</i>	17
1.2.2	<i>Objetivos específicos</i>	17
1.3	PROCESSO METODOLÓGICO.....	17
1.4	ORGANIZAÇÃO DA MONOGRAFIA.....	19
2	FUNDAMENTAÇÃO TEÓRICA	21
2.1	ACESSIBILIDADE	21
2.2	LIBRAS	22
2.3	TECNOLOGIA UML	23
2.4	DIAGRAMA DE CASO DE USO.....	23
2.5	TECNOLOGIAS ASSISTIVAS	24
2.5.1	<i>VLibras</i>	25
2.5.2	<i>Hand Talk</i>	26
2.5.3	<i>Transcrição Instantânea</i>	26
2.5.4	<i>Prodeaf</i>	26
2.5.5	<i>WebLibras</i>	26
2.5.6	<i>Rybená</i>	27
2.5.7	<i>&Ava</i>	27
2.5.8	<i>Primeira Mão</i>	28
3	TECNOLOGIAS UTILIZADAS	29
3.1	NODEJS COM FRAMEWORK EXPRESS.....	29
3.2	POSTGRESQL.....	30
3.3	REACTJS	30
3.4	STYLED-COMPONENTS	31
3.5	VLIBRAS.....	31
3.6	MXGRAPH.....	32
4	TRABALHOS RELACIONADOS	34
4.1	UMLIBRAS VERSÃO DESKTOP	34
4.2	FERRAMENTAS SEMELHANTES.....	35
5	FERRAMENTA DESENVOLVIDA	37
5.1	LEVANTAMENTO E ANÁLISE DE REQUISITOS.....	37
5.1.1	<i>Requisitos funcionais</i>	37

5.1.2	<i>Requisitos não-funcionais</i>	38
5.2	PROJETO	38
5.2.1	<i>Estrutura básica</i>	39
5.2.2	<i>Diagrama de caso de uso</i>	39
5.2.3	<i>Descrição dos casos de uso</i>	40
5.2.4	<i>Modelo conceitual do banco de dados</i>	42
5.2.5	<i>Modelo lógico do banco de dados</i>	43
5.2.6	<i>Diagrama de classes</i>	44
5.2.7	<i>Interface</i>	45
5.3	CODIFICAÇÃO	47
5.3.1	<i>Biblioteca mxGraph</i>	47
5.3.2	<i>Gerar gráfico</i>	48
5.3.3	<i>Vértices e arestas</i>	48
5.3.4	<i>Modelo</i>	49
5.3.5	<i>Folha de estilo</i>	50
5.3.6	<i>CSS</i>	50
5.3.7	<i>Gerar XML do gráfico</i>	51
5.3.8	<i>Arquivos</i>	52
5.3.9	<i>Integração VLibras</i>	52
5.3.10	<i>Banco de Dados</i>	53
5.4	IMPLEMENTAÇÃO	54
5.4.1	<i>Tela de login</i>	54
5.4.2	<i>Tela principal</i>	55
6	CONSIDERAÇÕES FINAIS	60
6.1	TRABALHOS FUTUROS.....	61

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO E JUSTIFICATIVA

No cenário atual da educação inclusiva, observa-se a implementação de um modelo de ensino equitativo que visa atender de forma eficaz às necessidades educacionais do grupo de pessoas com exigências especiais, conforme Lei nº 13.146, de 6 de julho de 2015 (Lei Brasileira de Inclusão da Pessoa com Deficiência). Nesse contexto, o Art. 2º dessa lei considera que uma pessoa com deficiência (PcD) é 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).

Este marco legal foi posteriormente fortalecido pela Lei nº 14.191/2021, que alterou a Lei de Diretrizes e Bases da Educação Nacional para incluir disposições específicas sobre a educação bilíngue de surdos, reconhecendo a importância da modalidade específica de ensino para esta comunidade (BRASIL, 2021). Adicionalmente, o Plano Nacional dos Direitos da Pessoa com Deficiência - Plano Viver Sem Limite (Decreto 7.612/2011) estabelece diretrizes fundamentais para garantir o acesso à educação inclusiva de qualidade (BRASIL, 2011).

Por meio de práticas da educação inclusiva, pessoas com necessidades especiais têm a oportunidade de participar do processo educacional de maneira equiparada aos seus colegas, sem que isso prejudique seu desempenho acadêmico (ORTEGA, 2019). Para corroborar com esse processo educacional, surgem às tecnologias assistivas, que englobam diversas ferramentas utilizadas com o objetivo de proporcionar acessibilidade, maior independência e autonomia às pessoas com necessidades especiais, inclusive servindo como apoio ao corpo docente.

A acessibilidade das pessoas com deficiência auditiva ou surdez é comprometida pela barreira comunicativa, um obstáculo crucial para o desenvolvimento social e intelectual desses indivíduos no contexto inclusivo. No Brasil, segundo dados recentes do Censo 2022 do Instituto Brasileiro de Geografia e Estatística (IBGE), aproximadamente 18,6 milhões de pessoas (8,9% da população) possuem algum tipo de

deficiência auditiva, representando um aumento significativo em relação aos dados anteriores (IBGE, 2022).

Nesse contexto, as tecnologias assistivas são de fundamental importância na inclusão e na diminuição da barreira entre surdos e ouvintes. Um exemplo significativo é a suíte VLibras, que é um conjunto de ferramentas computacionais de código aberto desenvolvido através de parceria entre o Ministério da Economia e a Universidade Federal da Paraíba. Esta suíte realiza a tradução automática de conteúdos digitais (texto, áudio e vídeo) para a Língua Brasileira de Sinais, sendo disponibilizada gratuita e constantemente atualizada para melhor atender às necessidades da comunidade surda (BRASIL, 2023).

A Língua Brasileira de Sinais, representada pela sigla Libras, constitui-se como uma língua gestual-visual que possibilita a comunicação por meio de gestos, expressões faciais e corporais. Amplamente empregada na interação com indivíduos surdos, a Libras emerge como uma ferramenta crucial para promover a inclusão social. Sua importância foi ainda mais reconhecida com a implementação da educação bilíngue, que considera a Libras como primeira língua e a modalidade escrita da língua portuguesa como segunda língua para estudantes surdos, conforme estabelecido pela legislação mais recente (CRISTIANO, 2017).

Na contemporaneidade, diversas tecnologias assistivas foram desenvolvidas para apoiar a comunidade surda. Uma variedade de aplicativos tem se destacado no mercado, oferecendo diferentes funcionalidades para facilitar a comunicação e o acesso à informação. Entre as principais soluções disponíveis, destacam-se: VLibras: ferramenta governamental de código aberto que realiza tradução automática para Libras (VLibras, 2024); Hand Talk: aplicativo premiado internacionalmente que oferece tradução em tempo real (Hand Talk, 2024); Rybená: solução que converte textos em Libras e voz (Rybená, 2024); WebLibras: plataforma web para tradução e comunicação em Libras (WebLibras, 2024); &Ava e Primeira Mão: ferramentas mais recentes que incorporam recursos de inteligência artificial para melhorar a precisão das traduções (Ava, 2024), (Primeira Mão, 2024).

Além dessas tecnologias específicas, aplicativos de mensagens, como o WhatsApp, Telegram, dentre outros, têm se mostrado fundamentais para a comunidade surda, facilitando a comunicação escrita para aqueles que dominam a língua portuguesa.

No contexto específico do ensino de tecnologia, FONSECA (2018) destaca uma lacuna significativa no desenvolvimento de tecnologias assistivas voltadas ao ensino e aprendizagem de diagramas da UML (Unified Modeling Language). As ferramentas mais populares de modelagem UML como Astah, Umbrello, Argo UML, Poseidon, Diagrame, dentre outras, não possuem recursos de acessibilidade, com exceção da iniciativa pioneira desenvolvida no Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM), Campus Manaus Centro.

Sob a orientação do professor Sergio Augusto C. Bezerra, alunos do Curso de Análise e Desenvolvimento de Sistemas (TADS) desenvolveram a UMLibras, uma ferramenta computacional inovadora que busca preencher essa lacuna. A primeira versão da ferramenta, desenvolvida para ambiente *desktop*, permitia a criação de Diagramas de Caso de Uso da UML. Visando expandir o alcance e as funcionalidades da ferramenta, o projeto evoluiu para incluir outros diagramas da UML, mas voltados para uma plataforma web, tornando-se dessa forma mais acessível e abrangente. Esta iniciativa representa um importante avanço na acessibilidade do ensino de modelagem de sistemas, demonstrando como projetos acadêmicos podem contribuir significativamente para a inclusão digital da comunidade surda no campo da tecnologia da informação. Vale ressaltar, que o IFAM desenvolve diversos outros trabalhos que contribuem com a comunidade surda, principalmente, sob o comando do professor Dalmir Pacheco de Souza no Campus Manaus Centro (PACHECO, 2024).

Nesse contexto, este trabalho contribui com o avanço da ferramenta UMLibras Web concernente ao desenvolvimento do módulo para elaboração de diagramas de Caso de Uso. Espera-se que este novo módulo da UMLibras contribua significativamente para o ensino-aprendizagem de tais modelos da UML, buscando oferecer suporte ao docente quanto ao ensino, bem como criar condições favoráveis para a aprendizagem desses discentes. Nesse sentido, o uso da UMLibras nas instituições de ensino pode promover de maneira inclusiva e igualitária tanto a educação quanto o aumento das ofertas de emprego.

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 Caso de Uso da *Unified Modeling Language* (UML) com recursos de acessibilidade em Língua Brasileira de Sinais (Libras).

1.2.2 Objetivos específicos

Com o intuito de atingir a meta global deste estudo, foram estabelecidos os seguintes objetivos específicos:

- Tornar acessível para pessoas surdas o processo de criação de diagrama de Caso de Uso 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 Caso de Uso;
- Incorporar imagens autoexplicativas e *links* à ferramenta para serem executados GIFs ou vídeos em Libras.

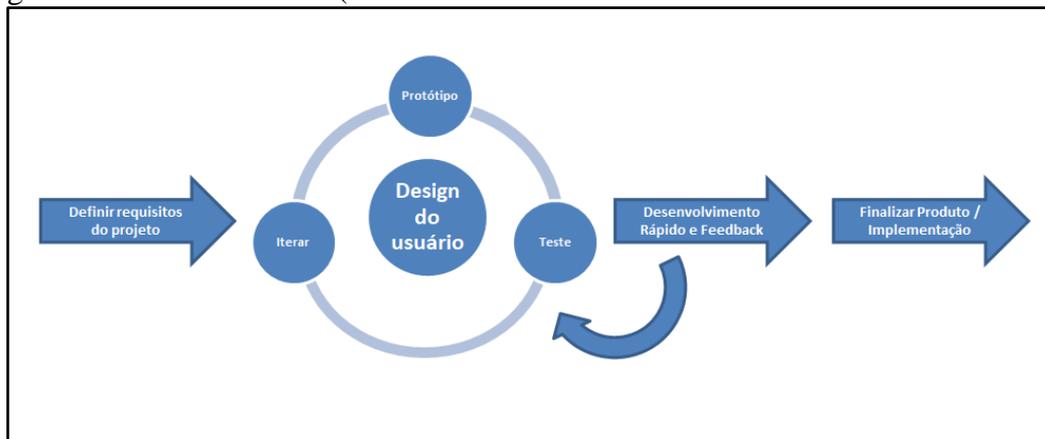
1.3 PROCESSO METODOLÓGICO

Para o desenvolvimento deste projeto, foi adotado o modelo RAD (Rapid Application Development). Esse modelo é adequado para projetos individuais que demandam desenvolvimento ágil e resultados rápidos, mantendo a qualidade do produto final (McGraw Hill, 2011).

O RAD é um modelo de processo de desenvolvimento de software incremental, que foi registrado por James Martin, em 1991. É um processo de desenvolvimento de aplicações de forma rápida com objetivos bem definidos e análise de requisitos extremamente bem alinhada. Esse modelo enfatiza um ciclo de desenvolvimento curto, com o intuito de ter um desenvolvimento melhor e mais rápido. A Figura 1 apresenta as

etapas de um processo de desenvolvimento usando o modelo RAD (McGraw Hill, 2011).

Figura 1 - MODELO RAD (DESENVOLVIMENTO RÁPIDO DE APLICATIVOS).



Fonte: Adaptado de RADIXWEB, 2024.

No desenvolvimento incremental, uma das características de RAD, o sistema é dividido em módulos, tomando por base a funcionalidade. Tendo os incrementos definidos, a cada ciclo é acrescentado de novas funcionalidades ou até mesmo modificações, caso seja necessário. Outra característica é justamente essa maleabilidade de adaptação dos processos e a capacidade de se manter em constante evolução (Pearson, 2011).

Segundo Roger S. (Pearson, 2011), o modelo RAD se estrutura em quatro fases principais, permitindo uma abordagem mais dinâmica e iterativa ao desenvolvimento:

Planejamento dos Requisitos: Nesta fase inicial, foram identificadas e documentadas as necessidades fundamentais do sistema. Diferentemente do modelo cascata tradicional, o RAD permite uma definição mais ágil dos requisitos, focando nas funcionalidades essenciais que agregam valor imediato ao projeto.

Design do Usuário: Durante esta etapa, foram desenvolvidos os protótipos da interface e definida a arquitetura do sistema. O processo envolveu a criação de modelos e a definição da lógica de negócio, com foco na experiência do usuário e na usabilidade da ferramenta.

Construção: A fase de construção no RAD é caracterizada por ciclos curtos de desenvolvimento, onde a codificação e os testes são realizados de forma integrada. Esta abordagem permitiu a identificação e correção rápida de problemas, sem a

necessidade de passar por longos ciclos de desenvolvimento como no modelo cascata.

Implementação: A fase final engloba a implantação do sistema em ambiente de produção, incluindo testes de aceitação e eventuais ajustes finais. O modelo RAD facilita esta transição por incorporar feedback e correções ao longo do processo de desenvolvimento.

Uma das principais vantagens do RAD em relação ao modelo cascata é sua flexibilidade para acomodar mudanças durante o desenvolvimento, característica especialmente importante em projetos acadêmicos como este. O modelo permite que ajustes sejam realizados de forma mais ágil, sem comprometer a estrutura do projeto ou gerar atrasos significativos (McGraw-Hill, 1994).

1.4 ORGANIZAÇÃO DA MONOGRAFIA

Este trabalho está estruturado em oito capítulos, conforme delineados a seguir.

No capítulo 1, o projeto é introduzido por meio de sua contextualização e justificativa, abordando a apresentação do problema e a proposta de solução. São destacados o objetivo geral, os objetivos específicos e a metodologia adotada durante o desenvolvimento do trabalho.

O capítulo 2 aborda os temas fundamentais para o desenvolvimento deste estudo. São discutidos os conceitos de acessibilidade e Libras, as legislações pertinentes à comunidade surda, algumas ferramentas de tecnologia assistiva, além de ser apresentado 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 positivos e negativos do trabalho referente à primeira versão da ferramenta UMLibras, bem como 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 neste trabalho.

Por fim, são listadas as referências bibliográficas que serviram de base para a construção deste trabalho, assim como é apresentado o apêndice com os casos de uso complementares.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo são apresentados os fundamentos que servem de base para o entendimento da acessibilidade na educação inclusiva, bem como o papel da tecnologia como ferramenta de apoio à comunicação e à aprendizagem da comunidade surda.

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 discentes, 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 docente 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. A Lei nº 13.146 (2015) reforça a importância de garantir condições para que pessoas com deficiência, incluindo surdos, tenham acesso igualitário à informação e à comunicação. 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).

Nesse contexto, é necessário também tornar computadores e dispositivos móveis acessíveis, facilitando o acesso ao conteúdo dessas tecnologias e reduzindo barreiras de comunicação e acesso à informação (ALMEIDA, 2015). A garantia de acessibilidade digital está em consonância com os princípios estabelecidos na Lei nº 13.146 (2015),

que prevê a inclusão por meio da eliminação de barreiras tecnológicas e comunicacionais.

As políticas públicas voltadas para a inclusão e a acessibilidade têm desempenhado um papel essencial na promoção da visibilidade social das pessoas com deficiência. Pacheco (2014) destaca que, apesar de vivermos em uma sociedade orientada por padrões de normalidade e pela lógica mercadológica, as iniciativas voltadas para o trabalho e a educação têm permitido que pessoas com deficiência conquistem maior autonomia e cidadania. A experiência do Projeto Curupira, desenvolvido pelo Instituto Federal do Amazonas entre 2008 e 2012, ilustra esse avanço ao promover a formação profissional de pessoas com deficiência e a produção de materiais didáticos acessíveis e tecnologia assistiva, reforçando o paradigma inclusivo e superando a visão caritativa e médica da deficiência (PACHECO, 2014).

2.2 LIBRAS

De acordo com a Lei nº 10.436, de 24 de abril de 2002, a Língua Brasileira de Sinais (Libras) é reconhecida como o sistema de comunicação e expressão das comunidades surdas no Brasil. Trata-se de uma linguagem de natureza visual-motora, com estrutura gramatical própria, utilizada para a transmissão de ideias e informações entre pessoas surdas (BRASIL, 2002).

A Libras desempenha um papel fundamental na comunicação da comunidade surda, pois possibilita maior autonomia e facilita a interação com pessoas ouvintes, reduzindo as barreiras comunicacionais e promovendo a inclusão (ALMEIDA, 2015). No Brasil, a oficialização da Libras como língua natural da comunidade surda, por meio da Lei nº 10.436/2002 e regulamentada pelo Decreto nº 5.626/2005, representa um avanço significativo para garantir o acesso à educação e à comunicação, respeitando as especificidades dessa comunidade (ALMEIDA, 2015).

A estrutura da Libras é composta por elementos linguísticos próprios, incluindo gramática, sintaxe e morfologia. Para que a mensagem em Libras seja compreendida com precisão, é necessário observar cinco parâmetros essenciais: a configuração das mãos, o ponto de articulação (local onde o sinal é realizado, como na testa ou à frente do corpo), a orientação das mãos (direção em que o sinal é feito), o movimento (se o sinal envolve ou não deslocamento) e a expressão facial e corporal, que complementa o significado e a intencionalidade da mensagem (ALMEIDA, 2015).

2.3 TECNOLOGIA UML

A modelagem de sistemas consiste no desenvolvimento de modelos abstratos para representar um sistema, onde cada modelo proporciona uma visão ou perspectiva única do mesmo. Essa prática geralmente emprega alguma forma de notação gráfica, sendo que, nos dias atuais, essa notação é frequentemente fundamentada em UML (Linguagem de Modelagem Unificada, do inglês Unified Modeling Language) (SOMMERVILLE, 2011).

Segundo GUEDES (2011), a UML teve origem a partir da integração 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. Até meados da década de 1990, esses métodos eram os mais populares na área de desenvolvimento de software.

Para assegurar que a compreensão de um modelo seja clara e inequívoca, é imprescindível utilizar uma linguagem cujos símbolos possuam semânticas (significados) e regras de combinação precisamente definidos. Além disso, para garantir a abrangência adequada do modelo, a linguagem deve ser expressiva, permitindo a descrição concisa de todos os aspectos necessários da realidade que está sendo modelada (PEREIRA, 2011).

2.4 DIAGRAMA DE CASO DE USO

Os diagramas de caso de uso (DCUs) fornecem uma representação visual das interações entre atores e os casos de uso em um sistema. Eles descrevem as funcionalidades ou serviços oferecidos por um sistema e como os atores interagem com essas funcionalidades. Os DCUs oferecem uma perspectiva de alto nível sobre o que o sistema faz, centrando-se nas relações entre os atores externos e as principais funcionalidades do sistema (PEREIRA, 2011).

A Figura 2 apresenta um exemplo de DCU, proporcionando uma representação visual dos seus princípios fundamentais. Este diagrama tem por objetivo capturar as interações inerentes aos DCUs denominado 'Criar, Salvar e Editar Diagrama de Caso de

Uso', onde um usuário possui a capacidade de criar um diagrama, salvá-lo e posteriormente editar as informações contidas no mesmo.

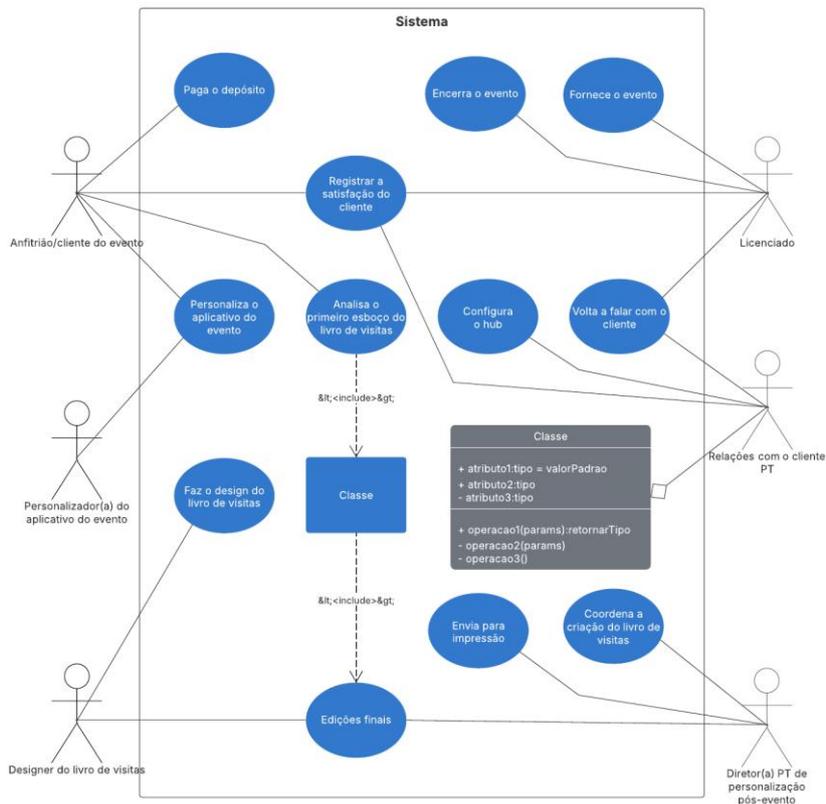
No diagrama de caso de uso, os atores, como o 'Usuário Logado', estão associados a casos de uso específicos, como 'Criar Diagrama de Caso de Uso', 'Salvar Diagrama de Caso de Uso' e 'Editar Informações do Diagrama de Caso de Uso'. Essas associações são representadas por linhas de comunicação que evidenciam as interações entre o ator e as funcionalidades do sistema. As conexões estabelecidas no diagrama permitem visualizar de forma clara o fluxo de execução e as responsabilidades de cada elemento, proporcionando uma melhor compreensão do funcionamento do sistema e da dinâmica entre os componentes (PRESSMAN, 2011).

Em um diagrama de caso de uso, os atores desempenham um papel fundamental na definição das funcionalidades do sistema, sendo conectados aos casos de uso por meio de associações. Essas associações ilustram como o ator interage com o sistema e quais funcionalidades são acessadas em diferentes cenários. A representação gráfica permite visualizar o comportamento do sistema, as interdependências entre os casos de uso e o fluxo de execução, facilitando o entendimento do processo e auxiliando na modelagem de requisitos (PRESSMAN, 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 descritas as seguintes ferramentas: VLibras, Hand Talk, Primeira Mão, &Ava entre outras.

Figura 2 - NO DIAGRAMA DE CASO DE USO, REPRESENTO A INTERAÇÃO DO ATOR 'USUÁRIO LOGADO' COM OS CASOS DE USO 'CRIAR', 'SALVAR' E 'EDITAR DIAGRAMA DE CASO DE USO'.



Fonte: (PRESSMAN, 2011).

2.5.1 VLibras

Um exemplo de tecnologia assistiva é a suíte VLibras (VLibras, 2024), 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 (VLibras, 2024).



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, exibido na Figura 4, que funciona como uma interface que traduz textos em Português para Libras. O software converte a mensagens SMS e se o surdo fotografar imagens com legendas também vai poder obter a tradução instantaneamente (Hand Talk, 2022)

Figura 4 - PERSONAGEM HUGO.



Fonte: (HandTalk, 2022)

2.5.3 Transcrição Instantânea

É um aplicativo do Google que está disponível na play store para aparelhos móveis, capaz de 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 (ASSISTIVA, 2022).

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 ou digita uma frase e um avatar, realiza a tradução para a língua de sinais. O aplicativo é totalmente gratuito, e pode ser utilizado nos aparelhos com Android (ASSISTIVA, 2022).

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 (ASSISTIVA, 2022).

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 (AVA, 2022).

Como descrito em (AVA, 2022), o aplicativo 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 cha. 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, o “Primeira Mão”, como mostrado na Figura 7, é outro aplicativo inovador que transmite informações via um jornal semanal bilíngue em Português e Libras (ALMEIDA, 2015). Ele compartilha as principais notícias do Brasil e do mundo, garantindo que a informação chegue ao público surdo. O aplicativo está disponível nas lojas Android e iOS.

Figura 7 - JORNAL PRIMEIRA MÃO.



Fonte: (INES, 2022)

3 TECNOLOGIAS UTILIZADAS

Neste capítulo são apresentadas as tecnologias utilizadas nas fases de desenvolvimento do DCU da ferramenta UMLibras Web.

3.1 NODEJS COM FRAMEWORK EXPRESS

A arquitetura da aplicação, centrada em uma API desenvolvida com Node.js e o framework Express, proporciona uma base sólida para a construção de sistemas robustos e eficientes. O Node.js, conhecido por sua eficiência em operações assíncronas, permite que a aplicação atenda a múltiplas requisições simultaneamente, otimizando o desempenho (NODE.JS, 2024). Por sua vez, o Express oferece uma estrutura minimalista e flexível, baseada em rotas e middlewares, que facilita a criação de pontos de extremidade (*endpoints*) bem definidos, promovendo boas práticas de desenvolvimento e garantindo uma comunicação eficaz entre o front-end e o back-end.

Além disso, o framework simplifica a implementação de autenticação, autorização e validação de dados, reforçando a segurança da aplicação. A combinação de Node.js e Express não só favorece a eficiência técnica, mas também garante a manutenibilidade e escalabilidade do código-fonte. Sua estrutura modular permite a expansão da aplicação de forma coesa, facilitando futuras atualizações e adições de funcionalidades (NODE.JS, EXPRESS.JS, s.d., 2024; EXPRESS.JS, 2024).

Em resumo, a utilização da API Node.js com o Express representa uma abordagem moderna e eficaz para o desenvolvimento de aplicativos web, oferecendo um ambiente flexível e eficiente para sistemas escaláveis e de alta performance. Ambos os frameworks são licenciados sob a Licença MIT, uma licença de software livre amplamente utilizada em projetos de código aberto (EXPRESS.JS, 2024).

O Node.js com o Express.js foi utilizado neste TCC para desenvolver uma API que gerencia o processo de login e redirecionamento para a página inicial. A API foi estruturada para garantir segurança e eficiência, utilizando middlewares para autenticação e autorização. Além disso, implementei o mxGraph para a criação e manipulação de gráficos interativos, proporcionando uma interface dinâmica e intuitiva para o usuário

3.2 POSTGRESQL

O PostgreSQL oferece um servidor de banco de dados SQL (Structured Query Language) altamente eficiente, projetado para suportar ambientes multithread, multiusuário e resilientes. Especialmente otimizado para sistemas de produção com cargas intensas e missões críticas, o servidor PostgreSQL é também uma escolha adequada para integração em software distribuído em larga escala (POSTGRESQL, 2023). Além disso, o PostgreSQL é licenciado sob a Licença PostgreSQL, uma licença de código aberto que promove a gratuidade e o acesso aberto ao seu código-fonte.

Neste TCC, o banco de dados PostgreSQL foi utilizado para armazenar os modelos de diagrama de caso de uso gerados pela aplicação. A estrutura foi desenvolvida de forma a permitir o armazenamento eficiente e organizado dos dados, garantindo fácil recuperação e manipulação das informações. A modelagem das tabelas foi realizada com base em uma arquitetura relacional, possibilitando a criação de relacionamentos entre os elementos dos diagramas e outras entidades do sistema. A integração entre o PostgreSQL e a API desenvolvida com Node.js e Express.js assegurou um fluxo contínuo de dados entre o frontend e o backend, permitindo que os modelos fossem salvos, recuperados e atualizados em tempo real.

3.3 REACTJS

O ReactJS é uma biblioteca de JavaScript utilizada para a construção de interfaces de usuário interativas e reativas. Ele permite a criação de componentes reutilizáveis que gerenciam seu próprio estado e se integram facilmente com outros componentes, facilitando o desenvolvimento de interfaces complexas (REACTJS, 2024).

Neste TCC, a interface de usuário foi desenvolvida com ReactJS, permitindo a criação de componentes reutilizáveis para as telas de Login, Cadastro e Home. A autenticação foi implementada com base em tokens, garantindo segurança e controle de acesso. Na tela Home, foi integrado o mxGraph para a elaboração e manipulação de diagramas de caso de uso, permitindo que o usuário criasse e editasse diagramas de forma intuitiva e dinâmica. A comunicação eficiente entre o front-end em ReactJS e o

back-end em Node.js com Express.js assegurou a atualização em tempo real dos dados, armazenados no PostgreSQL.

3.4 STYLED-COMPONENTS

Na implementação deste projeto, foi utilizada a biblioteca styled-components no ambiente ReactJS para a estilização de componentes. O styled-components permite definir estilos diretamente no nível do componente, tornando a estilização mais modular e fácil de gerenciar, o que facilita a manutenção do código (STYLED-COMPONENTS, 2024).

Assim como o CSS descreve a apresentação de documentos HTML ou XML, o styled-components desempenha um papel semelhante, mas integrado ao ambiente ReactJS. Ele possibilita a definição de estilos específicos para cada componente, influenciando diretamente na apresentação visual do aplicativo, seja na tela, em papel ou em áudio, alinhando-se com as melhores práticas de desenvolvimento e design (MDN, 2022).

A utilização do styled-components neste TCC foi escolhida devido à sua capacidade de permitir a escrita de código CSS real diretamente nos componentes do ReactJS. Essa abordagem facilita a criação de estilos específicos para cada componente, promovendo uma maior organização e modularidade no código. Além disso, o styled-components oferece suporte a **CSS-in-JS**, o que possibilita o uso de variáveis, herança de estilos e reutilização de código, tornando o desenvolvimento mais eficiente e a manutenção mais simples. Essa estratégia permitiu uma estilização mais consistente e alinhada com as melhores práticas de desenvolvimento de interfaces modernas.

3.5 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 às pessoas surdas (BRASIL, 2022).

Neste TCC, o VLibras foi integrado para proporcionar acessibilidade ao conteúdo da aplicação por meio da tradução de textos para a Língua Brasileira de Sinais

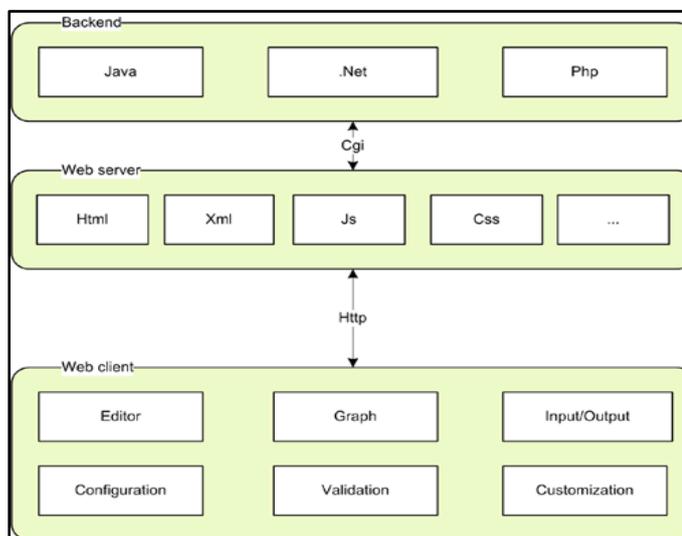
(Libras). Esta integração foi fundamental para tornar a plataforma mais inclusiva, permitindo que usuários surdos ou com deficiência auditiva acessassem informações de forma visual e interativa. A implementação foi realizada de forma a garantir que o conteúdo da aplicação fosse traduzido de maneira eficiente, oferecendo uma experiência de usuário acessível e facilitando a navegação por meio de sinais em tempo real.

3.6 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.

A principal vantagem desta tecnologia é que não há necessidade de nenhum plug-in de terceiros para seu funcionamento.

Figura 8 - ARQUITETURA DA BIBLIOTECA MXGRAPH.



Fonte: (MXGRAPH, 2020).

O mxGraph foi utilizado, neste TCC, para permitir a criação e manipulação de gráficos interativos na aplicação. A integração do mxGraph foi fundamental para proporcionar uma interface dinâmica e intuitiva, permitindo aos usuários desenharem e editarem diagramas de forma simples e eficiente. A implementação foi realizada de maneira a garantir que os diagramas fossem gerenciados diretamente no navegador, utilizando a arquitetura JavaScript da biblioteca. Isso proporcionou uma experiência interativa e visualmente rica, tornando o processo de criação de diagramas mais acessível e ágil.

4 TRABALHOS RELACIONADOS

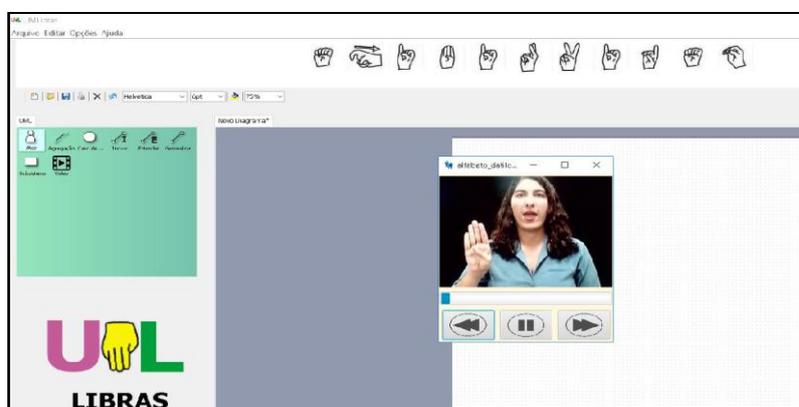
Neste capítulo é apresentado a primeira versão da ferramenta UMLibras Web, bem como uma listagem de ferramentas que implementam a UML existentes no mercado.

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 (FONSECA, 2018). O projeto foi desenvolvido pelo aluno Antônio José Barbosa Fonseca, do curso de Tecnologia em Análise e Desenvolvimento de Sistemas (TADS), sob orientação do professor M.Sc. Sergio Augusto C. Bezerra, ambos pertencentes ao Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM).

A ferramenta, mostrada na Figura 9, utiliza um algoritmo para converter palavras em um grupo de imagens, realizando a datilologia em Libras e proporcionando acessibilidade e interatividade para pessoas com deficiência auditiva no desenvolvimento do referido diagrama. Além disso, permite a associação de vídeos aos objetos, sejam eles em Libras ou não, bem como sua execução (FONSECA, 2018).

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 caso de uso, 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 Caso de Uso	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	<i>GPL</i>	Não
Poseidon	Sim	<i>Paga/Community</i>	Não
Diagrams	Sim	<i>Free</i>	Não
UMLibras desktop	Sim	<i>Free</i>	Sim
UMLibras web	Sim	<i>Free</i>	Sim

Fonte: (Próprio autor, 2023).

A Tabela 1 apresenta uma relação de ferramentas UML disponíveis no mercado, destacando informações como suporte a diagramas de caso de uso, tipo de licença e acessibilidade. Observa-se que todas as ferramentas listadas possuem suporte para a criação de diagramas de caso de uso. No entanto, a acessibilidade é um fator pouco contemplado nessas soluções. Das dez ferramentas analisadas, apenas o UMLibras Desktop e o UMLibras Web oferecem recursos voltados para acessibilidade, demonstrando uma lacuna significativa nesse aspecto. Além disso, percebe-se que há uma diversidade de licenciamento entre as ferramentas, com opções pagas, gratuitas e

de código aberto, proporcionando diferentes alternativas para os usuários conforme suas necessidades e restrições.

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 web.

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, 2023).

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, 2023).

5.2 PROJETO

Nesta fase, foram delineadas as especificações e estabelecida a estrutura da ferramenta desenvolvida, evidenciando a coesão lógica entre seus componentes. Foram abordados minuciosamente os procedimentos e caracterizada a interface da ferramenta, proporcionando uma visão abrangente e detalhada do seu funcionamento.

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.

Na construção da interface do usuário (*front-end*) foi utilizado o ReactJS para a organização estrutural e *style-components* para a estilização, ambos implementados com TypeScript. Essa escolha estratégica de tecnologias foi fundamental para garantir uma base sólida e eficiente, resultando em uma interface visualmente atraente e de alta qualidade.

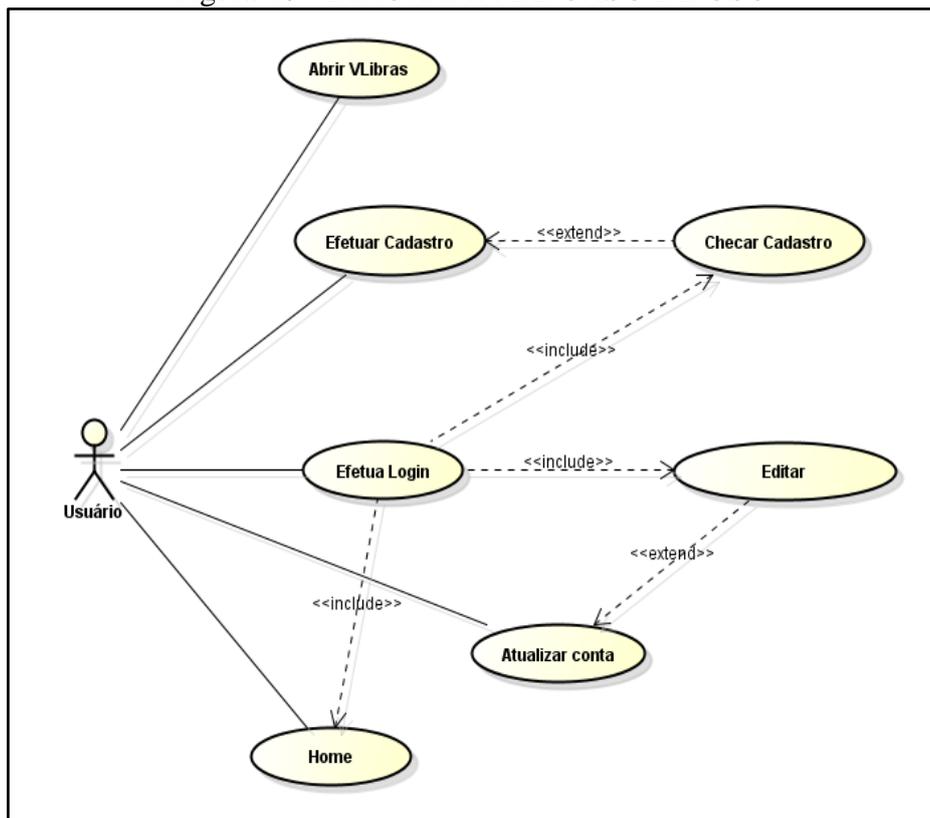
No desenvolvimento do *back-end* foram utilizados o Node.js e o TypeORM em conjunto com o PostgreSQL como Sistema de Gerenciamento de Banco de Dados (SGBD). Essa configuração permitiu o processamento eficiente das informações recebidas, resultando no envio dos resultados do processamento de volta para o navegador. Essa escolha de tecnologias no *back-end* proporcionou uma implementação sólida e eficaz para a lógica de negócios da aplicação.

Foi utilizado como base para o desenvolvimento a 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 oferece uma visualização das interações entre os usuários e as funcionalidades incorporadas em um sistema. Na Figura 10 é apresentado o diagrama de caso de uso específico da ferramenta desenvolvida, destacando o usuário do sistema por meio de uma representação simbólica. Este usuário possui diversas opções de interação com o sistema, retratadas por balões, abrangendo ações como a criação de um diagrama, a preservação do diagrama elaborado, a visualização de vídeos com informações em libras e a ativação do tradutor virtual VLibras. Essa representação esquemática proporciona uma compreensão visual das operações disponíveis ao usuário dentro do contexto da ferramenta.

Figura 10 - DIAGRAMA DE CASO DE USO.



Fonte: (Próprio autor, 2023).

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, 2023).

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, 2023).

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.

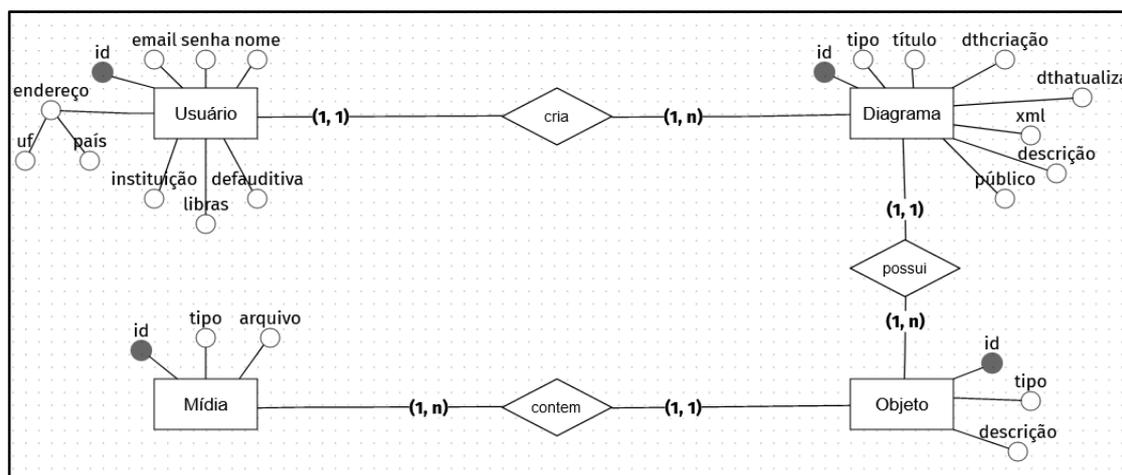
Fonte: (Próprio autor, 2023).

5.2.4 Modelo conceitual do banco de dados

Um modelo conceitual é uma representação do banco de dados independente de implementação em um Sistema Gerenciador de Banco de Dados (SGBD). Este modelo registra quais dados podem ser armazenados no banco de dados, mas não especifica como esses dados são armazenados no nível do SGBD. Uma abordagem amplamente utilizada para modelagem conceitual é a técnica Entidade-Relacionamento (ER) (HEUSER, 2009), que se materializa em um diagrama gráfico para representar os elementos definidos para o Banco de Dados (BD).

Ao aplicar esta técnica na construção do banco de dados relacional, foi possível descrever de maneira concisa todos os atributos essenciais necessários para o desenvolvimento do módulo da ferramenta, incluindo suas entidades e os relacionamentos entre elas. A Figura 11 apresenta o modelo Entidade-Relacionamento do banco de dados do módulo da ferramenta UMLibras, focalizado na elaboração de diagramas de Caso de Uso. Essa representação gráfica oferece uma visão clara e organizada das entidades e dos vínculos entre elas, destacando os elementos fundamentais para o funcionamento do sistema.

Figura 11 - MODELO ENTIDADE-RELACIONAMENTO DA FERRAMENTA UMLIBRAS.

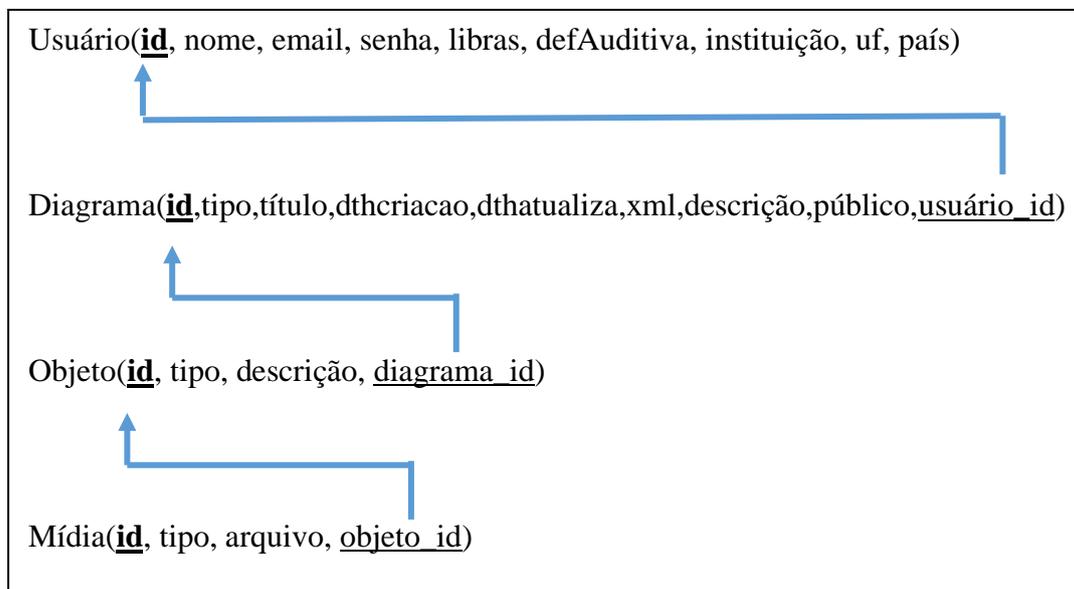


Fonte: (Próprio autor, 2023).

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.

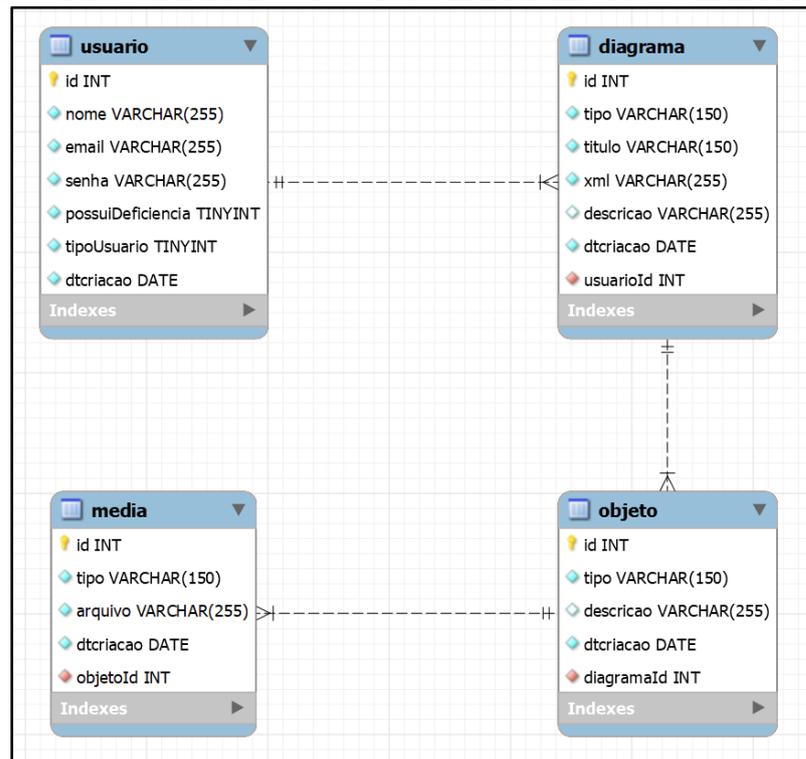
Figura 12 - MODELO LÓGICO (FORMA TEXTUAL) DA FERRAMENTA UMLIBRAS.



Fonte: (Próprio autor, 2023).

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

Figura 13 - DIAGRAMA LÓGICO DA FERRAMENTA UMLIBRAS.



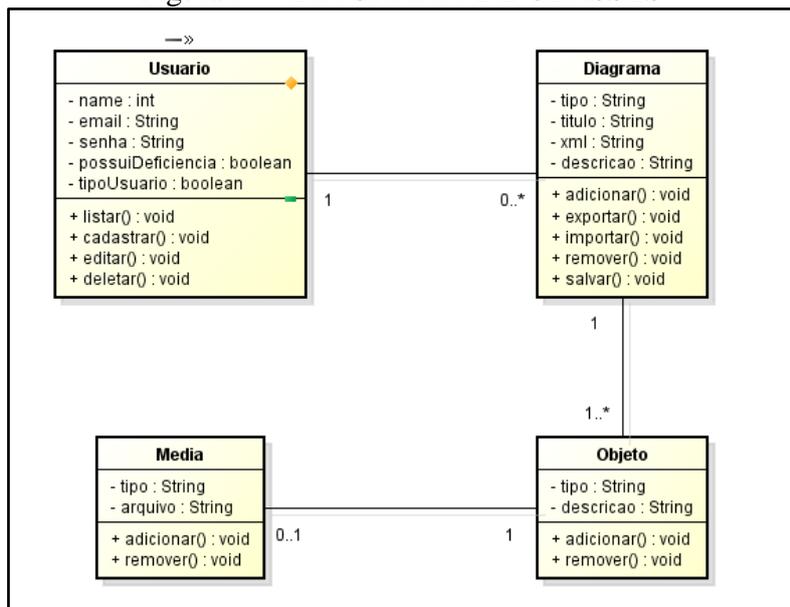
Fonte: (Próprio autor, 2023).

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.

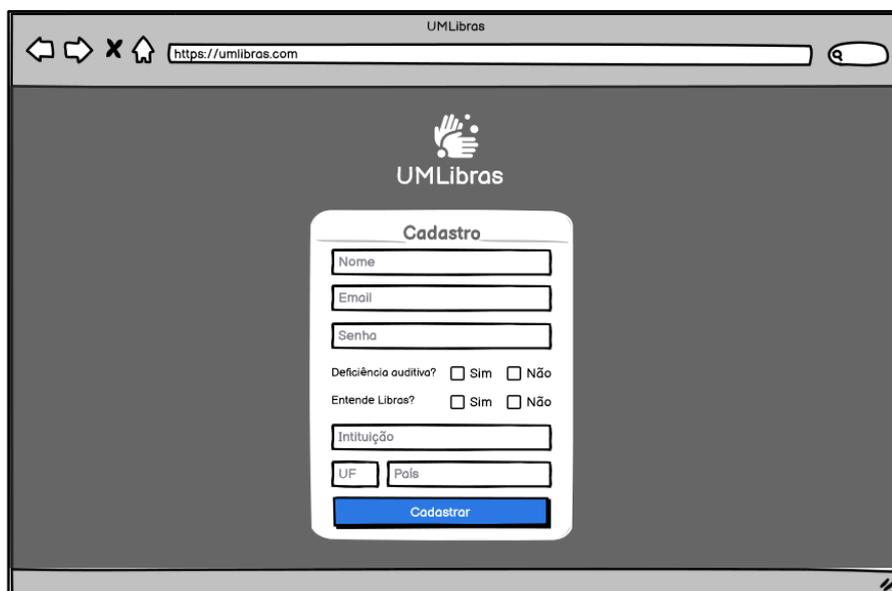


Fonte: (Próprio autor, 2023).

5.2.7 Interface

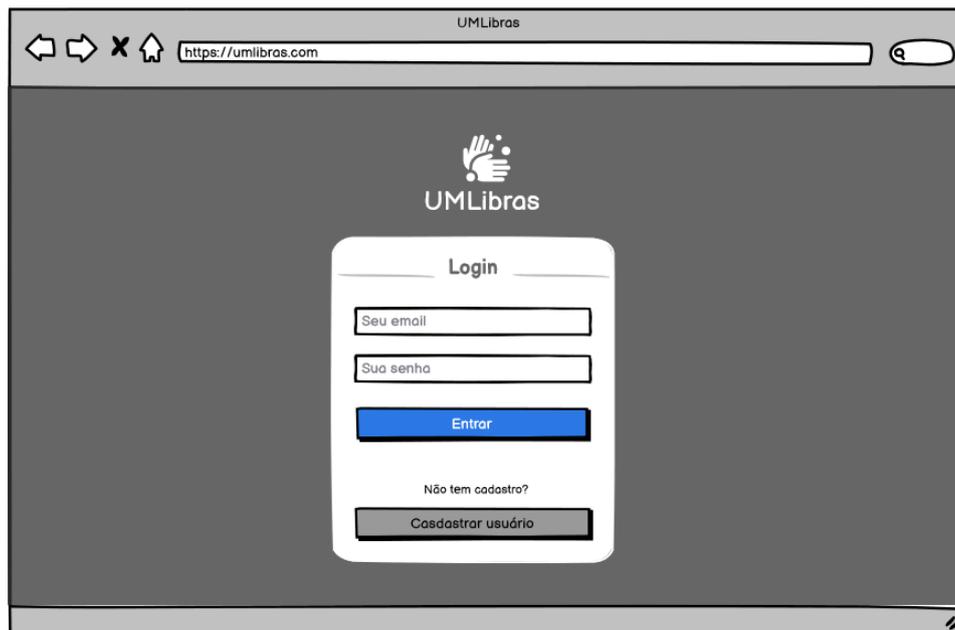
Essa seção demonstra os protótipos das telas da ferramenta UMLibras, onde foram desenhados utilizando a ferramenta de design de interface de usuário Balsamiq, disponível em balsamiq.com. As Figuras de 15 a 18 especificam as telas de Login, Cadastro de usuário, Principal e Gerenciar mídias.

Figura 15 - PROTÓTIPO DA TELA DE LOGIN.



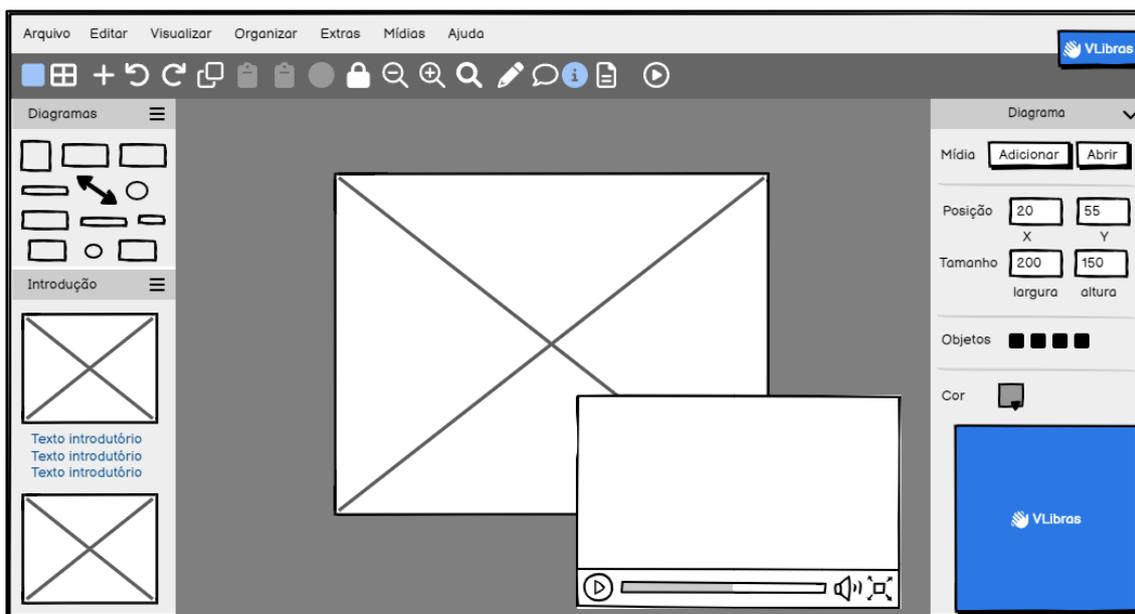
Fonte: (Próprio autor, 2023).

Figura 16 - PROTÓTIPO DA TELA DE CADASTRO DE USUÁRIO.



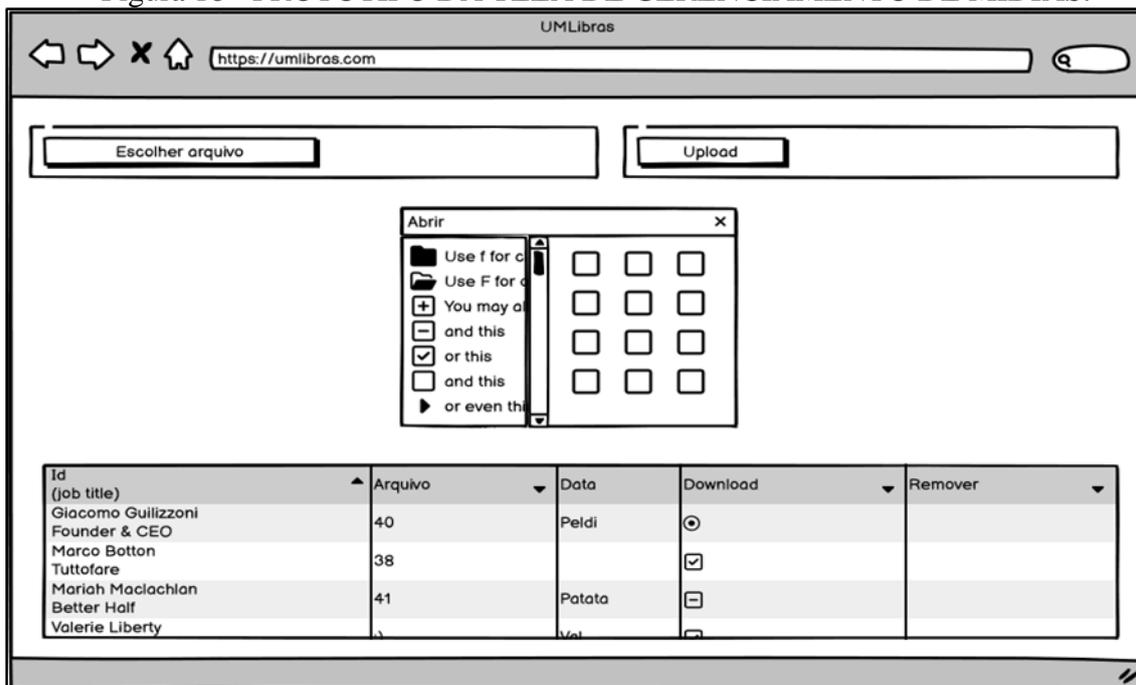
Fonte: (Próprio autor, 2023).

Figura 17 - PROTÓTIPO DA TELA PRINCIPAL DA FERRAMENTA UMLIBRAS.



Fonte: (Próprio autor, 2023).

Figura 18 - PROTÓTIPO DA TELA DE GERENCIAMENTO DE MÍDIAS.



Fonte: (Próprio autor, 2023).

5.3 CODIFICAÇÃO

A etapa de implementação engloba todas as atividades relacionadas ao desenvolvimento efetivo do sistema, dando vida ao modelo de *design* elaborado na fase anterior. Os diferentes componentes da aplicação são codificados e testados de maneira independente, assegurando a integridade interna de cada um (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 empregadas para a inserção de vértices e arestas, conforme ilustrado na Figura 21. A `endUpdate()` deve sempre ser incorporada em um bloco `finally` para garantir sua execução mesmo se `beginUpdate()` falhar. No entanto, `beginUpdate()` não deve ser incluída no bloco `try` para assegurar que `endUpdate()` não seja executada se `beginUpdate()` não for concluída com êxito. Essa precaução é essencial para manter o modelo em um estado consistente, garantindo que para cada chamada de `beginUpdate()`, haja precisamente uma chamada correspondente de `endUpdate()`.

A seção dentro do bloco `try` é responsável por criar os vértices e arestas no gráfico. O pai padrão é obtido do gráfico e geralmente corresponde ao primeiro filho da célula raiz no modelo, gerado automaticamente ao utilizar 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).

A utilização dos métodos `beginUpdate()` e `endUpdate()` não apenas aprimora o desempenho da visualização como também desempenha o papel de delimitar as alterações passíveis de serem desfeitas ao empregar a funcionalidade de desfazer/refazer.

5.3.4 Modelo

A função `mxCell()` define os elementos do modelo gráfico, que são implementados 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 funcionalidades de salvar, abrir, readGraphModel e writeGraphModel implementam um mecanismo padrão para a manipulação de arquivos no mxEditor. A função padrão de mxEditor.save é invocada com um argumento indicando se o salvamento foi iniciado pelo usuário ou pelo sistema. Posteriormente, ela utiliza a variável urlPost do objeto do editor para verificar se uma requisição POST deve ser enviada. Caso essa variável esteja definida, o editor dispara uma requisição POST para a URL especificada, transmitindo o XML como uma variável POST denominada xml.

A Figura 28 apresenta um exemplo de arquivo PHP localizado no mesmo diretório da página HTML. Se o nome do arquivo for server.php, a variável urlPost no editor deve ser configurada como server.php. O arquivo PHP recupera o XML da solicitação POST e o armazena em um arquivo denominado 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, 2023).

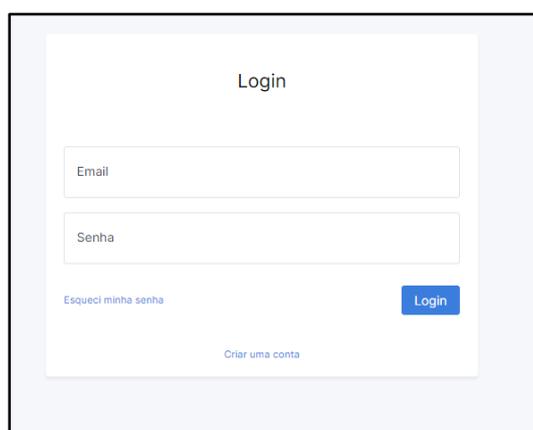
5.4 IMPLEMENTAÇÃO

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

5.4.1 Tela de login

Para acessar a ferramenta, um utilizador necessita estar cadastrado com um usuário e senha. A Figura 31 ilustra a tela de login da ferramenta UMLibras Web.

Figura 31 - TELA DE LOGIN DA FERRAMENTA UMLIBRAS.



A imagem mostra a interface de login da ferramenta UMLibras Web. O formulário é centralizado e contém os seguintes elementos: o título "Login" no topo; um campo de entrada para "Email"; um campo de entrada para "Senha"; um link "Esqueci minha senha" em azul; um botão "Login" em azul; e um link "Criar uma conta" em azul na base do formulário.

Fonte: (Próprio autor, 2023).

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 32.

Figura 32 - TELA DE CADASTRO DE USUÁRIO.



A imagem mostra a interface de cadastro de usuário da ferramenta UMLibras Web. O formulário é centralizado e contém os seguintes elementos: o título "Criar uma conta" no topo; um campo de entrada para "Nome Completo"; um campo de entrada para "Email"; dois campos de entrada para "Senha" e "Confirmar senha"; duas opções de radio button para "Possui Deficiência (Sim)" e "Possui Deficiência (Não)"; duas opções de radio button para "Aluno" e "Professor"; e um botão "Criar conta" em azul na base do formulário.

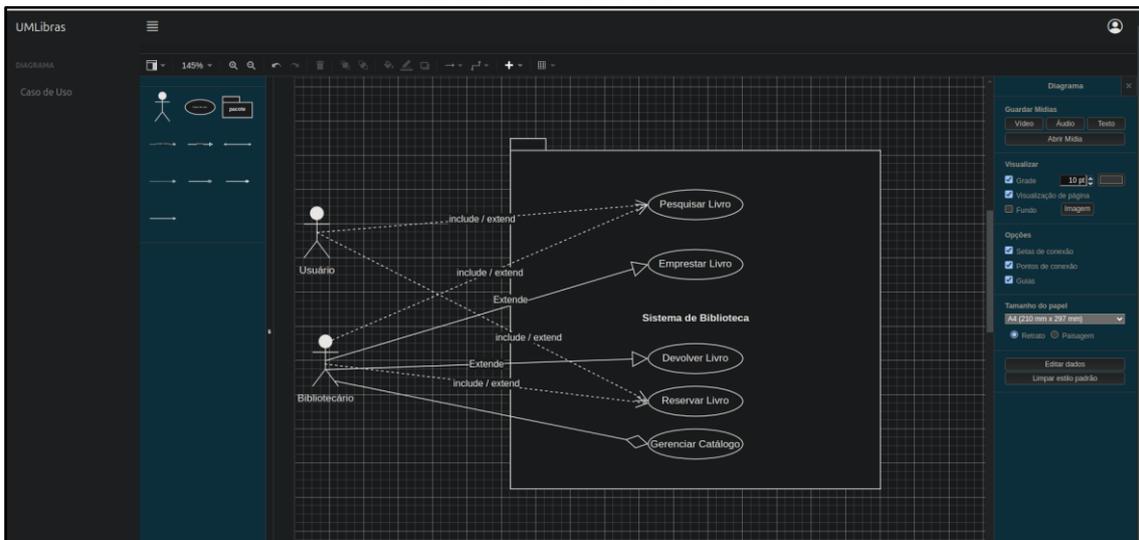
Fonte: (Próprio autor, 2023).

5.4.2 Tela principal

Na interface principal da ferramenta, representada na Figura 33, o usuário pode acionar o caso de uso "Criar Novo Diagrama" utilizando os elementos disponíveis na paleta denominada "Diagrama de Caso de Uso," conforme ilustrado na Figura 34. Além disso, é possível executar o caso de uso "Abrir Diagrama Existente," "Salvar Diagrama Atual," "Imprimir Diagrama," "Configurar," "Organizar Elementos" e "Adicionar Mídias" por meio dos menus e botões presentes nas barras de ferramentas horizontais, como exemplificado na Figura 35.

A interface também suporta os casos de uso "Exportar Diagrama em Formato XML" e "Importar Diagrama em Formato XML," assim como o caso de uso "Gerar Arquivos de Imagem" nos formatos PNG, JPEG, SVG, GIF e PDF, conforme ilustrado na Figura 37. Essa abordagem proporciona ao usuário uma variedade de casos de uso para interagir com a ferramenta, adaptando-se às diversas necessidades durante o processo de criação e manipulação de diagramas de Caso de Uso.

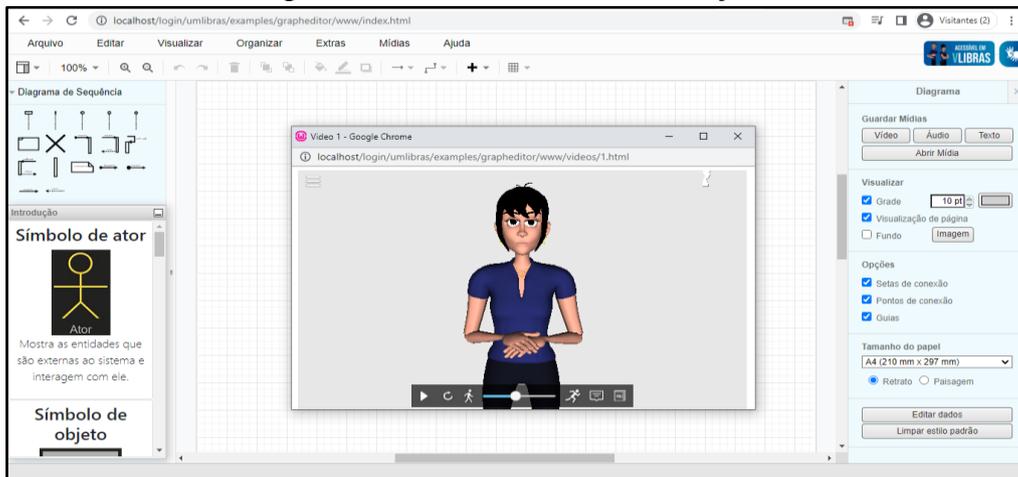
Figura 33 - TELA PRINCIPAL COM UM DIAGRAMA DE CASO DE USO E UM EXEMPLO PRÁTICO.



Fonte: (Próprio autor, 2023).

intuitiva de ajustar as configurações e propriedades do diagrama, melhorando a experiência de edição e personalização.

Figura 36 - VÍDEO DE INTRODUÇÃO.



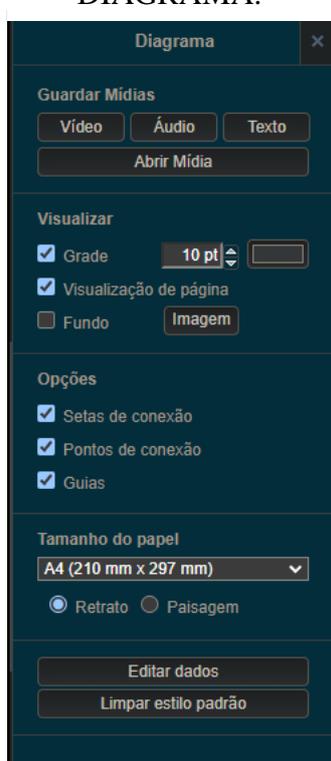
Fonte: (Próprio autor, 2023).

Figura 37 - JANELA DE INTRODUÇÃO.



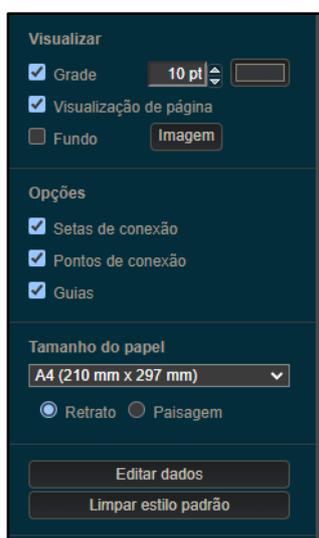
Fonte: (Próprio autor, 2023).

Figura 38 - PALHETA DIAGRAMA.



Fonte: (Próprio autor, 2023).

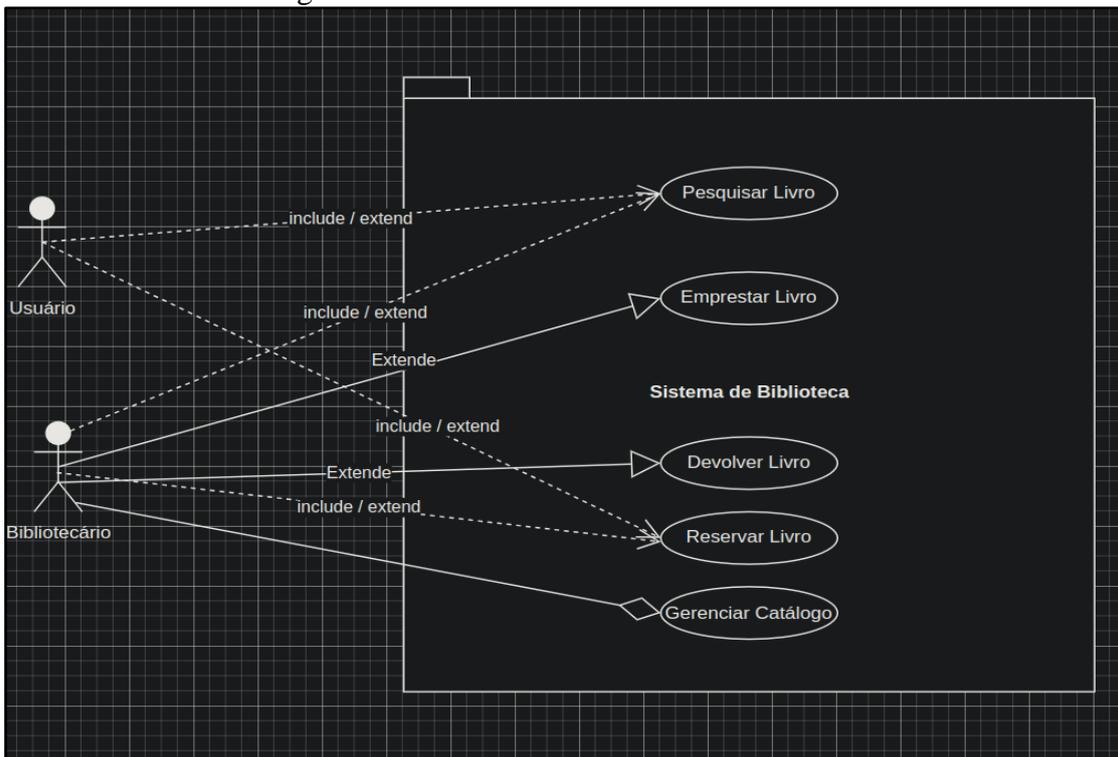
Figura 39 - PROPRIEDADES DE UM ELEMENTO SELECIONADO.



Fonte: (Próprio autor, 2023).

A Figura 40 apresenta um exemplo de diagrama de caso de uso, modelado e gerado diretamente na ferramenta UMLibras Web, ilustrando uma parte do sistema.

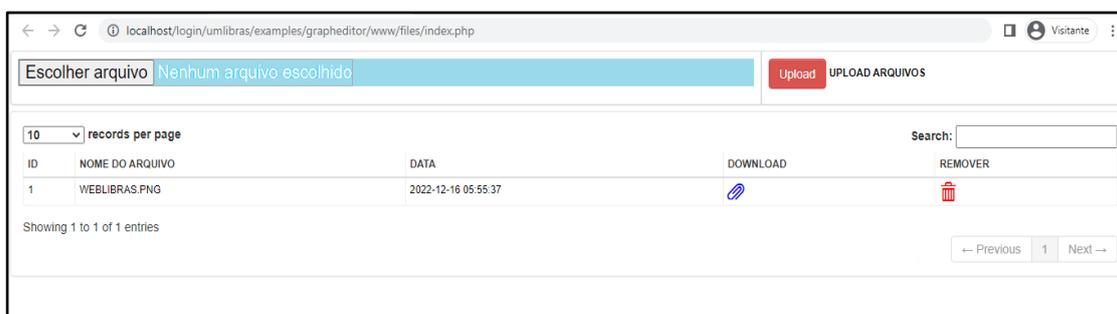
Figura 40 - DIAGRAMA DE CASO DE USO.



Fonte: (Próprio autor, 2023).

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 41 - TELA DE GERENCIAMENTO DE MÍDIAS, IMPLEMENTAÇÃO FUTURA.



Fonte: (Próprio autor, 2023).

6 CONSIDERAÇÕES FINAIS

Considerando a mudança de ênfase para diagramas de caso de uso, observou-se que o cenário atual do mercado oferece diversas ferramentas de tecnologia assistiva. No entanto, essa variedade ainda não atende completamente às demandas específicas da comunidade surda. Ferramentas renomadas, como VLibras, Rybená, Hand Talk, ProDeaf, WebLibras, &Ava, Primeira Mão e Transcrição Instantânea, desempenham um papel crucial ao traduzir textos e áudios para Libras, facilitando a comunicação e compreensão de informações por pessoas com deficiência auditiva ou surdez.

Identificou-se lacunas significativas ao abordar ferramentas voltadas para a tecnologia UML, pois as aplicações disponíveis no mercado até o momento não atendem adequadamente às necessidades de acessibilidade, dificultando a utilização por pessoas com deficiência auditiva.

Este trabalho concentrou-se no desenvolvimento do módulo de Caso de Uso da UML, integrado à plataforma computacional UMLibras Web. Sob a liderança do professor M.Sc. Sergio Augusto Coelho Bezerra do IFAM e sua coordenação, alunos do Curso de Análise e Desenvolvimento de Sistemas do Departamento Acadêmico de Informação e Comunicação do Campus Manaus Centro contribuíram para potencializar a ferramenta, visando contemplar outros diagramas da UML no futuro.

É importante destacar a importância do uso do modelo RAD para o desenvolvimento do novo módulo da UMLibras Web. Isso porque as regras de negócio e requisitos foram implementados de forma incremental, permitindo validações constantes e garantindo que o produto atendesse às necessidades estabelecidas. Esta abordagem mostrou-se mais eficiente para um desenvolvimento individual, onde o mesmo profissional é responsável por todas as etapas do processo.

A ferramenta desenvolvida atende aos requisitos fundamentais de acessibilidade, proporcionando uma interface simples e intuitiva. Essa abordagem facilita não apenas a utilização da tecnologia UML em relação ao diagrama de caso de uso, mas também contribui para a inclusão efetiva de pessoas com deficiência auditiva, tornando o aprendizado mais acessível e integrado.

6.1 TRABALHOS FUTUROS

Foram identificados os seguintes trabalhos futuros que poderão tornar a ferramenta UMLibras web mais robusta:

- Realizar testes de usabilidade com usuários, abrangendo tanto usuários com ou sem deficiência auditiva. Essa abordagem visa assegurar que a ferramenta seja acessível e eficaz para um público diversificado.
- Aplicar um questionário de avaliação junto aos usuários participantes dos testes. O questionário teria como propósito verificar a adequação da ferramenta às atividades propostas pelo professor, bem como avaliar se a interface favorece o aprendizado do aluno de maneira efetiva.
- Prosseguir com o desenvolvimento de outros módulos na ferramenta UMLibras, visando ampliar seu escopo para contemplar os demais diagramas da UML. Essa expansão contribuirá para uma experiência mais abrangente e completa no uso da ferramenta.

REFERÊNCIAS BIBLIOGRÁFICAS

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

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. *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/2002/110436.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/113146.htm. Acesso em: 2 maio. 2022.

BRASIL. Decreto nº 7.612, de 17 de novembro de 2011. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/decreto/d7612.htm. Acesso em: 12 fev. 2024.

BRASIL. Ministério da Economia. *VLibras*. Disponível em: <https://www.gov.br/governodigital/pt-br/vlibras>. Acesso em: 12 fev. 2024.

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 *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.

IBGE. *Censo Demográfico 2022: Características Gerais da População*. Disponível em: <https://censo2022.ibge.gov.br/>. Acesso em: 12 fev. 2024.

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: 20 out. 2023.

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

NODE.JS. (n.d.). *Node.js®: JavaScript runtime built on Chrome's V8 JavaScript engine*. Disponível em: <https://nodejs.org/>. Acesso em: 12 fev. 2024.

PACHECO, Dalmir. *Políticas públicas como promotoras da visibilidade social das pessoas com deficiência: estudo de caso do Projeto Curupira*. 2015. Tese (Doutorado em Educação) – Universidade Federal do Amazonas, Manaus, 2014. Disponível em: <https://tede.ufam.edu.br/bitstream/tede/4131/2/Tese%20-%20Dalmir%20Pacheco%20de%20Souza.pdf>.

PACHECO, Dalmir. *Deficiência e Tecnologia Assistiva: Projetos e Pesquisas*. Instituto Federal de Educação, Ciência e Tecnologia do Amazonas (IFAM). Disponível em: <https://integra.ifam.edu.br/portfolio/pessoas/dalmir-pacheco-de-souza/defici%C3%Aancia>. Acesso em: 12 fev. 2024.

PMI. *GUIA PMBOK - Guia do conhecimento em gerenciamento de projetos*. 6. ed. Project Management Institute, 2017.

POSTGRESQL. *PostgreSQL 17devel Documentation - General Information*. Disponível em: <https://www.postgresql.org/docs/devel/index.html>. Acesso em: 20 out. 2023.

PRESSMAN, Roger S. *Engenharia de Software: uma abordagem profissional*. 7. ed. São Paulo: McGraw Hill, 2011.

RADIXWEB. *radixweb – Desenvolvimento rápido de aplicativos: um guia definitivo para 2025*. Disponível em: <https://radixweb.com/blog/introduction-to-rapid-application-development>. Acesso em: 12 fev. 2024.

REACTJS. *React – A JavaScript library for building user interfaces*. Disponível em: <https://react.dev/>. Acesso em: 12 fev. 2024.

RYBENA. *Solução que converte textos em Libras e voz*. Disponível em: <http://www.rybena.com.br/>. Acesso em: 12 fev. 2024.

SILVA, A.; VIDEIRA, C. *UML, Metodologias e Ferramentas CASE*. Porto: Centro Atlântico, 2011.

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

STYLED-COMPONENTS. *Styled-components – Visual primitives for the component age*. Disponível em: <https://styled-components.com/>. Acesso em: 12 fev. 2024.

STYLED-COMPONENTS. *styled-components*. Disponível em: <https://styled-components.com/docs>. Acesso em: 20 out. 2023.

VLibras. *Ferramenta governamental de código aberto que realiza tradução automática para Libras*. Disponível em: <https://www.vlibras.gov.br/>. Acesso em: 12 fev. 2024.

WEBLIBRAS. *Plataforma web para tradução e comunicação em Libras*. Disponível em: <https://www.weblibras.gov.br/>. Acesso em: 12 fev. 2024.

APÊNDICE A

Neste apêndice, são descritos os casos de usos que complementam os casos de uso demonstrados na Seção 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</p>

	<p>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).