

ETEC EURO ALBINO DE SOUZA

**Ensino Técnico Integrado ao Médio de Desenvolvimento de
Sistemas**

Giovanna Gabrielle Cirino dos Santos

Isabela Leme Cruz

Kimberly Moura Batista

Laura Gonçalves Flausino

Mayra Vitória Carlos

Miguel Martins Maia

Náthali Pasquim Ribeiro

Patrick Heloi da Silva Júnior

SCAN HANDS: Sistema para o Aprendizado de LIBRAS

Mogi Guaçu-SP

2025

Giovanna Gabrielle Cirino dos Santos

Isabela Leme Cruz

Kimberly Moura Batista

Laura Gonçalves Flausino

Mayra Vitória Carlos

Miguel Martins Maia

Náthali Pasquim Ribeiro

Patrick Heloi da Silva Júnior

SCAN HANDS: Sistema para o Aprendizado de LIBRAS

Trabalho de Conclusão de Curso apresentado ao Curso Ensino Técnico integrado ao Médio em Desenvolvimento de Sistema da Etec Euro Albino de Souza, orientado pelo professor Maércio Girardi Bisco e Pedro Ramires da Silva Amalfi Costa, como requisito parcial para a obtenção do título de Técnico em Desenvolvimento de Sistemas

Mogi Guaçu-SP

2025

S237s

Santos, Giovanna Gabrielle Cirino dos; Cruz, Isabela Leme; Batista, Kimberly Moura; Flausino, Laura Gonçalves; Carlos, Mayra Vitória; Maia, Miguel Martins; Ribeiro, Náthali Pasquim; Silva Júnior, Patrick Heloi da.

SCAN HANDS: Sistema para o Aprendizado de LIBRAS – Mogi Guaçu – SP. 2025.

298 p.

Trabalho de Conclusão do Curso de Ensino Médio com Habilitação Profissional de Técnico em Desenvolvimento de Sistemas da ETEC Euro Albino de Souza de Mogi Guaçu.

Orientadores: Professor Especialista Maércio Girardi Bisco e Professor Mestre Pedro Ramires da Silva Amalfi Costa

1. Acesso. 2. Inclusão. 3. Educação. I. Bisco, Maércio Girardi; Costa, Pedro Ramires da Silva Amalfi (orientadores) II. Título.

CDD 303.32

DEDICATÓRIA

Dedicamos nosso trabalho a ETEC e ao nosso professor João Marcelo, por ter nos dado o conhecimento e oportunidade de mostrarmos do que somos capazes.

AGRADECIMENTO

Agradecemos, em primeiro lugar, a Deus e aos professores, que estiveram ao nosso lado, oferecendo suporte e orientação ao longo do projeto e do curso. Também agradecemos de coração aos amigos e à nossa família, que nos acompanharam e nos deram força durante todo esse percurso.

EPÍGRAFE

“Temos que continuar aprendendo. Temos que estar abertos. E temos que estar prontos para espalhar nosso conhecimento a fim de chegar a uma compreensão mais elevada da realidade.”

Thich Nhat Hanh

RESUMO

SCAN HANDS é um TCC (Trabalho de Conclusão de Curso) que tem como objetivo permitir que os usuários possam praticar e aperfeiçoar suas habilidades em Libras de maneira intuitiva e interativa, também possibilitando a comunicação com professores. A plataforma permite a realização de atividades didáticas, com recursos que incluem o uso da câmera do dispositivo para reconhecimento de gestos, proporcionando *feedback* em tempo real sobre a execução correta dos sinais. Com isso, é solucionado o problema da dificuldade em encontrar e se comunicar com professores de libras, além de incentivar o aumento da acessibilidade criando um ambiente mais interativo e de fácil acesso a todos os surdos e ouvintes que se interessarem pela linguagem.

Palavras-Chaves: Libras; Inclusão; Linguagem; Aprendizado; Educação.

ABSTRACT

SCAN HANDS is a TCC (Course Completion Project) that aims to allow users to practice and improve their skills in Libras in an intuitive and interactive way, also enabling communication with teachers. The platform allows for educational activities, with resources that include the use of the device's camera for gesture recognition, providing real-time feedback on the correct execution of signs. With this, we solve the problem of difficulty in finding and communicating with Libras teachers, in addition to encouraging increased accessibility by creating a more interactive and easily accessible environment for all deaf and hearing people who are interested in the language.

Keywords: *Libras; Inclusion; Language; Learning; Education.*

LISTA DE FIGURAS

Figura 1 - HTML	28
Figura 2 - PHP.....	34
Figura 3 - Javascript.....	37
Figura 4 - Xampp.....	38
Figura 5 - MariaDB	40
Figura 6 - BRModelo	42
Figura 7 - Python.....	45
Figura 8 - MediaPipe	46
Figura 9 - LangFlow	49
Figura 10 - Exemplo de Imagem Digital	52
Figura 11 - Exemplo de Imagem Vetorial	53
Figura 12- Gráfico de respostas para “Qual a sua idade?”.....	87
Figura 13 - Gráfico de respostas para “Você já teve contato com a Língua Brasileira de Sinais (Libras)?”	88
Figura 14 – Gráfico de respostas para “Você acredita que Libras deveria ser ensinada nas escolas de forma obrigatória?”	89
Figura 15 - Gráfico de respostas para “Você estaria disposto(a) a aprender Libras se houvesse um aplicativo acessível para você?”	90
Figura 16 - Gráfico de respostas para “Você já enfrentou dificuldades ao tentar se comunicar com uma pessoa surda?”	91
Figura 17 – Gráfico de respostas para “Você conhece alguma iniciativa ou projeto que promove o ensino de Libras na sua cidade ou comunidade?”	92
Figura 18 - Tela de <i>Login</i>	94
Figura 19 - Tela inicial do aluno	96
Figura 20 - Tela de visualizar perfil do aluno.....	97
Figura 21 - Tela aulas do aluno.....	98
Figura 22 - Tela atividades do aluno	99
Figura 23 - Tela <i>chat</i>	100
Figura 24 - Tela inicial do professor	102
Figura 25 - Tela de visualizar perfil do Professor	103
Figura 26 - Tela aulas do professor.....	104
Figura 27 - Tela de criar aulas do Professor	105
Figura 28 – Tela de alterar e excluir aulas do Professor	106

Figura 29 - Tela criar atividade do professor	107
Figura 30 - Tela de excluir aulas do Professor	108
Figura 31 - Tela criar atividade do professor	109
Figura 32 - Tela de alterar e excluir atividades do Professor	110
Figura 33 - Tela de alterar atividades do Professor.....	111
Figura 34 - Tela de excluir atividades do Professor	112
Figura 35 - Tela de turmas do Professor.....	113
Figura 36 - Tela inicial da secretaria	114
Figura 37 - Tela aluno da secretaria.....	115
Figura 38 - Tela aluno da secretaria.....	116
Figura 39 - Tela professor da secretaria.....	117
Figura 40 – Tela secretaria turmas.....	118
Figura 41 – Tela filtro.....	119
Figura 42 – Tela barra de pesquisa.....	120
Figura 43 – Tela de adicionar aluno secretaria	121
Figura 44 – Tela de adicionar professor secretaria	122
Figura 45 – Tela de adicionar turma.....	123
Figura 46 – Tela de alterar aluno	124
Figura 47 – Tela de alterar professor	125
Figura 48 – Tela de alterar turma	126
Figura 49 – Tela de excluir aluno	127
Figura 50 – Tela de excluir professor	128
Figura 51 - Tela de excluir turma.....	129
Figura 52 - Tela BPMN secretaria	160
Figura 53 – Tela BPMN professor	162
Figura 54 - Tela do BPMN aluno.....	164
Figura 55 - Tela de diagrama de classe	170
Figura 56 - Tela de Diagrama de caso de uso	172
Figura 57 - Tela de <i>login</i>	174
Figura 58 - Tela de códigos do <i>login</i>	175
Figura 59 - Tela de redefinição de Senha	177
Figura 60 - Tela de códigos da redefinição de Senha	178
Figura 61 - Tela inicial do aluno	179
Figura 62 - Tela de código home do aluno	180

Figura 63 – Tela visualizar perfil.....	181
Figura 64 – Tela de Código visualizar perfil aluno.....	182
Figura 65 - Tela de aulas do aluno.....	183
Figura 66 – Tela de Código aulas aluno.....	184
Figura 67 – Tela filtrar aulas aluno.....	185
Figura 68 – Tela de Código filtrar aulas aluno.....	186
Figura 69 – Tela de Aula do aluno.....	187
Figura 70 – Tela de Código aula do aluno.....	188
Figura 71 – Tela atividades do aluno.....	189
Figura 72 – Tela Código atividades do aluno.....	190
Figura 73 – Tela da atividade sendo executada do aluno.....	191
Figura 74 – Tela de Código atividade sendo executada do aluno.....	192
Figura 75 - Tela inicial do Professor.....	193
Figura 76 – Tela Código tela inicial do Professor.....	194
Figura 77 - Tela de visualizar perfil do Professor.....	195
Figura 78 – Tela Código visualizar perfil do professor.....	196
Figura 79 - Tela de aulas do Professor.....	197
Figura 80 – Tela Código aulas do Professor.....	198
Figura 81 - Tela aula selecionada do professor.....	199
Figura 82 – Tela de Código aula selecionada do professor.....	200
Figura 83 - Tela criar aulas do Professor.....	201
Figura 84 – Tela de código de criar aulas do Professor.....	202
Figura 85 – Tela alterar e excluir aulas do professor.....	203
Figura 86 – Tela do Código alterar e excluir aulas do professor.....	204
Figura 87 - Tela do código de alterar aulas do Professor.....	205
Figura 88 – Tela de código aula do professor.....	206
Figura 89 - Tela de excluir aulas do Professor.....	208
Figura 90 - Tela do Código excluir aulas do Professor.....	208
Figura 91 - Tela de atividades do Professor.....	209
Figura 92 – Tela de código da Atividades do Professor.....	210
Figura 93 - Tela atividade selecionada do professor.....	211
Figura 94 –Tela de código atividade selecionada do professor.....	212
Figura 95 - Tela de filtrar atividades.....	213
Figura 96 – Tela de código de filtrar atividades.....	214

Figura 97 - Tela de Alterar atividades do Professor	215
Figura 98 – Tela de código de alterar atividades do Professor	216
Figura 99 - Tela de Excluir atividades do Professor	217
Figura 100 – Tela de código de excluir atividades do Professor	218
Figura 101 - Tela criar Atividades do Professor	219
Figura 102 – Tela de código de criar Atividades do Professor	220
Figura 103 - Tela de turmas do Professor	222
Figura 104 – Tela de código de turmas do Professor.....	223
Figura 105 - Tela turma selecionada do professor	224
Figura 106 - Tela de código da turma selecionada do professor.....	225
Figura 107 - Tela turma membros do professor	227
Figura 108 – Tela de código da turma membros do professor	228
Figura 109 – tela inicial da secretaria.....	229
Figura 110 – Tela de código da tela inicial da secretaria.....	230
Figura 111 – Tela visualizar perfil da secretaria	231
Figura 112 - Tela do código visualizar perfil da secretaria	232
Figura 113 – Tela alunos da secretaria	233
Figura 114 – Tela de código alunos da secretari.....	234
Figura 115 – Tela professores da secretaria.....	236
Figura 116 – Tela do código professores da secretaria	237
Figura 117 – Tela de turmas da secretaria.....	239
Figura 118 – Tela de código das turmas da secretaria	240
Figura 119 – Tela de filtrar alunos da secretaria	242
Figura 120 – Tela de código de filtrar alunos da secretaria.....	243
Figura 121 – Tela de pesquisar alunos da secretaria.....	245
Figura 122 – Tela de código de pesquisar alunos da secretaria	246
Figura 123 – Tela de adicionar aluno da secretaria	247
Figura 124 – Tela de código adicionar aluno da secretaria.....	248
Figura 125 – Tela de adicionar professor da secretaria	250
Figura 126 – Tela de código adicionar professor da secretaria.....	251
Figura 127 – Tela de adicionar turma da secretaria	253
Figura 128 – Tela de código adicionar turma da secretaria.....	254
Figura 129 – Tela alterar aluno da secretaria.....	256
Figura 130 – Tela de código alterar aluno da secretaria	257

Figura 131 – Tela alterar professor da secretaria.....	259
Figura 132 – Tela de código alterar professor da secretaria	261
Figura 133 – tela alterar turma da secretaria.....	263
Figura 134 – Tela de código alterar professor da secretaria	264
Figura 135 – Tela de excluir aluno da secretaria.....	266
Figura 136 – Tela do código da tela excluir aluno da secretaria	267
Figura 137 – Tela de excluir professor da secretaria	268
Figura 138 – Tela do código de excluir professor da secretaria	269
Figura 139 - Tela de excluir aluno	271
Figura 140 – Tela do código de excluir turma da secretaria.....	272
Figura 141 – Tela da câmera em funcionamento	273
Figura 142 – Tela do código do funcionamento da câmera (parte 1)	275
Figura 143 – Tela do código do funcionamento da câmera (parte 2)	277
Figura 144 – Tela do código do funcionamento da câmera (parte 3)	278
Figura 145 - Tela do código do funcionamento da câmera (parte 4).....	280

LISTA DE TABELAS

Tabela 1- Cronograma Documentação	65
Tabela 2 – Cronograma <i>Design</i>	66
Tabela 3 – Cronograma Programação - Python.....	68
Tabela 4 – Cronograma Programação – <i>Machine Learning</i>	69
Tabela 5 – Cronograma Programação – <i>Front-End</i>	70
Tabela 6 - Cronograma Programação – Lang Flow	71
Tabela 7 - Cronograma Programação – Banco de Dados.....	72
Tabela 8 - Cronograma Geral.....	73
Tabela 9 - Economia	75

LISTA DE ABREVIATURAS E SIGLAS

- AJAX – *Asynchronous JavaScript + XML* (JavaScript e XML Assíncronos)
- API – *Application Programming Interface* (Interface de Programação de Aplicações)
- AVI – *Audio Video Interleave* (Intercalação de áudio e vídeo)
- ASL – *American Sign Language* (Língua de Sinais Americana)
- AWS – *Amazon Web Services* (Serviços Web da Amazon)
- AWS EC2 – *Amazon Elastic Compute Cloud* (Nuvem de Computação Elástica da Amazon)
- BPMN – *Business Process Model and Notation* (Modelo e Notação de Processos de Negócio)
- CCD – *Charge-Coupled Device* (Dispositivo de Carga Acoplada)
- CDC – Código de Defesa do Consumidor
- CLT – Consolidação das Leis do Trabalho
- CERN – *Conseil Européen pour la Recherche Nucléaire* (Organização Europeia para a Pesquisa Nuclear)
- CSS – *Cascading Style Sheets* (Folhas de Estilo em Cascata)
- CPS – Centro Paula Souza
- CORBA – *Common Object Request Broker Architecture* (Arquitetura Comum do Agente de Solicitação de Objetos)
- CUDA – *Compute Unified Device Architecture* (Arquitetura Unificada de Dispositivos de Computação)
- DDS – *Data Distribution Service* (Serviço de Distribuição de Dados)
- DNA – *Deoxyribonucleic Acid* (Ácido Desoxirribonucleico)
- DOM – *Document Object Model* (Modelo de Objeto de Documento)
- DNS – *Domain Name System* (Sistema de Nomes de Domínio)
- ETEC – Escola Técnica Estadual

ETESP – Escola Técnica Estadual de São Paulo

EPS – *Encapsulated PostScript* (*PostScript* Encapsulado)

FATEC – Faculdade de Tecnologia do Estado de São Paulo

FK – *Foreign Key* (Chave Estrangeira)

GEGL – *Generic Graphics Library* (Biblioteca de Gráficos Genéricos)

GIF – *Graphical Interchange Format* (Formato de Intercâmbio de Gráficos)

NVDA- *NonVisual Desktop Access* (Acesso à Área de Trabalho Não Visual)

GIMP – *GNU Image Manipulation Program* (Programa de Manipulação de Imagens GNU)

GUI – *Graphical User Interface* (Interface Gráfica do Usuário)

GPT – *Generative Pre-Trained Transformer* (Transformadores Generativos Pré-Treinados)

GPU – *Graphics Processing Unit* (Unidade de Processamento Gráfico)

HDR – *High Dynamic Range* (Alta Faixa Dinâmica)

HiDPI – *High Dots Per Inch* (Telas de Alta Densidade de *Pixels*)

HTML – *HyperText Markup Language* (Linguagem de Marcação de Hipertexto)

HTTP – *Hypertext Transfer Protocol* (Protocolo de Transferência de Hipertexto)

IA – *Intelligence Artificial* (Inteligência Artificial)

IBM – *International Business Machines Corporation* (Corporação Internacional de Máquinas Comerciais)

ID – *Identify* (Identidade)

IOS – *Iphone Operating System* (Sistema Operacional do *Iphone*)

ISO – *International Organization for Standardization* (Organização Internacional de Normalização)

JIT – *Just-in-Time* (Na-hora-certa)

JPEG – *Joint Photographic Experts Group* (Grupo Conjunto de Peritos Fotográficos)

JSON – *JavaScript Object Notation* (Notação de objeto JavaScript)

LAN – *Local Area Network* (Rede Local)

LBI – Lei de Estatuto da Pessoa com Deficiência

LDB – Lei de Diretrizes e Bases da Educação

LIBRAS – Língua Brasileira de Sinais

LGPD – Lei Geral de Proteção de Dados

LLMs – *Large Language Models* (Modelos de Linguagem de Grande Escala)

MOF – *Meta-Object Facility* (Instalação de Meta-Objeto)

MPEG-4 – *Moving Picture Experts Group-4* (Grupo de Especialistas em Imagens em Movimento-4)

NSA – Novo Sistema Acadêmico

OMG – *Object Management Group* (Grupo de Gerenciamento de Objetos)

OMT – *Object Modeling Technique* (Técnica de Modelagem de Objetos)

OOP – *Object Oriented Programming* (Programação Orientada a Objetos)

OOSE – *Object-Oriented Software Engineering* (Engenharia de *Software* Orientada a Objetos)

PDO - *PHP Data Objects* (Pré-processador de Hipertexto de Objetos de Dados)

PHP – *Hypertext Preprocessor* (Pré-processador de Hipertexto)

PHPNG – *PHP Next Generation* (Pré-processador de Hipertexto da Próxima Geração)

PME – Pequenas e Médias Empresas

PNE – Plano Nacional de Educação

PNG – *Portable Network Graphics* (Gráficos Portáteis de Rede)

PK – *Primary Key* (Chave Primária)

PTD – Plano de Trabalho Docente

PSD – *Photoshop Document* (Documento do *Photoshop*)

POUR–

Perceivable, Operable, Understandable, Robust (Perceptível, Operável, Compreensível, Robusto)

RM – Registro Matrícula

SEO – *Search Engine Optimization* (Otimização para Mecanismos de Busca)

SDK – *Software Development Kit* (Kit de Desenvolvimento de *Software*)

SVG – *Scalable Vector Graphics* (Gráficos Vetoriais Escaláveis)

SSD – *Solid State Drive* (Unidade de Estado Sólido)

SO – Sistema Operacional

SQL – *Structured Query Language* (Linguagem de Consulta de Estrutura)

TEA – Transtorno do Espectro Autista

TCC – Trabalho de Conclusão de Curso

TIFF – *Tag Image File Format* (Formato de Arquivo de Imagem com Marcação)

UML – *Unified Modeling Language* (Linguagem de Modelagem Unificada)

URL – *Uniform Resource Locators* (Localizadores Uniformes de Recursos)

UX – *User Experience* (Experiência do Usuário)

W3C – *World Wide Web Consortium* (Consórcio Rede Mundial de Computadores)

WAN – *Wide Area Network* (Rede de Longa Distância)

WCAG – *Web Content Accessibility Guidelines* (Diretrizes de Acessibilidade para Conteúdo da *Web*)

WPF – *Windows Presentation Foundation* (Fundação de Apresentação do Windows)

WWW – *World Wide Web* (Rede Mundial de Computadores)

XAMPP – X-os Apache, MariaDB, PHP e Perl (Multiplataforma Apache, MariaDB, PHP e Perl.)

XHTML – *Extensible Hypertext Markup Language* (Linguagem de Marcação de Hipertexto Extensível)

XML – *Extensible Markup Language* (Linguagem de Marcação Extensível)

LISTA DE SÍMBOLOS

SUMÁRIO

DEDICATÓRIA	3
AGRADECIMENTO	4
EPÍGRAFE	5
RESUMO.....	6
ABSTRACT.....	7
LISTA DE FIGURAS	8
LISTA DE TABELAS	13
LISTA DE ABREVIATURAS E SIGLAS	14
LISTA DE SÍMBOLOS.....	19
SUMÁRIO	20
INTRODUÇÃO.....	24
1 – TECNOLOGIAS	26
1.1 – Lógica de Programação	26
1.2 – Programação <i>Web</i>	26
1.2.1 – HTML	27
1.2.2 – CSS	28
1.3 – Linguagem de Programação	29
1.3.1 – <i>Mobile</i>	30
1.3.2 – <i>Web</i>	31
1.3.3 – <i>Desktop</i>	32
1.4 – PHP	32
1.4.1 – Javascript.....	34
1.5 – XAMPP.....	37
1.6 – BANCO DE DADOS.....	38
1.6.1 – Modelagem	39
1.6.2 – MariaDB.....	39

1.6.3 – BRModelo	41
1.6.4 – Sintaxe Básica SQL	42
1.6.5 – Relacionamentos	43
1.7 – Python	43
1.7.1 – MediaPipe	46
1.8 – LÓGICA DE INTELIGÊNCIA ARTIFICIAL	47
1.8.1 – <i>Machine Learning</i>	47
1.8.2 – <i>LangFlow</i>	48
1.9 – SERVIDOR	49
1.9.1 – Servidor <i>Web</i>	50
1.9.2 – Servidor Local	50
1.10 – NAVEGADORES	51
1.11 – IMAGENS DIGITAIS	52
1.11.1 – Imagens Vetoriais	53
1.12 – <i>Design</i>	53
1.12.1 – <i>Design de Interfaces</i>	54
1.12.2 – Canva	55
1.12.3 – Inkscape	55
1.12.4 – Figma	57
1.12.5 – GIMP	58
1.13 – FORMATAÇÃO DE VÍDEOS	59
1.13.1 – MP4	59
2 – ESTUDO DE CASO	60
2.1 – VIABILIDADE	61
2.1.1 – Viabilidade Operacional	61
2.1.2 – Viabilidade Técnica	61
2.1.3 – Viabilidade de Cronograma	64

2.1.4 – Viabilidade Econômica.....	74
2.2 – TÉCNICA DE LEVANTAMENTO DE REQUISITOS	76
2.2.1 – Entrevista.....	76
2.2.2 – Questionário.....	78
2.2.3 - Análise de texto	80
2.3 – PROTOTIPAGEM	93
2.3.1 – Usuário comum.....	93
2.3.2 – Aluno.....	95
2.3.3 – Professor	101
2.3.4 – Secretaria	114
2.4 – CLIENTE	130
2.4.1 – Cliente Interno.....	130
2.4.2 – Cliente Externo	130
2.4.3 – Cliente Ativo.....	130
2.4.4 – Cliente Inativo	131
2.4.5 – Cliente Satisfeito.....	131
2.4.6 – Cliente Insatisfeito.....	131
2.5 – ÁREA DE ATUAÇÃO	131
2.5.1 – Microrregião.....	132
2.5.2 – Macrorregião.....	132
2.6 – Concorrentes.....	133
2.6.1 – Diretos	134
2.6.2 – Indiretos	138
2.6.3 – Conclusão.....	142
2.7 – LEIS E REGRAS CONTRATUAIS	142
2.7.1 – Termos e Condições de Uso do Sistema SCAN HANDS	143
2.7.2 – Contrato de Prestação de Serviços	150

3 – DOCUMENTAÇÃO	156
3.1 – OMG.....	156
3.1.1 – Utilização no Sistema	157
3.1.2 – Instalação de Meta-Objeto.....	158
3.2 – BPMN.....	158
3.2.1 – BPMN Secretaria	160
3.2.2 – BPMN Professor	162
3.2.3 – BPMN Aluno	164
3.3 – UML	165
3.3.1 – Aplicação no Sistema	167
3.4 – DIAGRAMAS.....	168
3.4.1– Diagrama de Classe	169
3.4.2 – Diagrama de Caso de Uso.....	171
3.5 – APLICAÇÃO WEB	173
3.5.1 - Tela de <i>Login</i>	174
3.5.2 – Tela inicial do Aluno.....	179
3.5.3 – Tela inicial do Professor.....	193
3.5.4 – Tela inicial da Secretaria.....	229
3.5.5 – Tela do funcionamento da Câmera.....	273
CONSIDERAÇÕES FINAIS	283
OBJETIVOS ALCANÇADO	283
OBJETIVOS INALCANÇADOS	284
ANEXOS	286
REFERÊNCIAS BIBLIOGRÁFICAS	287

INTRODUÇÃO

O desenvolvimento tecnológico tem contribuído para facilitar o trabalho e transformar ideias em projetos viáveis, reduzindo erros que anteriormente demandavam maior especialização. No Brasil, o acesso crescente a aparelhos eletrônicos é evidente, atingindo 94,9% da população entre 30 e 39 anos (O GLOBO, 2024). Nesse contexto, as aplicações móveis ganham importância, desempenhando um papel crucial no cotidiano ao transformar a forma como vive-se e gera-se oportunidades. A evolução de dispositivos móveis, que agora têm poder de processamento similar ao de *desktops*, e os avanços em tecnologias como 4G e 5G, refletem como essas inovações continuam moldando o futuro.

Com base nessas perspectivas, o projeto SCAN HANDS foi desenvolvido com o objetivo de facilitar o aprendizado e a prática da LIBRAS (Língua Brasileira de Sinais) de forma intuitiva e interativa. A plataforma visa solucionar o problema da dificuldade de acesso a professores especializados e promover a acessibilidade de forma ampla. Por meio de recursos tecnológicos, como o uso da câmera do dispositivo para o reconhecimento de gestos, o aplicativo oferece *feedback* em tempo real sobre a execução correta dos sinais. Essa funcionalidade permite que os usuários pratiquem, corrijam seus erros e aperfeiçoem suas habilidades de forma eficaz.

O SCAN HANDS também possibilita a interação com professores, criando um ambiente educativo mais dinâmico e acessível. Além disso, a plataforma inclui atividades didáticas interativas, incentivando tanto surdos quanto ouvintes interessados na linguagem a explorarem e dominarem a Libras. Essa abordagem torna o aprendizado mais inclusivo, ao mesmo tempo em que fortalece a comunicação entre comunidades.

A construção do projeto envolveu pesquisas detalhadas, entrevistas com potenciais usuários e especialistas, a fim de compreender as principais necessidades e desafios. Foram investigadas tecnologias adequadas para o desenvolvimento da aplicação, com uma análise profunda sobre origens, funcionalidades e evolução ao longo do tempo. Os aspectos técnicos e operacionais da solução, como interface do usuário, usabilidade e processos de interação, foram abordados cuidadosamente para garantir uma experiência prática e agradável. O projeto busca impactar positivamente os usuários, promovendo maior acessibilidade, inclusão e interesse pela Libras. A expectativa é que o SCAN HANDS não apenas resolva o problema da dificuldade de

aprendizado e prática da linguagem de sinais, mas também contribua para um ambiente mais inclusivo e acessível para todos os seus usuários.

Além disso, o projeto foi estruturado em três capítulos principais no Trabalho de Conclusão de Curso (TCC). O primeiro capítulo apresenta uma análise abrangente das tecnologias utilizadas, abordando sua origem, evolução histórica, características, e como essas ferramentas foram aplicadas no desenvolvimento do SCAN HANDS. Esse capítulo fornece uma base sólida e detalhada sobre os aspectos tecnológicos que sustentam a aplicação. No segundo capítulo, a viabilidade do projeto é explorada, documentando as respostas coletadas durante as entrevistas e explicando como o aplicativo opus na prática. Aspectos como interface do usuário, interação e usabilidade são detalhados, demonstrando as funcionalidades e benefícios da aplicação. Por fim, o terceiro capítulo foca no desenvolvimento técnico, apresentando os métodos, códigos e etapas de criação. Essa seção inclui representações visuais do progresso do projeto, além de uma análise final dos resultados obtidos. O capítulo conclui com uma avaliação abrangente, conectando o impacto da aplicação às necessidades do público-alvo. O SCAN HANDS representa uma inovação tecnológica que pode inspirar o desenvolvimento de novas soluções voltadas para a inclusão social e educacional.

A evolução contínua do aplicativo é um dos seus pontos fortes, pois ele pode ser expandido para incluir funcionalidades adicionais no futuro. Por exemplo, futuras versões poderiam integrar traduções automáticas em Libras para textos ou voz, ou até mesmo criar comunidades virtuais onde os usuários possam trocar experiências e aprender em grupo. Essas inovações reforçam o potencial do SCAN HANDS como uma ferramenta essencial para a inclusão e acessibilidade, oferecendo um impacto duradouro na vida de seus usuários.

1 – TECNOLOGIAS

Tecnologias são conjuntos de conhecimentos, habilidades, métodos e processos usados para criar, desenvolver, melhorar ou aplicar ferramentas, máquinas, sistemas ou técnicas que atendem a necessidades humanas. Elas englobam desde invenções simples, como uma roda ou uma lâmpada, até inovações mais complexas, como inteligência artificial, *internet*, biotecnologia e engenharia espacial. A palavra tecnologia tem origem do grego “*teknhe*” e “*logia*” que significam “técnico” e “estudo”, portanto, tecnologias são estudos técnicos. (ENCICLOPÉDIA SIGNIFICADOS, 2024)

1.1 – LÓGICA DE PROGRAMAÇÃO

A lógica de programação é a capacidade de organizar instruções e comandos em conjuntos que possibilitem a resolução de problemas, e que façam sua tarefa ser realizada e executada de forma desejada, se concentrando em dividir em etapas um projeto, possibilitando sua realização passo a passo, utilizando uma linguagem de programação ou uma pseudolinguagem. De forma simples, um conjunto de instruções para resolver um problema.

É uma das habilidades mais requisitadas na vida profissional de uma pessoa desenvolvedora, pois as permite desenvolver códigos claros, eficientes e de fácil manutenção, garantindo um bom funcionamento do *software*. Ela também possibilita a compreensão e a implementação de algoritmos e como melhorar a lógica em geral ou encontrar soluções para novos desafios, sendo indispensável para a viabilidade de qualquer projeto. (LOCAWEB, 2024).

1.2 – PROGRAMAÇÃO WEB

A programação *web* é a arte de desenvolver e manter *sites* e aplicações *online*. Esse processo envolve o uso de diversas linguagens, como HTML (*HyperText Markup Language* – Linguagem de Marcação de Hipertexto) e CSS (*Cascading Style Sheets* – Folhas de Estilo em Cascata), que ajudam a estruturar e estilizar páginas da *web*. Programadores *web* escrevem códigos que tornam possível a exibição de conteúdos, a interação dos usuários e a comunicação com servidores. Além disso, a programação *web* engloba o uso de *frameworks*, otimização para dispositivos móveis, técnicas de SEO (*Search Engine Optimization* – Otimização para Mecanismos de Busca), segurança, desempenho, integração com API (*Application Programming*

Interface - Interface de Programação de Aplicações), controle de versões e testes, garantindo *sites* funcionais e seguros. (ESCRITÓRIO VIRTUAL, 2024).

1.2.1 – HTML

O HTML (*HyperText Markup Language* – Linguagem de Marcação de Hipertexto) é uma linguagem de marcação utilizada na construção de páginas na *Web*. Foi desenvolvido em 1990 por Tim Berners-Lee na CERN (*Conseil Européen pour la Recherche Nucléaire* – Organização Europeia para a Pesquisa Nuclear), em Genebra, Suíça, juntamente com o protocolo HTTP (*Hypertext Transfer Protocol* – Protocolo de Transferência de Hipertexto), dedicado à transferência de documentos nesse formato. (TECNOBLOG,2025)

Em uma página *web*, o HTML é responsável por estruturar o conteúdo, permitindo a inserção de textos, imagens, vídeos e outros elementos multimídia. Ele define a organização e a hierarquia dos componentes, facilitando a interpretação e exibição das informações pelos navegadores.

O HTML é utilizado por meio de marcações conhecidas como "*tags*", que são elementos de código que estruturam a página, consistindo em uma abertura e um fechamento, como `<tag></tag>`, funcionando como blocos fundamentais na construção das páginas, organizando e definindo os diferentes componentes do *site*. Por exemplo, a *tag* `<p>` indica um parágrafo, enquanto a *tag* `<a>` cria um *hiperlink*. Essas marcações permitem aos desenvolvedores *web* construir páginas de forma estruturada e semântica. A Figura 1 ilustra o logotipo do HTML5, em forma de escudo com tonalidades em vermelho e laranja. O número "5" centralizado em branco representa a quinta versão da linguagem, destacando sua robustez e modernidade. O estilo visual reforça a ideia de estrutura, força e padronização no desenvolvimento *web*.

A primeira versão do HTML, conhecida como HTML 1.0, foi lançada em 1991. Essa versão incluía *tags* básicas para formatar o texto, como `<h1>` para títulos, `<p>` para parágrafos e `<a>` para *links*. Também incluía imagens, embora apenas em preto e branco. (NETCODERS, 2024)

Em 1995, a especificação do HTML foi atualizada para o HTML 2.0, que incluía suporte a tabelas, formulários e *framesets* (permitindo que diferentes páginas da *web* fossem exibidas dentro de uma única janela). Nesse mesmo ano, foi lançado o navegador Mosaic, que se tornou popular entre os usuários da *web*. (NETCODERS, 2024)

Em 1997, o HTML 3.0 foi lançado, trazendo suporte para folhas de estilo (CSS) e elementos multimídia, como vídeo e som. No mesmo ano, a Netscape lançou o navegador *Netscape Navigator*, que se tornou o navegador mais popular da época. (NETCODERS, 2024)

Em 1999, foi lançado o HTML 4.0, que incluía novos recursos como tabelas avançadas, *frames* e suporte para *scripts* do lado do cliente (JavaScript). O HTML 4.0 também incluía a especificação de documentos XHTML (*Extensible Hypertext Markup Language* – Linguagem de Marcação de Hipertexto Extensível), que combinava o HTML com o XML (*Extensible Markup Language* – Linguagem de Marcação Extensível). (NETCODERS, 2024)

Em 2004, o desenvolvimento do HTML foi retomado com a criação do grupo de trabalho HTML da W3C (*World Wide Web Consortium* – Consórcio Rede Mundial de Computadores). Esse grupo trabalhou no desenvolvimento do HTML5, que foi lançado em 2014. O HTML5 incluiu muitos novos recursos, como elementos semânticos (como *e*), suporte para vídeo e áudio, geolocalização e capacidade de armazenamento local. (NETCODERS, 2024)

Figura 1 - HTML



FONTE: PINTEREST, 2025

1.2.2 – CSS

O CSS é um padrão que define como os dados são apresentados no navegador. Enquanto o HTML fornece informação sobre a estrutura de um documento, o CSS indica o aspecto que ele deve ter: a fonte, o fundo, o texto, as cores dos *links*, as margens e a disposição dos objetos na página. (EBAC, 2023)

O principal objetivo do desenvolvimento do CSS é separar o conteúdo seja este escrito em HTML ou outra linguagem de marcação do *design* de um documento. Essa separação pode aumentar a acessibilidade dos documentos, proporcionar mais flexibilidade e controle de sua apresentação, bem como reduzir a complexidade e a repetição de conteúdos estruturados. Além disso, o CSS permite apresentar o mesmo documento em diferentes estilos ou métodos de saída, como em uma tela, impressão,

leitura por voz ou quando a saída utiliza dispositivos que usam Braille. (EBACONLINE, 2023)

1994 – Em 10 de outubro de 1994, Hakon criou a primeira proposta do que viria a ser o CSS. Sua intenção era diminuir as tensões entre autores e leitores da *web*, facilitando o ambiente. (BLOG, 2021)

Versões:

“CSS1: A especificação CSS1, lançada em 1996, introduziu estilos básicos como cores, fontes e planos de fundo, mas tinha limitações no controle de *layout*.” (ALMABETTER, 2024)

“CSS2: Lançado em 1998, introduziu uma infinidade de novos recursos, incluindo posicionamento, controle aprimorado sobre *layout* e a introdução de tipos de mídia para diferentes dispositivos;” (ALMABETTER, 2024)

“CSS2.1: Esta versão teve seu lançamento em 2011, ela se concentrou em esclarecer e melhorar a especificação CSS2, proporcionando maior estabilidade e compatibilidade com navegadores;” (ALMABETTER, 2024)

“CSS3: É composto por módulos introduzidos gradualmente, abrangendo recursos como animações, gradientes e *layouts* flexíveis, e ainda está em desenvolvimento.” (ALMABETTER, 2024)

“CSS4: Em desenvolvimento, essa versão é o futuro do CSS, com o objetivo de aprimorar ainda mais os recursos de *web design*, incluindo *design* responsivo, fontes variáveis e sistemas de grade aprimorados.” (ALMABETTER, 2024)

1.3 – LINGUAGEM DE PROGRAMAÇÃO

Uma linguagem de programação é basicamente uma forma de comunicação entre o humano e a máquina. Ela é utilizada de forma a instruir o programa de computador, *site* de *internet*, aplicativo de celular e até os modelos mais modernos de televisores. (TECMUNDO, 2024)

Estas instruções são passadas em forma de *scripts*, que nada mais são do que rotinas e instruções que os dispositivos devem seguir para funcionar de maneira apropriada. Para isto são utilizados comandos, regras semânticas e sintáticas, palavras-chave e até símbolos, de forma a permitir aos aparelhos entenderem o que devem ou não fazer dependendo da ação do usuário. (ROCKCONTENT, 2022)

Constituída por três áreas principais, as linguagens de programação são definidas por: *mobile*, *web* e *desktop*, cada uma voltada para diferentes necessidades no desenvolvimento de *software*. No desenvolvimento *mobile*, linguagens como Java e Kotlin são utilizadas para Android, enquanto Swift é a principal escolha para iOS

(*Iphone Operating System* – Sistema Operacional do *Iphone*), mas outras como o C#, que ajuda o desenvolvimento *mobile* utilizando estruturas como o Xamarin e o Objective-C que é muito relevante no caso de mais antigos. Além disso, *frameworks* como React Native e Flutter permitem criar aplicativos multiplataforma. Na *web*, HTML, CSS e JavaScript são essenciais para interfaces interativas, enquanto linguagens como Python, Ruby, PHP (*Hypertext Preprocessor* - Pré-processador de Hipertexto) e Java são extremamente utilizadas no *back-end* para garantir aplicações escaláveis e robustas. Já no desenvolvimento *desktop*, linguagens como C#, Java, Swift, C++ e Python são frequentemente utilizadas para criar *softwares* que maximizam o desempenho do *hardware* e protegem uma experiência de usuário rica e eficiente.

1.3.1 – *Mobile*

O termo “*mobile*” significa “móvel” ou “portátil” e passou a ser utilizado no contexto tecnológico para se referir a dispositivos moveis, como celulares e *tablets*. (AGÊNCIA FOLLOW, 2024)

Em 1992, Frank Canova, engenheiro da IBM (*International Business Machines Corporation* – Corporação Internacional de Máquinas Comerciais), desenvolveu o primeiro *smartphone* do mundo, chamado Simon Personal Communicator ou IBM Simon. (TECTUDO, 2023)

O sistema iOS possui várias linguagens que tornam o Android ser amplamente mais utilizado, como o Java que possui recursos como o Android SDK (*Software Development Kit* – Kit de Desenvolvimento de *Software*) e bibliotecas de terceiros, como Firebase e Google Maps API (*Application Programming Interface* - Interface de Programação de Aplicações).

Desde 2017, o Kotlin se tornou a principal linguagem para desenvolvimento Android por ser uma alternativa mais concisa e segura ao Java. É mais concisa, segura e eficiente Java. (ALURA,2025)

O JavaScript é essencial para adicionar interatividade a página *web*, tendo *frameworks* como o React Native que permite criar aplicativos híbridos para iOS e Android com uma única base de código, escrever código em JavaScript e renderizar componentes nativos, proporcionando uma experiência de usuário similar aos aplicativos nativos. Swift é uma linguagem de programação desenvolvida pela Apple para a criação de aplicativos para iOS.

É uma linguagem poderosa e intuitiva que visa facilitar o desenvolvimento de aplicativos de alta performance e seguros, além de ser compatível com Objective-C, C#, usado com o *framework* Xamarin, permite compartilhar até 90% do código entre diferentes plataformas (Android, iOS e Windows), oferecendo uma sintaxe familiar, suporte à programação assíncrona e recursos avançados para gerenciamento de memória e depuração.

O Flutter, com a linguagem Dart, permite criar aplicativos nativos para Android e iOS com uma base de código única. A escolha da tecnologia depende do projeto, com Android oferecendo mais liberdade e iOS mais estabilidade. Essas ferramentas evoluem para melhorar a produtividade e a experiência do usuário.

1.3.2 – Web

A *web* significa teia ou rede, é uma rede global de documentos interconectados e outros recursos, acessíveis pela *Internet*. As páginas *web* são documentos criados para serem exibidos em navegadores da *web*, e elas podem variar muito em termos de complexidade e *design*. Estes documentos são escritos com a linguagem HTML.

1990 – Tim Berners Lee cria WWW (*World Wide Web* - Rede Mundial de Computadores) que posteriormente tornaria a *internet* no que ela é hoje. (BBC, 2024)

Versões:

Web 1.0: Proposta por Tim Berners-Lee em 1989, a primeira geração da *web* baseava-se em *hiperlinks* e era utilizada para, basicamente, exibir conteúdo estático com um *design* semelhante ao de páginas impressas. Vídeos eram raros. (AGATETEPE, 2023)

Web 2.0: A chegada da segunda versão da *web* revolucionou a experiência dos usuários, sendo distinta de tudo que já tinha sido visto antes. Nela era possível ir além das páginas estáticas que imprimiam conteúdo na tela, já que os *sites* passaram a oferecer novas formas de interação. (AGATETEPE, 2023)

Web 3.0: Também conhecida como *Web3*, a terceira geração da *internet* explorou muito mais do que as ideias sugeridas pela W3C e normalmente é mencionada em temas relacionados à *blockchain*, NFTs (*non-functional tokens* - *tokens* não-fungíveis) e descentralização. A *Web 3.0* nasce dos processos de

transformação digital e baseia-se no uso de máquinas para melhorar a eficiência de atividades antes realizadas manualmente pelo usuário. (AGATETEPE, 2023)

Web 4.0: Com base nas tendências vistas atualmente, é relativamente fácil imaginar como será a próxima grande transformação da *web*. Nunca na história da *internet* se falou tanto sobre políticas de privacidade e esse tende a ser um ponto central daqui em diante. (AGATETEPE, 2023)

Essa tendência já vem sendo notada em conceitos como o Open Finance, Open Health, nas permissões explícitas para armazenamento de *cookies* em navegadores e na ideia geral do próprio usuário ter controle sobre o compartilhamento dos próprios dados na rede. (AGATETEPE, 2023)

1.3.3 – *Desktop*

Quando se fala de *desktop* no contexto de linguagens de programação, o foco está no desenvolvimento de *software* para esse ambiente, ele que permite aos usuários executarem programas, editar documentos, navegar na *internet* e realizar outras atividades. As linguagens mais comuns para criar aplicativos de *desktop* incluem C++, conhecida pela performance e acesso direto aos recursos do sistema operacional; Java, famosa pela portabilidade entre sistemas; Python, amplamente utilizada devido à simplicidade e diversas bibliotecas disponíveis; C#, ideal para aplicativos Windows com o suporte da *.NET framework*; e *Swift*, projetada para desenvolvimento de aplicações no macOS. (SABER TECNOLOGIAS, 2025)

Essas linguagens permitem criar interfaces gráficas, também chamadas de GUI (*Graphical User Interface* – Interface Gráfica do Usuário), que incluem elementos como janelas, botões e menus. Por exemplo, ao desenvolver um editor de texto ou *software* financeiro, o programador pode utilizar bibliotecas como *Tkinter* (em Python), WPF (em C#) ou *Swing* (em Java) para construir componentes interativos no ambiente de *desktop*. Além disso, os aplicativos *desktops* frequentemente se comunicam com os sistemas operacionais para gerenciar recursos como arquivos, memória e dispositivos externos.

1.4 – PHP

Criado em 1994 por Rasmus Lerdorf, inicialmente conhecida como *Personal Home Page*, o PHP tem como objetivo criar aplicações para a *web*, como *sites* dinâmicos, *blogs*, redes sociais e sistemas de gerenciamento de conteúdo. Sendo uma das suas principais vantagens a simplicidade, tornando-o uma ótima opção tanto para iniciantes quanto para desenvolvedores experientes. (UDEMY, 2020)

Em 1997, foi lançado a versão 3, esta foi uma reescrita completa do PHP/FI por Andi Gutmans e Zeev Suraski. Introduziu uma nova sintaxe e funcionalidades, tornando-se uma linguagem de *script* mais robusta e popular para desenvolvimento *web*. (HOSTMÍDIA, 2025)

Em meados de 2000 foi lançado a versão 4, baseado no Zend Engine 1, desenvolvida pelos mesmos criadores da versão 3, o PHP 4 trouxe melhorias significativas em desempenho e funcionalidades. Suporte para sessões, *buffers* de saída e maior modularidade foram algumas das novidades. Em 2004, o PHP 5 introduziu o Zend Engine 2, que trouxe suporte completo o OOP (*Object Oriented Programming* – Programação Orientada a Objetos). Outras melhorias incluíram suporte a XML, integração com o SQLite e a introdução da biblioteca de objetos PDO (*PHP Data Objects* – Pré-processador de Hipertexto de Objetos de Dados) para acesso a banco de dados. (HOSTMÍDIA, 2025)

De 2005 a 2010 a nova versão seria o PHP 6, este projeto visava adicionar suporte nativo a Unicode, mas enfrentou muitos desafios e nunca foi oficialmente lançado. Muitos dos recursos planejados foram posteriormente incorporados em outras versões. (HOSTMÍDIA, 2025)

Em 2015 o PHP 7, baseado no projeto PHPNG (*PHP Next Generation* – Pré-processador de Hipertexto da Próxima Geração), o PHP 7 trouxe melhorias significativas em desempenho, com um aumento de quase 100% em *benchmarks* baseados no WordPress. Introduziu o Zend Engine 3, suporte a tipos escalares, operadores de coalescência nula e muito mais. (HOSTMÍDIA, 2025)

O PHP 8 em 2020 introduziu o compilador JIT (*Just-in-Time* – Na-hora-certa), que melhorou ainda mais o desempenho. Outras novidades incluíram atributos, operadores de atribuição nula, e melhorias na tipagem e tratamento de erros. (HOSTMÍDIA, 2025)

Um ano depois em 2021 o PHP 8.1 adicionou *enums* (uma classe ou funcionalidade que representa um conjunto de valores constantes), fibras (*fibers*) para programação assíncrona, e melhorias na tipagem e desempenho. (HOSTMÍDIA, 2025)

PHP 8.2 em 2022, continuou a evolução com melhorias na segurança, desempenho e novas funcionalidades, como tipos de interseção e melhorias na sintaxe. (HOSTMÍDIA, 2025)

PHP 9 seria o futuro dessa linguagem. A comunidade PHP continua a trabalhar em novas funcionalidades e melhorias, com foco em desempenho, segurança e facilidade de uso. O futuro do PHP promete continuar a evolução da linguagem, mantendo-a relevante e poderosa para o desenvolvimento *web*. A Figura 2 apresenta o logotipo oficial do PHP, criado por Colin Viebrock. Ele consiste em uma elipse azul escura com a palavra “php” centralizada, escrita em letras minúsculas e em negrito, representando a identidade visual simples e direta da linguagem. (HOSTMÍDIA, 2025)

Figura 2 - PHP



FONTE: LIBLOGO, 2024

1.4.1 – Javascript

O Javascript, é uma linguagem de programação amplamente utilizada no desenvolvimento *web*, permitindo a criação de páginas interativas e dinâmicas. Em 1995 Netscape contratou o desenvolvedor Brendan Eich para criar uma linguagem de *script* que trouxesse “vida” às páginas *Web*. Desse modo, foi criada uma linguagem que foi chamada de LiveScript. (ALURA, 2025)

Após uma parceria com a Sun Microsystems, empresa responsável pela criação da linguagem Java, o nome LiveScript foi alterado para JavaScript. Essa foi uma jogada de *marketing* feita com o objetivo de chamar atenção para o JavaScript através da fama que o Java já tinha conquistado naquele momento. A linguagem evoluiu significativamente e se tornou um dos pilares da *web* moderna, ao lado de HTML e CSS. Com a capacidade de manipular o DOM (*Document Object Model* –

Modelo de Objeto de Documento, o Javascript possibilita que desenvolvedores criem experiências ricas e responsivas para os usuários.

Uma das principais características do Javascript é sua natureza interpretada, o que significa que o código é executado diretamente pelo navegador, sem a necessidade de compilação prévia. Isso proporciona um ciclo de desenvolvimento mais rápido e facilita a depuração. Além disso, o Javascript é uma linguagem de tipagem dinâmica, permitindo que variáveis sejam declaradas sem a necessidade de especificar seu tipo, o que proporciona flexibilidade ao programador.

No contexto do desenvolvimento *web*, o Javascript é essencial para a criação de interfaces de usuário interativas. Com o uso de bibliotecas e *frameworks* como React, Angular e Vue.js, os desenvolvedores podem construir aplicações *web* complexas de forma mais eficiente. Essas ferramentas oferecem componentes reutilizáveis e facilitam a gestão do estado da aplicação, tornando o desenvolvimento mais organizado e escalável.

O Javascript também é amplamente utilizado para gerenciar eventos, como cliques de *mouse*, teclas pressionadas e movimentos do cursor. Através do uso de *event listeners*, os desenvolvedores podem definir ações que devem ocorrer em resposta a esses eventos, tornando a interação com a página mais intuitiva e responsiva. Essa capacidade de resposta é fundamental para a criação de aplicações *web* modernas e interativas. A Figura 3 apresenta o logotipo oficial do JavaScript, criado por Chris Williams. A imagem mostra um escudo amarelo com a letra “S”, representando a linguagem de forma moderna e compacta, símbolo comum em materiais didáticos e interfaces de desenvolvimento.

Versões:

ECMAScript 1 (1997): Primeira edição, estabeleceu a base do JavaScript como linguagem padronizada. (LOCAWEB,2024)

ECMAScript 2 (1998): Atualizações editoriais para alinhamento com padrões ISO (*International Organization for Standardization* – Organização Internacional de Normalização), sem alterações funcionais. (LOCAWEB,2024)

ECMAScript 3 (1999): Grande atualização, com introdução de expressões regulares, manipulação de *strings* aprimorada e blocos *try/catch* para tratamento de erros. (LOCAWEB,2024)

ECMAScript 4: Nunca foi oficialmente lançado, mas continha ideias que influenciaram versões futuras. (LOCAWEB,2024)

ECMAScript 5 (2009): Adicionou suporte ao "*strict mode*", métodos de iteração de *arrays* como *forEach*, *map* e suporte a JSON (JavaScript *Object Notation* – Notação de objeto JavaScript) nativo. (LOCAWEB,2024)

ECMAScript 5.1 (2011): Revisão menor para melhor alinhamento com a especificação do padrão. (LOCAWEB,2024)

ECMAScript ES6 (2015): Mudança significativa com classes, *let*, *const*, funções de seta, promessas (*Promise*), módulos *import/export* e *templates* literais. (LOCAWEB,2024)

ECMAScript ES7 (2016): Introduziu o operador de exponenciação (******) e o método *Array.prototype.includes*. (LOCAWEB,2024)

ECMAScript ES8 (2017): Destacou-se por *async/await* para melhor gerenciamento de operações assíncronas e métodos *Object.entries* e *Object.values*.

ECMAScript ES9 (2018): Adicionou o operador de propagação/*rest* para objetos (...) e melhorias para expressões regulares. (LOCAWEB,2024)

ECMAScript ES10 (2019): Introduziu o método *Array.prototype.flat* e o operador de encadeamento opcional (**?.**). (LOCAWEB,2024)

ECMAScript ES11 (2020): Apresentou o operador de coalescência nula (**??**), importação dinâmica (*import()*) e outras melhorias. (LOCAWEB,2024)

ECMAScript ES12 (2021): Adicionou o método *String.prototype.replaceAll* e sinalizadores de coleções fracas (*WeakRefs*). (LOCAWEB,2024)

ECMAScript ES13 (2022): Recursos como métodos *at()* para *arrays* e *strings*, além de novas funcionalidades em módulos. (LOCAWEB,2024)

Figura 3 - Javascript



FONTE: PINTEREST, 2024

1.5 – XAMPP

XAMPP (X-os, Apache, MariaDB, PHP e Perl – Multiplataforma Apache, MariaDB, PHP e Perl.) é um *software* de uso grátis e que possui o código aberto. Ele facilita o desenvolvimento e por ter acesso ao pacote dos principais servidores, se tornou muito popular e com uma comunidade muito ativa. Desde seu lançamento em 2002, o XAMPP evoluiu para acompanhar as tecnologias modernas. Em 2015, o MariaDB foi introduzido no lugar do MySQL, trazendo melhorias e mantendo o compromisso com o código aberto. Com ele, é possível rodar sistemas como WordPress e Drupal localmente, o que facilita e agiliza o desenvolvimento. Como o conteúdo estará armazenado numa rede local, o acesso aos arquivos é realizado instantaneamente. O pacote de servidores é baixado cerca de 600 mil vezes por mês, de acordo com dados do SourceForge. (HIGA, 2012)

O XAMPP, é um pacote de *software* gratuito e de código aberto que facilita a criação de servidores locais para desenvolvimento *web*. Ele inclui componentes essenciais como o servidor Apache, o banco de dados MariaDB (anteriormente MySQL), e suporte para linguagens como PHP e Perl. O XAMPP é amplamente utilizado por desenvolvedores para testar e criar *sites* e aplicativos antes de lançá-los *online*.

Uma das principais vantagens do XAMPP, conforme ilustrado na Figura 4, é sua facilidade de instalação e uso. A imagem apresenta o logotipo do XAMPP, que consiste em um símbolo laranja com um “X” estilizado branco ao centro, remetendo à simplicidade e integração da ferramenta. Ele é compatível com sistemas operacionais como Windows, Linux e macOS, e oferece ferramentas como o PhpMyAdmin para gerenciar bancos de dados de forma intuitiva. Além disso, o XAMPP permite configurar ambientes de desenvolvimento completos sem a necessidade de configurações complexas.

Figura 4 - Xampp



FONTE: WRITE, 2019

1.6 – BANCO DE DADOS

Um banco de dados é a estrutura essencial para qualquer aplicação existente, já que pode ser utilizado em vários contextos, desde armazenamentos de dados de pacientes, a lanches ou controle de animais. Ele é a organização e armazenamento de informações sobre algo, sendo um “estoque” para informações e dados que serão utilizados em aplicações. Sem essas informações salvas o aplicativo não conseguiria funcionar corretamente, já que sem ele haverá uma lacuna no que é possível ser feito e o que não é, sendo a primeira coisa a ser definida nesses programas atuais. (ORACLO, 2024)

A operação geralmente é feita e modelada em linhas e colunas, em uma série de tabelas para tornar o processamento e consulta eficiente, os dados podem ser facilmente acessados e modificados pelos administradores, mas nunca a um usuário comum, já que ele é a base de tudo, e qualquer mudança errada poderá atrapalhar o funcionamento da aplicação.

Além de organizar dados de forma estruturada, um banco de dados também oferece mecanismos robustos para garantir a integridade e segurança das informações. Ferramentas como transações e controle de acesso garantem que apenas usuários autorizados possam modificar ou acessar dados sensíveis. Essa capacidade de controle e segurança é crucial para aplicações que lidam com informações críticas, como registros médicos ou informações financeiras, protegendo contra perda de dados e violações de privacidade. Além disso, sistemas modernos de bancos de dados utilizam tecnologias avançadas como armazenamento em nuvem, análise de *Big Data* e inteligência artificial para expandir suas capacidades e atender às demandas de aplicações contemporâneas.

1.6.1 – Modelagem

A modelagem de banco de dados desempenha um papel fundamental no desenvolvimento de sistemas organizados e eficientes. Ela garante que os dados sejam representados de maneira clara, consistente e fácil de compreender, proporcionando uma base sólida para a construção de qualquer aplicação. O processo começa com a modelagem conceitual, que identifica as entidades e seus atributos, além de mapear os relacionamentos entre elas. Por exemplo, cada um possui suas respectivas características, como nome, preço ou data. (ALURA, 2025)

O modelo lógico é uma versão refinada do modelo conceitual, convertendo entidades e atributos em tabelas e colunas que podem ser implementadas diretamente em um sistema de banco de dados. Possui nome de entidade, um ID (*Identity* - Identidade), (podendo ser compreendido como identificador), algo que define quem ele é e PK (*Primary Key* – Chave Primária) sendo entendido como chave primária relacionada ao identificador, relacionamentos FK (*Foreign Key* – Chave Estrangeira) sendo uma chave que busca dados únicos em tabelas estrangeiras), cada qual possuindo um nome simples de se entender como por exemplo “Cliente Nome”.

A modelagem física, por sua vez, é responsável por determinar como o banco de dados será implementado no nível técnico. Isso inclui a escolha dos tipos de dados apropriados para cada coluna (como INT para números inteiros ou *VARCHAR* para textos), a criação de índices para agilizar consultas e o planejamento do armazenamento em disco.

1.6.2 – MariaDB

O MariaDB é um sistema de gerenciamento de banco de dados relacional (RDBMS) de código aberto, amplamente utilizado em aplicações que exigem alta performance, segurança e escalabilidade. Ele foi criado como um *fork* do MySQL, em 2009 após a aquisição deste pela Oracle, e é mantido por uma comunidade independente. Essa independência garante que o MariaDB continue evoluindo com foco em inovação e melhorias, sem restrições comerciais. (PURE STORAGE, 2025)

O MariaDB é escrito em C e C++ e é compatível com várias linguagens de programação, incluindo C, C#, Java, Python, PHP e Perl. O MariaDB também é compatível com todos os principais sistemas operacionais, incluindo Windows, Linux

e macOS, isso permite realizar operações como consultas, inserções, atualizações e exclusões de dados. Ele também suporta transações, garantindo que conjuntos de operações sejam executados de forma completa ou não sejam realizados, o que é essencial para manter a integridade dos dados em sistemas críticos. A Figura 5 apresenta o logotipo do MariaDB, criado pelo desenvolvedor Michael "Monty" Widenius. A imagem mostra uma foca em tons de azul e bege voltada para cima, simbolizando o nome "Maria" (filha do criador) e a leveza e robustez do sistema, sendo uma marca registrada da identidade visual do banco de dados.

Figura 5 - MariaDB



FONTE: MEDIUM, 2023

Versões:

5.1 (2010): Uma bifurcação do MySQL 5.1, trazendo melhorias de desempenho e otimizações específicas. (AWS, 2025)

5.5 (2012): Introduziu diversas melhorias de desempenho e novos recursos, como suporte aprimorado à replicação. (AWS, 2025)

10.0 (2014): Adicionou suporte a subconsultas, melhorias no desempenho e funcionalidades avançadas para transações. (AWS, 2025)

10.1 (2015): Incluiu recursos como suporte ao formato JSON, otimizações para *clusters* e ferramentas avançadas de segurança. (AWS, 2025)

10.2 (2017): Trouxe suporte a janelas de função, aprimoramentos em mesclagem de tabelas e novas funções analíticas. (AWS, 2025)

10.3 (2018): Introduziu funcionalidades temporais para dados históricos, além de melhorias no desempenho e replicação. (AWS, 2025)

10.4 (2019): Incorporou avanços significativos em segurança, maior robustez na replicação e suporte a conexões criptografadas. (AWS, 2025)

10.5 (2020): Adicionou suporte avançado para dados JSON, aprimoramentos na performance de consultas e recursos otimizados de replicação. (AWS, 2025)

10.6 (2021): Implementou melhorias na escalabilidade e integração com sistemas de armazenamento em nuvem. (AWS, 2025)

10.11 (2023): Versão com suporte a análises avançadas e melhorias voltadas para *Big Data* e inteligência artificial. (AWS, 2025)

1.6.3 – BRModelo

O BrModelo, é uma ferramenta gratuita desenvolvida em 2005 por um grupo de banco de dados da Universidade Federal de Santa Catarina, a finalidade do BrModelo é incentivar, apoiar e auxiliar o desenvolvimento de bancos de dados relacionais. (CÂNDIDO, 2020)

A Figura 6 apresenta a logomarca do BrModelo, composta por um diagrama simplificado de modelagem de dados com símbolos geométricos: retângulos verdes e amarelos interligados por linhas, além de losangos e conectores pretos e azuis, representando visualmente entidades, relacionamentos e fluxos entre eles. Essa imagem remete diretamente ao propósito da ferramenta, que é a criação de modelos entidade-relacionamento.

Versões:

1.0: Lançamento inicial da ferramenta, com recursos básicos de modelagem de banco de dados, focando na criação de diagramas de entidade-relacionamento e conceitos fundamentais. (BRMODELO, 2020)

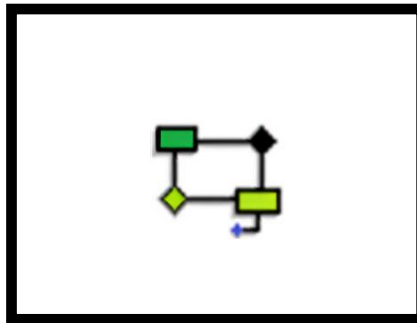
2.0: Introduziu melhorias na interface do usuário e novos recursos de modelagem, como suporte aprimorado para diagramas lógicos e físicos. (BRMODELO, 2020)

2.1: Versão de atualização que trouxe correções de *bugs* e melhorias de desempenho, garantindo maior estabilidade e usabilidade. (BRMODELO, 2020)

3.0: Uma versão significativa que incluiu novos recursos e melhorias na usabilidade, tornando a ferramenta mais intuitiva e eficiente. (BRMODELO, 2020)

4.0: Se houver, poderia trazer suporte a plataformas mais recentes e recursos avançados de modelagem, como integração com sistemas de banco de dados modernos. (BRMODELO, 2020)

Figura 6 - BRModelo



FONTE: PINTEREST, 2025

1.6.4 – Sintaxe Básica SQL

A sintaxe básica do SQL (*Structured Query Language* – Linguagem de Consulta de Estrutura) define as regras e estruturas utilizadas para manipular dados em bancos de dados relacionais. Baseada na linguagem inglesa e influenciada por outras linguagens, a sintaxe do SQL é projetada para ser intuitiva e eficiente, permitindo criação, consulta e modificação de bancos de dados.

Essa linguagem é composta por comandos que desempenham funções específicas, como inserir, recuperar e excluir dados, além de criar ou alterar estruturas de tabelas e definir regras para os dados armazenados. Combinando diferentes comandos e cláusulas, o SQL oferece uma maneira poderosa de interagir com os bancos de dados, garantindo organização e segurança nas operações realizadas.

Exemplo: *INSERT*, *SELECT*, *DELETE*.

O comando *INSERT* é utilizado para adicionar novos registros em uma tabela do banco de dados. Sua sintaxe básica envolve especificar o nome da tabela e os campos onde os valores serão inseridos, utilizando a cláusula *VALUES* para definir os novos dados. (RLSYSTEM, 2025)

O comando *SELECT* permite recuperar informações armazenadas em uma ou mais tabelas do banco de dados. Ele possibilita a seleção de campos específicos e, através da cláusula *WHERE*, aplicar filtros que refinam os resultados obtidos conforme critérios definidos. (SUPPORT.MICROSOFT, 2025)

O comando *DELETE* é usado para remover registros de uma tabela do banco de dados. Para evitar a exclusão acidental de todos os dados, é essencial utilizar a cláusula *WHERE* e especificar claramente quais registros devem ser apagados. (DIDÁTICA TECH, 2024)

1.6.5 – Relacionamentos

Os relacionamentos de banco de dados são associações entre tabelas que desempenham um papel crucial na organização e eficiência do sistema. Eles permitem a recuperação integrada de dados por meio de instruções de junção (*JOIN*), evitando a repetição desnecessária de informações e promovendo a normalização e eficiência do armazenamento. Além disso, garantem a integridade referencial, assegurando que as conexões entre tabelas sejam consistentes e válidas, evitando dados órfãos ou inconsistentes. Dessa forma, os relacionamentos contribuem para a estruturação adequada do banco de dados e para a execução de consultas complexas de maneira eficaz. (IBM, 2025)

Entre os principais tipos de relacionamento estão: o Um para Um (1:1), onde cada registro de uma tabela está ligado a um único registro de outra tabela, ideal para separar informações sensíveis ou específicas; o Um para Muitos (1:N), em que um registro de uma tabela pode se relacionar com múltiplos registros de outra, comum em casos como clientes e pedidos; e o Muitos para Muitos (N:N), que envolve múltiplos registros de ambas as tabelas, geralmente gerenciado por uma tabela intermediária, como ocorre em alunos e cursos. Esses relacionamentos são essenciais para evitar redundância de dados, estruturar bancos de dados de maneira organizada e executar consultas complexas com eficiência.

1.7 – PYTHON

Python é uma linguagem (de código aberto) de programação, esta trajetória de sucesso teve seu início no final dos anos 1980, tendo como ideia ser orientada a objetos fácil de interpretar e de alto nível com sintaxe fácil de ler, conhecida por sua simplicidade, legibilidade e versatilidade. Ideal para prototipagem e tarefas *ad-hoc*, o Python tem amplo uso em computação científica, desenvolvimento *web* e automação. Como uma linguagem de programação de uso geral e amigável para iniciantes, o Python suporta muitos dos principais cientistas da computação e desenvolvedores de aplicativos globalmente. (ORACLE,2025)

Sua sintaxe é clara e intuitiva, facilitando tanto o aprendizado de iniciantes quanto o trabalho de desenvolvedores mais experientes, podendo assim ser usado por qualquer pessoa. Além disso o uso da linguagem é registrado em diversos campos, incluindo ciências e dados, Inteligência Artificial, funcionando em diversos sistemas operacionais como Windows, Linux e macOS, o que o torna uma escolha prática para desenvolvedores que trabalham em ambientes diversos. Ele também possui uma vasta comunidade de usuários e desenvolvedores que contribuem para seu crescimento contínuo, criando bibliotecas e *frameworks* úteis. Essa comunidade é um dos pilares de seu sucesso, já que novos usuários encontram suporte e recursos facilmente disponíveis.

No campo de computação científica e Inteligência Artificial (IA), o Python desempenha um papel fundamental graças a bibliotecas como NumPy, Pandas e TensorFlow, que facilitam cálculos avançados, manipulação de dados e desenvolvimento de modelos de IA. O Python também é uma escolha natural para automação, desde tarefas simples, como renomear arquivos em massa, até orquestração de processos complexos.

Seu *design* orientado a objetos permite a reutilização de código por meio de conceitos como classes e herança, promovendo a eficiência e manutenção do código. Além disso, o Python é ideal para desenvolvimento *web*, com *frameworks* como Django e Flask, que possibilitam a criação de aplicações robustas e escaláveis. A Figura 7 mostra o logotipo oficial da linguagem Python, composto por dois elementos gráficos semelhantes a serpentes estilizadas, entrelaçadas nas cores azul e amarela, formando uma espiral que representa dinamismo e continuidade. Abaixo do símbolo está o nome “Python” escrito em uma tipografia simples e moderna.

Versões:

2.0 (2000): Essa versão teve grande impacto ao popularizar a linguagem. Além de apresentar os *list comprehensions*, ela possibilitou a criação de códigos mais concisos e expressivos. Contudo, uma limitação significativa foi sua abordagem a *strings* e *caracteres*, que não oferecia suporte nativo a Unicode, gerando complicações com textos multilíngues. (AWARI, 2025)

3.0 (2008): Foi um marco por resolver problemas fundamentais e modernizar a linguagem. Além do suporte total ao Unicode, o Python 3 fez mudanças estruturais

que garantiram uma melhor base para o futuro da linguagem. Com foco na clareza e simplicidade, ele ajudou os desenvolvedores a escreverem códigos mais limpos e robustos. (AWARI, 2025)

3.5 (2015): O suporte a *Async/Await* revolucionou o Python, permitindo a criação de aplicativos assíncronos e aumentando a eficiência em tarefas como comunicação entre servidores e manipulação de eventos. (AWARI, 2025)

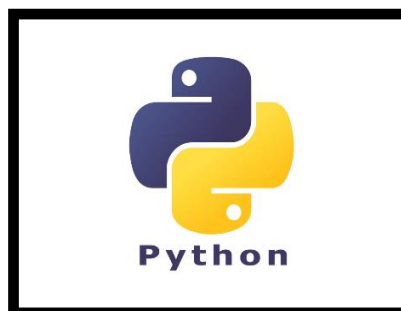
3.6 (2016): As *f-strings* são consideradas uma das adições mais populares, pois simplificaram a formatação de *strings* com uma sintaxe muito intuitiva. Outra melhoria foi o desempacotamento avançado de iteráveis, que aumentou a flexibilidade no processamento de dados. (AWARI, 2025)

3.7 (2018): Além das *dataclass*, que facilitaram a criação de estruturas de dados, essa versão também trouxe aprimoramentos nas anotações de tipos, promovendo uma maior integração com ferramentas de análise de código. (AWARI, 2025)

3.8 (2019): A introdução do operador tornou algumas construções de código mais elegantes. Ele permite atribuições dentro de expressões, reduzindo a necessidade de declarações redundantes. Os separadores numéricos foram outra grande vantagem, ajudando os desenvolvedores a lidarem com números grandes com mais clareza. (AWARI, 2025)

3.9 (2020): O operador de união “|” para conjuntos trouxe uma abordagem mais intuitiva e elegante para operações em coleções. Além disso, melhorias no desempenho de dicionários permitiram maior eficiência na manipulação de dados. (AWARI, 2025)

Figura 7 - Python



FONTE: GSS-TECHNOLOG, 2025

1.7.1 – MediaPipe

O MediaPipe, é uma biblioteca de código aberto, desenvolvida pelo Google, que simplifica a implementação de soluções de visão computacional e aprendizado de máquina em diferentes plataformas. Ela disponibiliza ferramentas avançadas e modelos pré-treinados que tornam possível realizar tarefas como rastreamento de mãos, análise de poses corporais, reconhecimento facial e até mesmo segmentação de imagens. Graças à sua capacidade de processamento em tempo real, o MediaPipe, é utilizado em projetos que necessitam de eficiência e personalização.

Entre os principais recursos que a biblioteca oferece estão a detecção de objetos, essencial para identificar e classificar elementos visuais, e o rastreamento de gestos, ideal para interfaces interativas baseadas no controle por movimentos. Além disso, a segmentação de imagens permite separar objetos ou áreas específicas em fotos e vídeos, enquanto a análise de áudio viabiliza tarefas como classificação de sons e detecção de fala. O MediaPipe também é amplamente reconhecido por seu suporte a aprendizado de máquina, permitindo personalizações por meio de ferramentas como o Model Maker, além de oferecer uma interface intuitiva com o MediaPipeStudio, que possibilita avaliar soluções diretamente no navegador. A Figura 8 apresenta o logotipo do MediaPipe, composto por ícones verticais estilizados em azul turquesa que remetem a sinais de processamento digital ou barras de sinal, seguidos do nome da biblioteca em fonte moderna e limpa. O *design* visual comunica tecnologia e fluidez, refletindo os princípios de desempenho em tempo real e inovação que a biblioteca oferece.

Figura 8 - MediaPipe



FONTE: PINTEREST, 2025

1.8 – LÓGICA DE INTELIGÊNCIA ARTIFICIAL

A inteligência artificial (IA), lançada na década de 1950, refere-se ao desenvolvimento de sistemas de computador que podem executar tarefas que normalmente exigem inteligência humana. (IBERDROLA,2025)

Essas tarefas incluem aprender com a experiência (aprendizado de máquina), entender a linguagem natural, reconhecer padrões, resolver problemas e se adaptar às situações de mudança. Os sistemas de IA podem ser categorizados em dois tipos principais, a IA estreita ou fraca, projetada para tarefas específicas e IA geral ou forte, que possuiria habilidades cognitivas do tipo humano em uma ampla gama de tarefas.

A IA é um campo amplo que abrange muitas disciplinas diferentes, como ciência da computação, estatísticas e análises de dados, engenharia de *hardware* e *software*, linguística, neurociência, filosofia e psicologia, sendo utilizado em vários campos, incluindo saúde, finanças, educação e muito mais. Em um nível operacional para uso comercial, a IA é um conjunto de tecnologias baseadas principalmente em *Machine Learning* e aprendizado profundo, usada para análise de dados, previsões e previsão, categorização de objetos, processamento de linguagem natural, recomendações, recuperação inteligente de dados e muito mais.

1.8.1 – *Machine Learning*

Machine Learning (ou Aprendizado de Máquina) é um ramo da Inteligência Artificial (IA) dedicado ao desenvolvimento de sistemas que possuem a habilidade de aprender e tomar decisões baseadas em dados, reconhecer padrões e fazer com que computadores aprendam com modelos gerando assim *insights* inteligentes sem a necessidade de pré-programação. Em vez de serem explicitamente programados para executar uma tarefa específica, os modelos de aprendizado de máquina processam grandes volumes de informações, identificam padrões e utilizam esses padrões para fazer previsões ou escolhas de forma autônoma. Logo, ao serem expostas a novos dados, elas se adaptam a partir dos cálculos anteriores e os padrões se moldam para oferecer respostas confiáveis. (SALESFORCE, 2025).

Na prática, o *Machine Learning* possui 4 tipos de variações, que definem diferentes tipos de aprendizado. O aprendizado Supervisionado utiliza um conjunto de dados rotulados para treinamento, ou seja, cada exemplo de treinamento é

acompanhado de uma resposta correta. O Não Supervisionado o conjunto de dados não são rotulados, o que significa que não há respostas corretas fornecidas. Já o Semi Supervisionado utiliza características do aprendizado supervisionado e não supervisionado, tendo como objetivo informações tanto das amostras rotuladas quanto das não rotuladas para melhorar o desempenho geral do modelo. O Aprendizado por Reforço os algoritmos aprendem através da interação contínua com um ambiente, o objetivo do agente é aprender a tomar decisões que maximizem a recompensa ao longo do tempo. (SALESFORCE, 2025).

1.8.2 – LangFlow

O LangFlow, lançado em 2023, revolucionou a forma como os LLMs (*Large Language Models* – Modelos de Linguagem de Grande Escala), como GPT-3 (*Generative Pre-Trained Transformer-3* – Transformadores Generativos Pré-Treinados-3) e GPT-4 (*Generative Pre-Trained Transformer-4* – Transformadores Generativos Pré-Treinados-4), são integrados a fluxos de trabalho dinâmicos e complexos. Usando a API da OpenAI, ele facilita tarefas como geração de textos, análise de dados e automação de sistemas, promovendo personalização e eficiência. (DATA HACKERS NEWSLETTER, 2025)

Fundamentado na linguagem Python, o LangFlow, aproveita bibliotecas poderosas como TensorFlow, PyTorch, Transformers e Langchain, que tornam a interação com modelos de linguagem mais acessível e robusta. Sua estrutura modular é composta por *prompts*, *chains* e *agents*, que organizam e manipulam dados de maneira eficiente, permitindo integração simples com APIs externas e dados em tempo real. Essa integração enriquece os fluxos de trabalho e viabiliza soluções adaptáveis em diferentes setores.

Além disso, o LangFlow se destaca por suas ferramentas de depuração e otimização, que tornam o desenvolvimento de fluxos de trabalho complexos mais prático. Ele identifica falhas e aprimora o desempenho continuamente, reduzindo tempos de resposta e aumentando a eficiência dos processos automatizados. Suas aplicações incluem atendimento ao cliente, sistemas de recomendação e geração de relatórios empresariais, demonstrando impacto significativo em organizações que buscam automação avançada. A plataforma conta com extensa documentação e suporte comunitário, com tutoriais e exemplos práticos que ajudam desenvolvedores a explorar todo o seu potencial. Esse ecossistema colaborativo não apenas facilita o aprendizado, mas também impulsiona inovações através da troca de ideias e experiências. A Figura 9 apresenta o logotipo do LangFlow, formado por elementos gráficos que remetem a cadeias de DNA (*Deoxyribonucleic Acid* – Ácido

Desoxirribonucleico) estilizadas à esquerda, representando a ideia de fluxo e inteligência conectada, seguidos pela tipografia do nome em tons azulados. O *design* evoca modernidade, modularidade e inovação tecnológica, características centrais da plataforma.

Figura 9 - *LangFlow*



FONTE: PINTEREST, 2025

1.9 – SERVIDOR

Um servidor é um computador equipado com um ou mais processadores, bancos de memória, portas de comunicação, *softwares* e, ocasionalmente, algum sistema para armazenamento de dados, como *hard disks* internos ou memórias SSD (*Solid State Drive* – Unidade de Estado Sólido). (CONTROLE.NET, 2023)

Um servidor é um recurso essencial dentro de um sistema computacional, projetado para processar aplicações, prestar serviços e armazenar dados de maneira eficiente. Esses sistemas podem ser físicos, na forma de equipamentos dedicados exclusivamente a essas funções, ou virtuais, operando como instâncias de *software* dentro de infraestruturas maiores. Eles podem ser instalados localmente, em instalações próprias, ou remotamente, em servidores na nuvem, e são reconhecidos por sua escalabilidade e alto poder de processamento, o que os torna capazes de lidar com grandes volumes de trabalho e múltiplas solicitações simultâneas.

Os servidores são configurados para executar programas ou protocolos específicos que permitem fornecer serviços a outras máquinas ou clientes, garantindo a comunicação e a funcionalidade em ambientes computacionais. Eles desempenham um papel fundamental em redes LAN (*Local Area Network* – Rede Local) ou WAN (*Wide Area Network* – Rede de Longa Distância), viabilizando desde o compartilhamento de arquivos até o suporte a aplicações empresariais críticas.

Além disso, servidores são projetados para operar continuamente, 24 horas por dia, com alta confiabilidade e redundância, para evitar interrupções nos serviços. Eles podem ser utilizados em uma ampla variedade de aplicações, incluindo hospedagem de *sites*, gerenciamento de bancos de dados, serviços de *e-mail*, armazenamento em nuvem, computação distribuída e muito mais. A arquitetura de servidores modernos frequentemente inclui recursos avançados como virtualização, balanceamento de carga e sistemas de *backup*, que otimizam sua eficiência e garantem sua segurança.

1.9.1 – Servidor Web

Um servidor *web*, ou "*web server*", refere-se tanto à parte física quanto à parte lógica de um sistema que opera em conjunto para disponibilizar conteúdos na *internet*. No âmbito do *hardware*, um servidor *web* é essencialmente um computador projetado para armazenar os arquivos que compõem um *site*, como documentos HTML, imagens, folhas de estilo e arquivos JavaScript. Este computador está conectado à *internet* e pode ser acessado por meio de um nome de domínio (DNS – *Domain Name System* – Sistema de Nomes de Domínio). Por exemplo, "mozilla.org" é um nome de domínio que direciona os usuários ao servidor que hospeda os arquivos do *site* correspondente.

Já no contexto do *software*, um servidor *web* abrange diversos componentes responsáveis por regular como os usuários acessam os arquivos armazenados. Entre esses componentes, destaca-se o servidor HTTP que é um *software* programado para interpretar URLs (*Uniform Resource Locators* – Localizadores Uniformes de Recursos) e HTTP, o protocolo usado pelos navegadores para exibir páginas *web*. O servidor HTTP desempenha um papel crucial ao servir como intermediário na entrega de informações entre o servidor e o dispositivo do usuário.

1.9.2 – Servidor Local

Os servidores locais, por sua vez, são máquinas robustas que permanecem fisicamente localizadas em empresas ou residências. Eles são utilizados para atender dispositivos em uma rede local (LAN) e podem fornecer acesso remoto via *internet*, ideal para empresas que desejam maior controle sobre seus dados e segurança. Com a evolução da tecnologia, substituíram os antigos *mainframes*, oferecendo mais flexibilidade, escalabilidade e economia.

Não há exatamente um marco do surgimento dos servidores em termos de computação, da mesma forma como os primeiros computadores são resultado de uma série de ações de empresas como Apple e IBM e não há um “marco de fundação”. Mas de forma geral os primeiros servidores surgiram com o aumento do uso de redes de computadores e assim, começou-se a imaginar que seria produtivo e útil destinar alguns computadores da rede para usos específicos.

Embora redes de computadores já existiam antes da Arpanet (1969) a qual foi o embrião do que se conhece hoje como *internet*, foi só a partir de então que a adoção de redes com propósitos primeiramente acadêmicos e depois comerciais, ganhou corpo. Paralelamente a isso, o modelo antigo que se baseava em *mainframes* que eram grandes computadores e terminais, começou a ficar obsoleto e os servidores ganharam mais e mais importância. (HOSTMÍDIA, 2023)

1.10 – NAVEGADORES

Navegadores são programas que permitem aos usuários interagirem com documentos HTML armazenados em servidores *web*. Eles desempenham um papel essencial na navegação e exibição de conteúdo na *internet*, proporcionando acesso fácil e intuitivo a informações digitais.

A história dos navegadores remonta a 1990, quando o cientista britânico Tim Berners-Lee criou o primeiro servidor *web* e o primeiro navegador gráfico, chamado "*WorldWideWeb*", enquanto trabalhava no CERN, na Suíça. (MOZILLA,2025)

Essa invenção marcou o início da interação visual com a *internet*, sendo um passo significativo para o desenvolvimento da *web* como conhecemos hoje. Na evolução dos navegadores, destacou-se o lançamento do Google Chrome em 2008. (MOZILLA,2025)

Este navegador introduziu várias inovações, como a navegação em abas, carregamento rápido de páginas e suporte aos padrões W3C. O Chrome rapidamente conquistou uma grande base de usuários, ajudado pela introdução de complementos que expandiam suas funcionalidades. Com isso, ele rivalizou diretamente com o Internet Explorer e o Mozilla Firefox, dois grandes nomes da época.

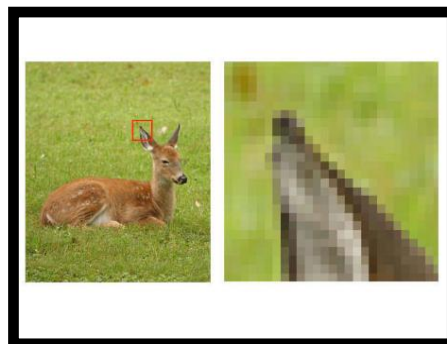
Em 2021, o Chrome liderava o mercado, usado por mais da metade da população conectada. Sua hegemonia estimulou a concorrência, impulsionando o avanço de tecnologias para sites mais sofisticados. (MOZILLA,2025)

1.11 – IMAGENS DIGITAIS

Imagens digitais são representações de dados visuais em forma de matrizes de números binários. Estas imagens são capturadas por sensores como CCD (*Charge-Coupled Device* – Dispositivo de Carga Acoplada), que transformam a luz em informações digitais. A resolução e qualidade dessas imagens dependem diretamente do número e disposição de *pixels*, ou seja, pequenas unidades que compõem a imagem. A relação de largura e altura dos *pixels* determina a chamada razão de aspecto.

Além disso, as imagens digitais podem ser salvas em diferentes formatos, cada qual com propósitos específicos. O formato PNG é conhecido por sua capacidade de preservar qualidade com compressão sem perdas e pela transparência de fundo. Já o formato TIFF é amplamente utilizado em aplicações profissionais como impressão, devido à sua alta qualidade e suporte a grandes tamanhos de arquivo. A Figura 10 ilustra um exemplo de imagem digital composta por dois quadros: à esquerda, a imagem de um animal em um ambiente natural, e à direita, o mesmo recorte da imagem ampliado, revelando a estrutura em *pixels*. Essa representação mostra como os detalhes visuais são armazenados digitalmente por meio de pequenas unidades coloridas. Esse processo é fundamental para a visualização de imagens em dispositivos eletrônicos, sendo a base para operações de processamento e análise digital.

Figura 10 - Exemplo de Imagem Digital



FONTE: ANIMATION, 2024

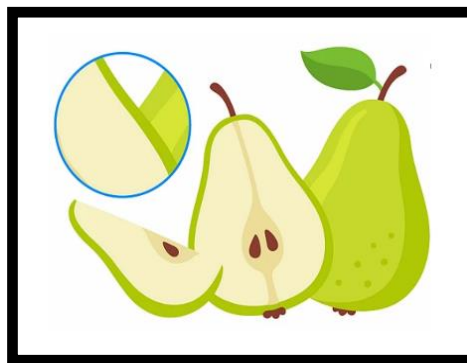
1.11.1 – Imagens Vetoriais

As imagens vetoriais são representações gráficas compostas por elementos matemáticos como pontos, linhas, curvas e formas. Diferentemente das imagens *raster*, que são constituídas por *pixels*, as imagens vetoriais mantêm sua qualidade independentemente do tamanho ou escala, a Figura 11 ilustra um exemplo claro desse tipo de imagem, mostrando ilustrações de peras com detalhes suaves e bem definidos, inclusive quando ampliados. Isso as torna ideais para projetos gráficos que requerem flexibilidade, como logotipos, ícones, ilustrações e *designs* digitais.

A principal vantagem das imagens vetoriais está na sua capacidade de preservar detalhes e nitidez quando redimensionadas, sendo amplamente utilizadas em trabalhos que precisam ser aplicados em diferentes mídias e tamanhos. Alguns dos formatos mais comuns para armazenar imagens vetoriais são SVG, conhecido por sua capacidade de escalabilidade, e EPS (*Encapsulated PostScript* – PostScript Encapsulado), usados frequentemente em impressões e projetos profissionais.

Existem várias plataformas que disponibilizam imagens vetoriais gratuitas, como Depositphotos, Freepress e Vecteezy. Elas oferecem uma ampla variedade de ilustrações e *clipart* livres de *royalties* para uso pessoal e comercial. Além disso, esses *sites* possuem milhões de recursos acessíveis em diferentes formatos, promovendo praticidade e qualidade para *designers* e criadores.

Figura 11 - Exemplo de Imagem Vetorial



FONTE: ANIMATION, 2024

1.12 – DESIGN

Design é a idealização, criação, desenvolvimento, configuração, concepção, processamento e especificação de produtos ou sistemas, frequentemente destinados à produção em massa ou fabricação industrial. O *design* abrange a padronização de

componentes e a aplicação de soluções criativas para atender às necessidades estéticas, funcionais, práticas e econômicas de usuários e mercados específicos.

Além disso, o *design* não se limita apenas à criação de produtos tangíveis, mas também abrange serviços, interfaces digitais e UX (*User Experience* – Experiência do Usuário), onde o objetivo é garantir usabilidade, acessibilidade e eficiência. No caso de produtos industriais, o *design* padronizado é crucial para facilitar a produção em larga escala, reduzir custos e garantir consistência e qualidade.

Outro aspecto importante do *design* é a sua interdisciplinaridade, envolvendo conhecimentos de áreas como ergonomia, sustentabilidade, psicologia e até mesmo engenharia. O processo de *design* é iterativo, começando com pesquisa e análise de demandas, seguido pela geração de ideias (*brainstorming*), prototipagem, testes e ajustes até se chegar a um resultado final otimizado.

1.12.1 – Design de Interfaces

O *design* de interface é um elemento essencial para conectar o ser humano com produtos físicos ou virtuais, abrangendo uma ampla variedade de interações, como com brinquedos, aparelhos eletrodomésticos, aplicativos para *smartphone* e páginas *web*. Ele proporciona uma conexão intuitiva e funcional entre os usuários e os sistemas, garantindo uma experiência eficaz.

O papel do *designer* de interface vai além de compreender os problemas e as necessidades dos usuários. Esse tipo de *design* combina conhecimentos técnicos e estéticos para desenvolver ferramentas funcionais e soluções visuais atraentes que otimizam a interação entre o usuário e o produto. Na prática, o *design* de interface envolve aspectos como a parte visual, usabilidade, arquitetura da informação, navegação e transições de telas. Todos esses componentes são projetados para aprimorar a interação e facilitar o uso de sistemas e aplicações.

A experiência do usuário, ou UX (*User Experience* - Experiência do Usuário), ocupa um papel central no *design* de interface. Projetar um produto, serviço ou sistema de qualidade requer antecipação das expectativas e necessidades do público. O *designer* de interface deve constantemente avaliar se a usabilidade da aplicação é satisfatória e apropriada para o público-alvo, buscando soluções inovadoras e práticas.

1.12.2 – Canva

O Canva é uma plataforma de *design* gráfico *online* e gratuita, criada em 2013 para assegurar que qualquer pessoa crie seus *designs* e publique em qualquer lugar, pode ser utilizada para criar uma variedade de conteúdos visuais, como apresentações, cartazes, logotipos, *posts* para redes sociais, vídeos e muito mais. Possui uma interface intuitiva e amigável, ela oferece uma ampla variedade de recursos e *templates* incríveis, permitindo que criar *designs* impressionantes com facilidade e rapidez. (CANVA, 2025)

Com essa ferramenta, tudo fica ao alcance dos seus dedos. Seja para criar um logotipo único, um convite personalizado e deslumbrante ou um *banner* atrativo, as possibilidades são infinitas. Além disso, o Canva oferece uma vasta biblioteca de elementos gráficos, imagens de alta qualidade e fontes elegantes, garantindo que sempre tenha as ferramentas certas para expressar a criatividade e transmitir a mensagem de forma a impactar quem visualiza.

1.12.3 – Inkscape

O Inkscape, é um programa de desenho vetorial de qualidade profissional que roda em Windows, MacOS X e GNU/Linux. É usado por entusiastas e profissionais de *design* em todo o mundo, para a criação de uma grande variedade de gráficos, tais como ilustrações, ícones, logotipos, diagramas, mapas e gráficos da *web*. O Inkscape usa o padrão aberto SVG (*Scalable Vector Graphics* - Gráficos Vetoriais Escaláveis) do consórcio W3C como seu formato nativo e é uma aplicação de uso livre e de código aberto.

O Inkscape tem ferramentas sofisticadas de desenho com recursos comparáveis ao Adobe Illustrator, CorelDRAW e Xara Xtreme. Ele pode importar e exportar diversos formatos de arquivo, incluindo SVG (*Scalable Vector Graphics* - Gráficos Vetoriais Escaláveis), AI (*Artificial Intelligence* – Inteligência Artificial), EPS (*Encapsulated PostScript* – Pós-Script Encapsulado), PDF (*Portable Document Format* – Formato de Documento Portátil), PS (*PostScript* – Pós-Script) e PNG (*Portable Network Graphics* - Gráficos de Rede Portáteis). Tem um grande conjunto de funcionalidades, uma interface simples, suporte a multi-idiomas e foi projetado para ser extensível; os usuários podem personalizar as funcionalidades do Inkscape com extensões.

Versões:

0.35 (2003): Esta foi a primeira versão oficial do Inkscape, lançada como um *fork* do projeto Sodipodi. Focada em conformidade com o padrão SVG, estabeleceu

as bases para o desenvolvimento futuro, incluindo funcionalidades básicas de *design* vetorial e uma interface simples. (BLOG DO EDIVALDO,2025)

0.36 a 0.39 (2004): Durante este período, foram realizadas melhorias na interface do usuário e adicionadas funcionalidades, como suporte a marcadores e preenchimentos de padrões, aprimorando a experiência do *designer*. (BLOG DO EDIVALDO,2025)

0.40 (2005): Introduziu camadas, permitindo que *designers* organizassem elementos diferentes em seus projetos. Também implementou máscaras para efeitos visuais e melhorias na exportação de arquivos, incluindo suporte inicial ao formato PDF. (BLOG DO EDIVALDO,2025)

0.41 (2005): Focou em aprimorar recursos introduzidos anteriormente, com ajustes de estabilidade e refinamento nas ferramentas de camadas e máscaras. (BLOG DO EDIVALDO,2025)

0.42 (2005): Trouxe avanços na compatibilidade com o padrão SVG e melhorias na interface, além de suporte expandido para guias e grades. (BLOG DO EDIVALDO,2025)

0.43 (2006): Refinou o suporte à exportação de PDFs, oferecendo opções avançadas de qualidade. Também aprimorou ferramentas de caminho e transformações de formas. (BLOG DO EDIVALDO,2025)

0.44 (2006): Introduziu gradientes personalizados e manipulação avançada de objetos, além de ajustes de estabilidade e suporte para múltiplos idiomas, expandindo o alcance global do *software*. (BLOG DO EDIVALDO,2025)

0.45 (2007): Apresentou os efeitos de caminho, permitindo transformações avançadas de formas e edição criativa. Houve suporte inicial a fontes SVG, oferecendo mais opções de tipografia. (BLOG DO EDIVALDO,2025)

0.46 (2008): Melhorias na renderização, aprimoramentos nos gradientes e introdução de novos modos de edição de nós. (BLOG DO EDIVALDO,2025)

0.47 (2009): Adicionou ferramentas de criação mais fluidas e opções para *designers* com necessidades específicas, como a manipulação de texto mais integrada. (BLOG DO EDIVALDO,2025)

0.48 (2010): Trouxe suporte expandido para *spray* e borracha, além de refinamentos na funcionalidade geral, tornando o *software* mais versátil para projetos variados. (BLOG DO EDIVALDO,2025)

0.91 (2015): Uma versão estável e um marco importante, introduzindo o mecanismo de renderização Cairo para melhor desempenho e qualidade gráfica. (BLOG DO EDIVALDO,2025)

0.92 (2017): Concentrou-se em gráficos de alta resolução e unidades personalizadas, expandindo as ferramentas para precisão no *design*. (BLOG DO EDIVALDO,2025)

1.0 (2020): Primeira versão completa do Inkscape, com uma interface modernizada, suporte para telas HiDPI (*High Dots Per Inch* - Telas de Alta Densidade de *Pixels*) e uma experiência de usuário mais fluida. (BLOG DO EDIVALDO,2025)

1.1 (2021): Refinou as mudanças da versão 1.0, com diálogos reorganizados e integração aprimorada com sistemas externos. (BLOG DO EDIVALDO,2025)

1.2 (2022): Expandiu o suporte para múltiplas páginas e introduziu ferramentas colaborativas, sendo útil para projetos mais complexos. (BLOG DO EDIVALDO,2025)

1.3 (2023): Focou em melhorias de desempenho e manipulação avançada de texto e formas geométricas, facilitando a criatividade no *design*. (BLOG DO EDIVALDO,2025)

1.4 (2024): Versão mais recente, trazendo filtros aprimorados, suporte para novos formatos de arquivos e ferramentas avançadas para *grids* modulares e axonométricas. (BLOG DO EDIVALDO,2025)

1.12.4 – Figma

O Figma, foi lançado em 2016, é uma plataforma colaborativa de *design* que reúne elementos e funcionalidades cruciais para o desenvolvimento de produtos digitais: *Design UI* e *Design UX*. Por meio dela, é possível desenvolver componentes de *design*, *wireframes*, criar interfaces digitais e protótipos interativos de produtos com foco em usabilidade e experiência do usuário. (3PM3,2025)

A ferramenta oferece recursos como prototipação interativa, para simular o funcionamento do *design* antes da codificação; *AutoLayout*, que ajusta automaticamente elementos do *design*; e bibliotecas reutilizáveis, permitindo criar componentes consistentes em projetos grandes. Além disso, o Figma suporta diversos *plugins* para adicionar imagens, ícones e conteúdo de forma rápida. (FIGMA, 2025)

Uma das principais vantagens do Figma é sua acessibilidade: ele funciona diretamente no navegador ou em aplicativos para Windows e Mac, sem necessidade de instalação complexa. O Figma não possui versões numeradas tradicionais, como "1.1" ou "2.0", porque ele é baseado na nuvem e atualizado continuamente. Desde o seu lançamento, a plataforma recebe melhorias e novos recursos automaticamente, sem a necessidade de *downloads* ou instalações manuais. (CURSO DE DISEÑO GRÁFICO ONLINE, 2025)

1.12.5 – GIMP

O GIMP (GNU *Image Manipulation Program* - Programa de Manipulação de Imagens do GNU) lançado em 1995, é um *software* de edição de imagens gratuito e de código aberto. (GIMP,2025)

É amplamente utilizado para diversas finalidades, incluindo retoque de fotos, criação de artes digitais, *design* gráfico e produção de elementos visuais para redes sociais ou *websites*. Uma característica marcante do GIMP é sua capacidade de equilibrar robustez e simplicidade, tornando-se uma excelente opção para estudantes, *designers* em início de carreira e até mesmo profissionais em busca de uma solução gratuita. A princípio era voltado apenas para Linux, mas hoje em dia está presente até para Mac. Conta com diversos recursos e é o mais reconhecido programa no segmento de editores de imagem gratuitos.

Entre suas principais características, destaca-se a interface personalizável, que permite aos usuários organizar as ferramentas de acordo com suas necessidades. O GIMP oferece suporte para uma ampla gama de formatos de arquivos, como JPEG (*Joint Photographic Experts Group* – Grupo Conjunto de Peritos Fotográficos), PNG, GIF (*Graphical Interchange Format*– Formato de Intercâmbio de Gráficos), TIFF (*Tag Image File Format* – Formato de Arquivo de Imagem com Marcação) e até mesmo arquivos PSD (*Photoshop Document* – Documento do Photoshop) do Photoshop. Sua extensibilidade é outro ponto forte, possibilitando a

integração com *plugins* e *scripts* em linguagens como Python e Perl, o que aumenta significativamente suas funcionalidades. Ferramentas avançadas, como máscaras de camada, gradientes, pincéis personalizáveis e filtros, atendem tanto amadores quanto profissionais.

1.13 – FORMATAÇÃO DE VÍDEOS

Um formato de arquivo de vídeo é representado pela extensão que aparece no final de um arquivo, como MP4 ou AVI (*Audio Video Interleave* – Intercalação de Áudio e Vídeo).

Cada formato possui características próprias, variando em aspectos como tamanho do arquivo, métodos de compressão, tipos de codificação e compatibilidade com diferentes dispositivos ou programas. Em alguns projetos, priorizar o formato com menor tamanho pode ser ideal, enquanto em outros, a qualidade de resolução e a fidelidade do vídeo podem ter mais importância.

1.13.1 – MP4

O MPEG-4 (*Moving Picture Experts Group-4* – Grupo de Especialistas em Imagens em Movimento-4) ou MP4 como é mais conhecido, é um padrão amplamente utilizado para compressão de áudio e vídeo. Ele é conhecido por sua eficiência em oferecer alta qualidade de vídeo com tamanhos de arquivo relativamente pequenos, o que o torna ideal para a *web* e também para transmissões de televisão. O formato MP4, derivado do MPEG-4, é amplamente aceito como padrão para vídeos *online*, sendo uma escolha confiável para plataformas como YouTube ou para incorporar vídeos em *sites*.

Uma das poucas desvantagens do MP4 é que os processos de codificação (compressão) e decodificação (descompressão) exigem bastante poder de processamento. Embora os arquivos sejam compactos, a compressão durante o salvamento pode sobrecarregar o computador. Além disso, ao assistir a esses vídeos, o dispositivo precisa descomprimir os dados em tempo real para garantir a melhor qualidade possível.

2 – ESTUDO DE CASO

O estudo de caso visa proporcionar uma análise branda e detalhada referente ao desenvolvimento da aplicação voltada ao ensino e reconhecimento da Língua Brasileira de Sinais (Libras). Foram analisadas questões de viabilidade técnica, social e educacional, bem como sua abrangência e impacto na acessibilidade para a comunidade surda no Brasil.

Apesar de ser oficialmente reconhecida como uma das línguas do país, a Libras ainda enfrenta inúmeros desafios quanto à sua difusão, ensino e acessibilidade. Grande parte da população ouvinte desconhece os sinais básicos da língua, o que acarreta barreiras de comunicação, especialmente em ambientes educacionais, comerciais e de serviços públicos. Além disso, o número de intérpretes e profissionais capacitados é insuficiente para atender toda a demanda, e os recursos disponíveis para o aprendizado de Libras são, em sua maioria, restritos ou pouco interativos.

Diante dessa realidade, desenvolveu-se o projeto de um aplicativo educacional que não só ensina Libras de forma interativa, como também capta os sinais feitos pelo usuário por meio da câmera, fornecendo *feedback* em tempo real sobre a execução correta dos sinais. A proposta visa unir acessibilidade, tecnologia e educação, promovendo inclusão social e aprendizado prático, dinâmico e acessível a qualquer pessoa com um *smartphone*.

Durante a análise do protótipo, identificaram-se possíveis problemas relacionados à precisão no reconhecimento dos sinais captados pela câmera, que pode variar conforme a iluminação, ângulo ou qualidade da câmera do dispositivo. Outro desafio é a diversidade regional da Libras, que apresenta variações nos sinais dependendo da localidade, podendo gerar inconsistências no reconhecimento. Além disso, observou-se a necessidade de garantir uma base de dados ampla e representativa dos sinais oficiais, o que demanda parcerias com profissionais fluentes em Libras.

Para resolver esses problemas, propõem-se constantes atualizações no banco de dados de sinais, treinamento contínuo de modelos de inteligência artificial com base em *Machine Learning* e *Deep Learning*, testes de usabilidade com usuários surdos e ouvintes, além de investimentos em segurança de dados, principalmente no que diz respeito à privacidade das imagens captadas. O aplicativo também poderá

funcionar parcialmente *offline*, com módulos de ensino que não exigem conexão, permitindo que o usuário estude mesmo em locais sem acesso à *internet*.

O objetivo final é garantir que o aplicativo seja funcional, intuitivo, acessível e eficaz no ensino de Libras, contribuindo para uma sociedade mais inclusiva e preparada para se comunicar com a comunidade surda.

2.1 – VIABILIDADE

Um estudo de viabilidade de projeto é uma análise financeira de uma ideia ou um projeto. Este tipo de estudo avalia o potencial do projeto ser lucrativo ou não. E geralmente é feito em uma planilha, como uma de Excel. (TELES, 2024)

2.1.1 – Viabilidade Operacional

A viabilidade operacional está relacionada à capacidade de um produto ou negócio gerar receita suficiente para cobrir seus custos operacionais e ainda obter lucro. Em outras palavras, um negócio viável do ponto de vista operacional é aquele capaz de se sustentar financeiramente a longo prazo. (ESCOLA DNC, 2024)

Para desenvolver este projeto, a equipe precisa dominar várias tecnologias e ferramentas essenciais. É necessário o conhecimento pleno da linguagem HTML para a confecção da base estrutural do *site*, enquanto o CSS dará estilo e identidade visual. O conhecimento sobre o Bootstrap ajudará na criação de um *design* responsivo, e o JavaScript adicionará interatividade e animações. Com o conhecimento em *back-end*, o PHP permitirá a integração com o banco de dados, que será gerenciado por MySQL e MariaDB.

É necessário conhecimento em Python para desenvolver algoritmos de detecção de movimentos, com LangFlow auxiliando na construção de um *chat* informativo com IA. O conhecimento em OpenCV processará imagens da câmera, o MediaPipe identificará mãos nas imagens, e o TensorFlow treinará a máquina para reconhecer movimentos.

Além disso, o domínio de ferramentas como Visual Studio Code para edição de código, Figma para prototipação, GIMP e Canva para edição de imagens, Inkscape para criação vetorial, BrModelo para modelagem do banco de dados e XAMPP para testes em ambiente local será essencial. O *Teachable Machine* permitirá criar modelos de reconhecimento de posição das mãos para treinamento no TensorFlow.

2.1.2 – Viabilidade Técnica

A viabilidade técnica é a avaliação que determina se um projeto, produto ou sistema pode ser desenvolvido e implementado com os recursos tecnológicos,

operacionais e humanos disponíveis. Trata-se de uma etapa fundamental no planejamento, pois analisa a capacidade real de execução da proposta, considerando desde a infraestrutura necessária até o conhecimento técnico da equipe envolvida. (RYAZBEK,2022)

Essa análise verifica se as ferramentas, linguagens, equipamentos e sistemas já existentes são compatíveis e suficientes para sustentar o funcionamento da solução proposta. Além disso, leva em conta o tempo estimado para desenvolvimento, os possíveis desafios técnicos e a capacidade de manutenção futura. Ao comprovar que o projeto é tecnicamente possível, a viabilidade técnica contribui para reduzir riscos, otimizar recursos e garantir que as etapas seguintes sejam construídas sobre bases sólidas e realistas. (RYAZBEK,2022)

Itens Físicos:

Para a realização dos testes, é necessário dispor de computadores que possuam sistema operacional Windows 10 ou superior, garantindo compatibilidade com as ferramentas utilizadas. Além disso, é fundamental que os dispositivos tenham um navegador atualizado e acesso estável à *internet*, proporcionando um ambiente adequado para a execução dos procedimentos.

Também serão utilizados *smartphones* Android que atendam aos requisitos mínimos de 32GB de armazenamento e 3GB de RAM, permitindo o desempenho eficiente dos testes. Para a captura de sinais, serão empregadas câmeras *Web* ou as próprias câmeras dos *smartphones*, assegurando a qualidade na obtenção das imagens. O ambiente destinado aos testes deve ser preparado de maneira adequada, contando com iluminação suficiente para garantir a precisão na captura dos gestos, evitando interferências e melhorando a clareza dos registros. Esses fatores são essenciais para assegurar a eficácia dos processos e a obtenção de resultados confiáveis.

Medidas de segurança:

Para evitar interrupções nos testes, é essencial ter *backups* de equipamentos disponíveis para substituição imediata caso algum dispositivo apresente falhas. Além disso, o uso de câmeras alternativas garante que a captura de sinais ocorra sem comprometimento, mantendo a qualidade e a precisão dos registros.

Infraestrutura Digital:

Para a infraestrutura do sistema, será utilizado o MySQL como gerenciador de banco de dados, garantindo armazenamento eficiente e organização dos dados. A API do modelo de IA será hospedada em um serviço *web*, podendo ser Render, Heroku ou AWS, conforme a necessidade do projeto. O *front-end* do aplicativo *web* será disponibilizado por meio de plataformas como Vercel ou FirebaseHosting, proporcionando acessibilidade e desempenho na interface do usuário. Para o processamento da inteligência artificial, o modelo *Teaching Machine* será treinado e executado em um ambiente compatível com CUDA (*Compute Unified Device Architecture* – Arquitetura Unificada de Dispositivos de Computação).

O *Teaching Machine* é até o momento, a plataforma mais estável e confiável para colocar em prática. As versões mais novas ou alternativas, como outras plataformas emergentes, ainda são muito recentes, com pouca documentação e podem apresentar bugs ou problemas de compatibilidade. Possui integração completa com bibliotecas como o PyTorch e ferramentas associadas, facilitando bastante o desenvolvimento e o uso em diferentes ambientes. Além disso, existe muito mais material disponível na *internet* (exemplos, tutoriais, dúvidas já respondidas), o que ajuda muito no aprendizado e na solução de problemas.

Medidas de segurança:

Para assegurar a estabilidade e a continuidade do sistema, será mantido um *backup* da API em outro serviço de hospedagem, permitindo uma rápida recuperação em caso de falhas ou indisponibilidade do *host* principal. Além disso, serão implementados mecanismos de monitoramento e registros detalhados de atividades por meio de *logs*, possibilitando a identificação precoce de problemas e garantindo um funcionamento mais estável e eficiente. Essas medidas contribuem para minimizar impactos técnicos e assegurar a confiabilidade da infraestrutura.

Desenvolvimento e Tecnologias Utilizadas:

Para o desenvolvimento do sistema, será utilizada a linguagem de programação Python, oferecendo flexibilidade e eficiência na implementação dos algoritmos. Os *frameworks* de inteligência artificial incluem OpenCV, TensorFlow e *Teaching Machine*, que proporcionam ferramentas avançadas para processamento de

imagens e reconhecimento de padrões. O *back-end* será estruturado com Flask ou FastAPI, permitindo comunicação ágil e eficiente entre o modelo de IA e a interface do usuário.

No *front-end*, serão utilizados *React.js* ou *Next.js* para a construção de uma interface intuitiva e responsiva. Além disso, para a manipulação de imagens, OpenCV será empregado no pré-processamento dos sinais, garantindo uma análise precisa e otimizada. Essa combinação de tecnologias permite um desenvolvimento robusto e alinhado às demandas do projeto

Medidas de Contingência:

Para garantir a estabilidade e o desempenho do sistema, serão adotadas estratégias para lidar com possíveis desafios técnicos. Caso o reconhecimento de sinais não alcance um resultado satisfatório, será empregada a técnica de *Transfer Learning* ou a substituição por modelos alternativos que possam oferecer resultados mais precisos. Se houver falhas na infraestrutura de nuvem, a execução poderá ser realizada localmente em um computador com suporte a GPU, garantindo a continuidade do processamento sem comprometer o desempenho.

Além disso, para evitar perdas de dados, serão implementados *backups* automáticos regularmente, assegurando a integridade e a recuperação eficiente das informações armazenadas. Essas medidas são essenciais para manter a confiabilidade do sistema e minimizar impactos causados por imprevistos técnicos.

2.1.3 – Viabilidade de Cronograma

O cronograma elaborado para o trabalho mostrou-se fundamental para a organização das divisões de trabalho e definição dos prazos de entrega, visando otimizar o desempenho do grupo como um todo. Conforme demonstrado na Tabela 1, o projeto está estruturado com atribuições específicas para cada área de atuação, como *Design*, Programação e elaboração da Monografia.

Essa divisão de tarefas por áreas especializadas não apenas facilita a gestão e a execução do projeto, mas também promove uma distribuição equitativa de responsabilidades, garantindo um fluxo de trabalho eficiente e coordenado. Além disso, o cronograma proporciona uma visão clara dos prazos de entrega, permitindo

que o grupo acompanhe o progresso de cada etapa e faça ajustes conforme necessário para cumprir com êxito todas as metas estabelecidas. (LUCID,2025)

Tabela 1- Cronograma Documentação

Documentacao					
Introdução	Kimberly e mayra	100%	28/fev	11/abr	
Capitulo 1	Kimberly e mayra	100%	28/fev	11/abr	
Tecnologias	Kimberly e mayra	100%	28/fev	11/abr	
Capitulo 2	Kimberly e mayra	100%	18/abr	17/jun	
Viabilidade	Kimberly e mayra	100%	18/abr	17/jun	
Prototipagem	Kimberly e mayra	100%	18/abr	17/jun	
Área de atuação	Kimberly e mayra	100%	18/abr	17/jun	
Cliente	Kimberly e mayra	100%	18/abr	17/jun	
Leis e regras contratuais	Kimberly e mayra	100%	18/abr	17/jun	
Capitulo 3	Kimberly, Mayra e Natháli	0%	04/jul	16/set	
Documentação	Kimberly, Mayra e Natháli	0%	04/jul	16/set	
BPMN	Kimberly, Mayra e Natháli	0%	04/jul	16/set	
Diagrama	Kimberly, Mayra e Natháli	0%	04/jul	16/set	
Aplicação web	Kimberly, Mayra e Natháli	0%	04/jul	16/set	
Capitulo 4	Kimberly, Mayra e Natháli	0%	03/out	11/nov	
Considerações finais	Kimberly, Mayra e Natháli	0%	03/out	11/nov	
Referências bibliográficas	Kimberly, Mayra e Natháli	0%	03/out	11/nov	
Bibliografia	Kimberly, Mayra e Natháli	0%	03/out	11/nov	

FONTE: AUTORIA PRÓPRIA, 2025

Na tabela 1, é possível observar as tarefas a serem concluídas, seus responsáveis, o progresso relatado, a data de início e término da tarefa.

Tarefas Documentação:

1 – “Introdução” foi atribuída para os membros do grupo que cuidam da parte da Documentação, Mayra e Kimberly, com o prazo de 28/02/2025 a 11/04/2025

2 – “Capítulo 1” foi atribuída à Mayra e Kimberly, com o prazo de 28/02/2025 a 11/04/2025

3 – “Tecnologias” foi atribuída à Mayra e Kimberly, com o prazo de 28/02/2025 a 11/04/2025

4 – “Capítulo 2” foi atribuída à Mayra e Kimberly, com o prazo de 18/04/2025 a 17/06/2025

5 – “Viabilidade” foi atribuída à Mayra e Kimberly, com o prazo de 18/04/2025 a 17/06/2025

6 – “Prototipagem” foi atribuída à Mayra e Kimberly, com o prazo de 18/04/2025 a 17/06/2025

7 – “Área de atuação” foi atribuída à Mayra e Kimberly, com o prazo de 18/04/2025 a 17/06/2025

8 – “Cliente” foi atribuída à Mayra e Kimberly, com o prazo 18/04/2025 a 17/06/2025

9 – “Leis e regras contratuais” foi atribuída à Mayra e Kimberly, com o prazo de 18/04/2025 a 17/06/2025

10 – “Capítulo 3” foi atribuída à Mayra, Natháli e Kimberly, com o prazo de 04/07/2025 a 16/09/2025

11 – “Documentação” foi atribuída à Mayra, Natháli e Kimberly, com o prazo de 04/07/2025 a 16/09/2025

12 – “BPMN” foi atribuída à Mayra, Natháli e Kimberly, com o prazo de 04/07/2025 a 16/09/2025

13 – “Diagrama” foi atribuída à Mayra, Natháli e Kimberly, com o prazo de 04/07/2025 a 16/09/2025

14 – “Aplicação web” foi atribuída à Mayra, Natháli e Kimberly, com o prazo de 04/07/2025 a 16/09/2025

15 – “Capítulo 4” foi atribuída à Mayra, Natháli e Kimberly, com o prazo de 03/10/2025 a 11/11/2025

16 – “Considerações finais” foi atribuída à Mayra, Natháli e Kimberly, com o prazo de 03/10/2025 a 11/11/2025

17 – “Referências bibliográficas” foi atribuída à Mayra, Natháli e Kimberly, com o prazo de 03/10/2025 a 11/11/2025

18 – “Bibliografia” foi atribuída à Mayra, Natháli e Kimberly, com o prazo de 03/10/2025 a 11/11/2025

Tabela 2 – Cronograma *Design*

Design				
Escolha da paleta de cor	Isabela e Laura	100%	14/fev	14/fev
Design do mascote	Isabela	100%	14/fev	16/fev
Design da logo	Isabela	100%	21/fev	28/fev
Protótipo de baixa fidelidade	Giovanna e Laura	100%	28/fev	05/mar
Diagrama de fluxo de interação(DFI)	Giovanna, Isabela e Laura	100%	07/mar	19/mar
Protótipo de alta fidelidade	Isabela e Laura	100%	14/mar	19/mar
Eventuais designers caso o projeto vá para a fecetec	Isabela e Laura	0%	01/ago	15/out

FONTE: AUTORIA PRÓPRIA, 2025

Como pode ser observado na Tabela 2 acima, fica clara a distribuição de tarefas para as diversas etapas do projeto de *design*. Dentre as responsabilidades atribuídas, destacam-se a criação de uma identidade visual através do desenvolvimento de um logotipo distintivo. Além disso, há o planejamento e a implementação das múltiplas páginas que comporão o projeto, garantindo que cada uma delas seja elaborada com precisão e atenda aos objetivos estabelecidos.

Essa divisão estratégica de tarefas não apenas facilita o gerenciamento eficiente do projeto, mas também assegura que cada aspecto do *design* seja tratado com o cuidado e a criatividade necessários para alcançar um resultado impactante e alinhado com as expectativas do cliente ou do público-alvo. Vale destacar que essa organização permite uma melhor comunicação entre as equipes envolvidas, promovendo um fluxo de trabalho harmonioso e ágil. Ademais, possibilita a identificação rápida de possíveis ajustes ou melhorias ao longo do desenvolvimento, contribuindo para a qualidade final do produto entregue.

Tarefas *Design*:

1 – “Escolha da Paleta de Cor” foi designada para Isabela e Laura, se iniciou no dia 14/02/2025 e foi finalizado no mesmo dia

2 – “*Design* do Mascote” foi atribuída a Isabela, essa tarefa teve início no dia 14/02/2025 e terminou dia 16/02/2025

3 – “*Design* da Logo” atribuída a Isabela, teve início no dia 21/02/2025 e terminou dia 28/02/2025

4 – “Protótipo de Baixa Fidelidade” designada a Giovanna e Laura, essa tarefa teve início no dia 28/02/2025 e terminou dia 05/03/2025

5 – “Diagrama de Fluxo de Interação” foi atribuída a Giovanna, Isabela e Laura, essa tarefa teve início no dia 07/03/2025 e terminou dia 19/03/2025

6 – “Protótipo de Alta Fidelidade” foi atribuída a Isabela e Laura, com prazo de 14/03/2025 a 19/03/2025

7 – “Eventuais *Designers*” foi atribuída a Isabela e Laura, teve início no dia 01/08/2025 e terminou dia 15/10/2025

Tabela 3 – Cronograma Programação - Python

Programação- Python					
Aprender o básico de Python aplicado a IA	Patrick	100%	01/mar	10/mar	
Testar o Teaching Machine pré-treinado em Python	Patrick	100%	10/mar	20/mar	
Coletar e rotular um pequeno dataset usando LabelIMG	Patrick	100%	20/mar	30/mar	
Treinar Teaching Machine com poucos sinais primeiro	Patrick	100%	01/abr	10/abr	
Aprender a ajustar hiperparâmetros para melhorar a precisão.	Patrick	100%	10/abr	20/abr	
Testar como salvar e carregar o modelo para uso posterior	Patrick	0%	20/abr	30/abr	
Aprender o básico de Flask/FastAPI para criar uma API simples	Patrick	0%	10/mai	20/mai	
Fazer a API responder com o sinal identificado pelo modelo	Patrick	0%	20/mai	10/jun	
Testar o modelo com novas imagens e corrigir erros	Patrick	0%	10/jun	30/jun	
Ajustar a API para melhorar a resposta	Patrick	0%	01/jul	30/jul	
integração com o front-end	Patrick	0%	10/set	01/out	

FONTE: AUTORIA PRÓPRIA, 2025

Na tabela 3, apresenta-se as respectivas tarefas atribuídas aos responsáveis pela programação do Python no projeto. Nela há o progresso relatado até o momento, além de a data de início e término das tarefas.

Tarefas Programação - Python:

1 – “Aprender o Básico de Python aplicado a IA” foi atribuído somente ao Patrick, teve início no dia 01/03/2025 e terminou dia 10/03/2025

2 – “Testar o *Teaching Machine* Pré-Treinado em Python” atribuído ao Patrick, teve início no dia 10/03/2025 e terminou dia 20/03/2025

3 – “Coletar e Rotular um Pequeno DataSet Usando LabelIMG” foi atribuído somente ao Patrick, teve início no dia 20/03/2025 a 30/04/2025

4 – “Treinar *Teaching Machine* com Poucos Sinais Primeiro” designado ao Patrick, teve início no dia 01/04/2025 até 10/04/2025

5 – “Aprender a Ajustar Hiperparâmetros para Melhorar a Precisão” foi atribuído somente ao Patrick, teve início no dia 10/04/2025 a 20/04/2025

6 – “Testar como Salvar e Carregar o Modelo para Uso Posterior” foi atribuído somente ao Patrick, teve início no dia 20/04/2025 a 30/04/2025

7 – “Aprender o Básico de Flask/FastAPI para Criar uma API Simples” atribuído à Patrick, de 10/05/2025 a 20/05/2025

8 – “Fazer a API Responder com o Sinal Identificado pelo Modelo” foi atribuído somente ao Patrick, de 20/05/2025 a 10/06/2025

9 – “Testar o Modelo com Novas Imagens e Corrigir Erros” designado somente ao Patrick, de 10/06/2025 a 30/06/2025

10 – “Ajudar a API para Melhorar a Resposta” designado somente ao Patrick, de 01/07/2025 a 30/07/2025

11 – “Integração com o *Front-End*” designado somente ao Patrick, de 10/09/2025 a 01/10/2025

Tabela 4 – Cronograma Programação – *Machine Learning*

Programação- Machine learning					
Escolher a ferramenta para treinamento	Patrick	100%	01/mar	10/mar	
Reunir um conjunto inicial de gestos para treinar o modelo.	Patrick	100%	10/mar	20/mar	
Configurar a estrutura do projeto no computador	Patrick	100%	20/mar	30/mar	
Capturar imagens ou vídeos dos gestos que serão reconhecidos.	Patrick	100%	01/abr	10/abr	
Organizar os dados em classes no Teachable Machine.	Patrick	100%	10/abr	20/abr	
Avaliar a qualidade das imagens para garantir melhor precisão.	Patrick	0%	30/abr	10/mai	
Carregar os dados no Teachable Machine.	Patrick	0%	10/mai	20/mai	
Ajustar parâmetros do treinamento	Patrick	0%	20/mai	10/jun	
Realizar testes iniciais e refinar o conjunto de dados	Patrick	0%	10/jun	30/jun	
Exportar o modelo treinado no formato desejado	Patrick	0%	01/jul	20/jul	
Integração com o front-end	Patrick	0%	20/jul	10/ago	
Correção de erros e ajustes no sistema	Patrick	0%	10/set	01/out	

FONTE: AUTORIA PRÓPRIA, 2025

Na tabela 4, apresenta-se as respectivas tarefas atribuídas aos responsáveis pela programação do *Machine Learning* no projeto. Nela há o progresso relatado até o momento, além de a data de início e término das tarefas.

Tarefas Programação - *Machine Learning*:

1 – “Escolher a Ferramenta para Treinamento” foi atribuído somente ao Patrick, teve início no dia 01/03/2025 e terminou dia 10/03/2025

2 – “Reunir um Conjunto Inicial de Gestos para Treinar o Modelo” atribuído ao Patrick, teve início no dia 10/03/2025 e terminou dia 20/03/2025

3 – “Configurar a Estrutura do Projeto no Computador” designado ao Patrick, teve início no dia 20/03/2025 e terminou dia 30/03/2025

4 – “Capturar Imagens ou Vídeos dos Gestos que Serão Reconhecidos” foi atribuído somente ao Patrick, teve início no dia 01/04/2025 e terminou dia 10/04/2025

5 – “Organizar os Dados em Classes no *Teachable Machine*” foi atribuído ao Patrick, teve início no dia 10/04/2025 e foi finalizado dia 20/04/2025

6 – “Avaliar a Qualidade das Imagens para Garantir Melhor Precisão” designada ao Patrick, de 30/04/2025 a 10/05/2025

7 – “Carregar os Dados no *Teachable Machine*” designada ao Patrick, de 10/05/2025 até 20/05/2025

8 – “Ajustar os Parâmetros do Treinamento” designada ao Patrick, de 20/05/2025 a 10/06/2025

9 – “Realizar Testes Iniciais e Refinar o Conjunto de Dados” designada ao Patrick, teve início no dia 10/06/2025 e terminou dia 30/06/2025

10 – “Exportar o Modelo Treinado no Formato Desejado” foi atribuída ao Patrick, de 01/07/2025 a 20/07/2025

11 – “Integração com o *Front-End*” designada somente ao Patrick, de 20/07/2025 e até 10/08/2025

12 – “Correção de Erros e Ajustes no Sistema” designada ao Patrick, teve início no dia 10/09/2025 e terminou dia 01/10/2025

Tabela 5 – Cronograma Programação – *Front-End*

Programação front- end					
Tela de cadastro	Giovanna	0%	01/abr	30/abr	
Tela de login	Giovanna	0%	30/abr	10/jun	
Tela inicial	Giovanna	0%	10/jun	20/jun	
Tela visão professor	Giovanna	0%	20/jun	10/jul	
Tela visão aluno	Giovanna	0%	10/jul	30/jul	
Tela das fases	Giovanna	0%	30/jul	10/ago	
Correção de erros e ajustes no sistema	Giovanna	0%	10/ago	15/ago	

FONTE: AUTORIA PRÓPRIA, 2025

Na tabela 5, apresenta-se as respectivas tarefas atribuídas aos responsáveis pela programação do *Front-End* no projeto. Nela há o progresso relatado até o momento, além de a data de início e término das tarefas.

Tarefas Programação - *Front-End*:

1 – “Tela de Cadastro” designada à Giovanna, de 01/04/2025 até 30/04/2025

2 – “Tela de *Login*” foi atribuída à Giovanna, de 30/04/2025 a 10/06/2025

3 – “Tela Inicial” designada à Giovanna, com o prazo de 10/06/2025 até o dia 20/06/2025

4 – “Tela Visão Professor” foi atribuída à Giovanna, de 20/06/2025 até 10/07/2025

5 – “Tela Visão Aluno” designada à Giovanna, de 10/07/2025 a 30/07/2025

6 – “Tela das Fases” designada à Giovanna, com o prazo de 30/07/2025 até 10/08/2025

7 – “Correção de Erros e Ajustes no Sistemas” designada à Giovanna, com o prazo de 10/08/2025 até o dia 15/08/2025

Tabela 6- Cronograma Programação – LangFlow

Programação- Lang flow					
Instalar softwares necessários	Patrick	100%	28/fev	01/abr	
Programação da AI	Patrick	100%	01/abr	01/ago	
Fazer ligação com o banco de dados	Patrick	0%	01/ago	15/ago	
Conectar com chave API	Patrick	0%	15/ago	01/set	
Integração com o front-end	Patrick	0%	01/set	10/set	
Correção de erros e ajustes no sistema	Patrick	0%	15/set	01/out	

FONTE: AUTORIA PRÓPRIA, 2025

Na tabela 6, apresenta-se as respectivas tarefas atribuídas aos responsáveis pela programação do LangFlow no projeto. Nela há o progresso relatado até o momento, além de a data de início e término das tarefas.

Tarefas Programação – LangFlow:

1 – “Instalar *Softwares* Necessários” designada ao Patrick, teve início no dia 28/02/2025 e terminou dia 01/04/2025

2 – “Programação de AI” foi atribuída ao Patrick, com o prazo de 01/04/2025 até 01/08/2025

3 – “Fazer Ligação com o Banco de Dados” designada ao Patrick, de 01/08/2025 a 15/08/2025

4 – “Conectar com a Chave API” atribuída ao Patrick, de 15/08/2025 até 01/09/2025

5 – “Integração com o *Front-End*” designada ao Patrick, de 01/09/2025 a 10/09/2025

6 – “Correção de Erros e Ajustes no Sistema” designada ao Patrick, com o prazo de 15/09/2025 até 01/10/2025

Tabela 7 - Cronograma Programação – Banco de Dados

Banco de dados					
Modelo conceitual	Miguel	0%	30/abr	10/mai	
Modelo lógico	Miguel	0%	10/mai	20/mai	
Modelo físico	Miguel	0%	20/mai	30/mai	
Criar conta no serv00	Miguel	0%	30/mai	10/jun	
Inserção de dados no banco	Miguel	0%	10/jun	20/jun	
Integração com o front-end	Miguel	0%	20/jun	30/jun	
Correção de erros e ajustes no sistema	Miguel	0%	30/jun	10/jul	

FONTE: AUTORIA PRÓPRIA, 2025

Na Tabela 7, apresenta-se as respectivas tarefas atribuídas aos responsáveis pela programação do Banco de Dados no projeto. Nela, há o progresso relatado até o momento, além das datas de início e término das tarefas, permitindo um acompanhamento claro e transparente do andamento dos trabalhos. Esse controle detalhado é fundamental para garantir que os prazos sejam cumpridos e que eventuais desafios sejam identificados e solucionados rapidamente.

Além disso, a tabela facilita a coordenação entre os membros da equipe, assegurando que as atividades estejam alinhadas com os objetivos gerais do projeto. Com essa organização, é possível otimizar os recursos disponíveis e assegurar a qualidade no desenvolvimento do Banco de Dados.

Tarefas Programação – Banco de Dados:

- 1 – “Modelo Conceitual” designada ao Miguel, de 30/04/2025 até 10/05/2025
- 2 – “Modelo Lógico” foi atribuída ao Miguel, com o prazo de 10/05/2025 até 20/05/2025
- 3 – “Modelo Físico” foi designada ao Miguel, de 20/05/2025 a 30/05/2025
- 4 – “Criar Conta no Serv00” atribuída ao Miguel, de 30/05/2025 até 10/06/2025
- 5 – “Inserção de Dados no Banco” designada ao Miguel, de 10/06/2025 a 20/06/2025
- 6 – “Integração com o *Front-End*” designada ao Miguel, com o prazo de 20/06/2025 até 30/06/2025
- 7 – “Correção de Erros e Ajustes no Sistema” designada ao Miguel, com o prazo de 30/06/2025 até 10/07/2025

Tabela 8 - Cronograma Geral

Tcc Scan Hands					
Lider: Isabela					
Integrantes: Giovanna, Isabela, Kimberly, Laura, Mayra, Miguel Maia e Patrick					
Início do projeto: 07/02/2025					
Tarefas	Responsável por ela	Progresso	Início	Fim	
Grupo todo					
Escolha do tema	todos	100%	07/fev	07/fev	
Escolha do nome	todos	100%	07/fev	28/fev	
Viabilidade econômica	Isabela	100%	21/fev	28/fev	
Viabilidade técnica	Miguel	100%	21/fev	28/fev	
Viabilidade operacional	Giovanna	100%	21/fev	28/fev	
Viabilidade do cronograma	Isabela	100%	21/fev	28/fev	
Pesquisar como a área de atuação tem sofrido	Kimberly	100%	14/fev	21/fev	
Pesquisar sobre libras no brasil	Mayra	100%	14/fev	21/fev	
Pesquisar sobre machine learning	Giovanna	100%	14/fev	21/fev	
Pesquisa sobre a melhor linguagem de programação	Miguel	100%	14/fev	21/fev	
Pesquisa sobre ias e implementação destas no projeto	Patrick	100%	14/fev	21/fev	
Pesquisa sobre paletas de cores	Laura	100%	14/fev	21/fev	
Compilação das pesquisas	Isabela	100%	14/fev	21/fev	
Questionários	todos	100%	28/fev	12/mar	
Entrevistas	todos	100%	28/fev	12/mar	
Apuração do questionário e entrevista	Isabela e Giovanna	100%	12/mar	26/mar	
Teste de qualidade de software	todos	0%	01/out	07/out	
Apresentação final	todos	0%	01/out	21/nov	
Revisão bibliográfica	todos	0%	01/out	10/nov	

FONTE: AUTORIA PRÓPRIA, 2025

Na tabela 8, estão atribuídas as tarefas gerais do projeto, sendo elas das demais divisões de trabalho. Nesta parte, foram colocadas as tarefas que não se encaixariam nos outros grupos. Na tabela, há uma descrição da tarefa, a quem foi atribuída, o progresso relatado na tarefa, além de a data de início e término da tarefa.

Tarefas Gerais:

1 – “Escolha do Tema” designada a todos os membros do grupo, teve seu início no dia 07/02/2025 e foi finalizado no mesmo dia

2 – “Escolha do Nome” foi atribuída a todos os membros do grupo, com o prazo de 07/02/2025 até 28/02/2025

3 – “Viabilidade Econômica” foi designada somente à Isabela, de 21/02/2025 a 28/02/2025

4 – “Viabilidade Técnica” atribuída ao Miguel, de 21/02/2025 a 28/02/2025

5 – “Viabilidade Operacional” designada à Giovanna, de 21/02/2025 a 28/02/2025

6 – “Viabilidade do Cronograma” designada à Isabela, com o prazo de 21/02/2025 a 28/02/2025

7 – “Pesquisar como a Área de Atuação tem Sofrido” designada à Kimberly, com o prazo de 14/02/2025 até 21/02/2025

8 – “Pesquisar sobre as Libras no Brasil” designada à Mayra, com o prazo de 14/02/2025 até 21/02/2025

9 – “Pesquisar sobre *Machine Learning*” designada à Giovanna, com o prazo de 14/02/2025 até 21/02/2025

10 – “Pesquisar sobre a Melhor Linguagem de Programação” designada ao Miguel, de 14/02/2025 até 21/02/2025

11 – “Pesquisar sobre IAs e Implementação Destas no Projeto” designada ao Patrick, com o prazo de 14/02/2025 até 21/02/2025

12 – “Pesquisar sobre Paletas de Cores” designada à Laura, com o prazo de 14/02/2025 até 21/02/2025

13 – “Compilação das Pesquisas” designada à Isabela, com o prazo de 14/02/2025 até 21/02/2025

14 – “Questionários” designada a todos os membros do grupo, com o prazo de 28/02/2025 até 12/03/2025

15 – “Entrevistas” designada a todos os membros do grupo, com o prazo de 28/02/2025 até 12/03/2025

16 – “Apuração do Questionário e Entrevistas” designada à Giovanna e Isabela, com o prazo de 12/03/2025 até 26/03/2025

17 – “Teste de Qualidade de *Software*” designada a todos os membros do grupo, com o prazo de 01/10/2025 até 07/10/2025

18 – “Apresentação Final” designada a todos os membros do grupo, com o prazo de 01/10/2025 até 21/11/2025

19 – “Revisão Bibliográfica” designada a todos os membros do grupo, com o prazo de 01/10/2025 até 10/11/2025

2.1.4 – Viabilidade Econômica

Viabilidade econômica é um termo utilizado para analisar a capacidade de um projeto, empreendimento ou investimento em gerar lucros e retornos financeiros satisfatórios. É uma etapa fundamental em qualquer decisão de negócio, pois permite avaliar se

uma ideia é viável do ponto de vista econômico antes de ser colocada em prática. Neste processo, são considerados diversos fatores como custos, receitas, investimentos iniciais, prazos de retorno e taxas de retorno esperadas. A viabilidade econômica é feita através de estudos e análises detalhadas, que ajudam a identificar riscos, oportunidades e potenciais obstáculos que podem afetar o sucesso do empreendimento. (MAESTROVIRTUALE,2024)

Tabela 9 - Economia

CUSTOS	VALOR MENSAL	VALOR ANUAL		Profissão	Cargos	Salário	Total
	VALOR MENSAL	VALOR ANUAL		Designer	2	R\$ 3.000,00	R\$ 6.000,00
CUSTOS FIXOS				Programador Dev	2	R\$ 3.500,00	R\$ 7.000,00
Aluguel	R\$ 1.000,00	R\$ 12.000,00		Programador Bd	1	R\$ 3.500,00	R\$ 3.500,00
Internet	R\$ 100,00	R\$ 1.200,00		Administração	2	R\$ 2.250,00	R\$ 4.500,00
Salário	R\$ 22.500,00	R\$ 270.000,00		Limpeza	1	R\$ 1.500,00	R\$ 1.500,00
Dominio	R\$ 165,00	R\$ 1.980,00					
Marketing	R\$ 1.500,00	R\$ 18.000,00		Escolas	Unidades	Aluguel mensal por unidade	Total
Telefone	R\$ 20,00	R\$ 240,00		Municipais	1434	R\$ 20,00	R\$ 43.020,00
CUSTOS VARIÁVEIS				Estaduais	5300	R\$ 20,00	R\$ 159.000,00
Água	R\$ 65,28	R\$ 783,36					
Energia	R\$ 324,50	R\$ 3.894,00		Lucros	Valor mensal	Valor anual	
Mercado	R\$ 100,00	R\$ 1.200,00		Receita	R\$ 134.680,00	R\$ 1.616.160,00	
CUSTOS ÚNICOS				Custo Total	R\$ 55.181,78	R\$ 338.704,36	
Móveis	R\$	8.342,00		Lucro líquido	R\$ 79.498,22	R\$ 1.277.455,64	
Eletrônicos	R\$	21.000,00					
Câmera	R\$	65,00					
TOTAL	R\$ 55.181,78	R\$ 338.704,36					

FONTE: AUTORIA PRÓPRIA, 2025

A tabela 9 mostra um detalhado orçamento dos custos fixos, custos variáveis, custos únicos, salário dos funcionários do projeto e valor cobrado mensalmente das escolas para que a aplicação gere lucro. Os custos fixos incluem orçamentos que não sofrem mudanças constantes como aluguel, salário e *internet*, em contrapartida os custos variáveis contam com despesas que podem variar durante os meses ou anos, como água e mercado. Já os custos únicos são coisas que serão compradas uma única vez e que não gerarão gastos futuros como móveis e eletrônicos. A planilha tem como objetivo mostrar uma visão geral das finanças e ajudar na avaliação da viabilidade econômica da aplicação.

O objetivo do SCAN HANDS é ensinar Libras de maneira simplificada, portanto o financiamento do projeto será através da compra dele por escolas estaduais e municipais inicialmente do estado de São Paulo. Como identificado na tabela, os custos anuais de projeto são de R\$ 338.704,36 e o valor arrecadado se todas as escolas do estado de SP aderirem a aplicação é de R\$ 1.616.160,00, dessa forma o lucro líquido do projeto é de R\$ 1.277.455,64 anuais, o que representa aproximadamente 79% da receita total. Sendo assim, é concluído que a realização do projeto e implementação dele é viável financeiramente.

2.2 – TÉCNICA DE LEVANTAMENTO DE REQUISITOS

As técnicas de levantamento são fundamentais para identificar e entender as necessidades, requisitos e expectativas em qualquer tipo de projeto. Elas são empregadas para coletar informações de forma estruturada, com o objetivo de garantir que soluções desenvolvidas estejam alinhadas com os interesses dos envolvidos. (DEVMEDIA,2025)

Entre as principais técnicas, destaca-se a entrevista, que permite obter dados ricos por meio de conversas diretas com *stakeholders*. Questionários e pesquisas são eficazes quando se deseja alcançar um público maior, oferecendo uma visão ampla através de dados quantitativos e qualitativos. (DEVMEDIA,2025)

A observação direta, por sua vez, permite identificar processos e comportamentos reais no ambiente analisado, muitas vezes revelando detalhes que os próprios usuários não mencionam. *Workshops* e dinâmicas de grupo promovem o envolvimento coletivo, facilitando o alinhamento e a validação das necessidades em conjunto. (DEVMEDIA,2025)

Já a análise de documentos é ideal para projetos de melhoria ou auditoria, pois permite extrair informações de registros preexistentes. Por fim, a prototipagem cria modelos iniciais de soluções, possibilitando testar conceitos e ajustar o projeto com base no *feedback* dos usuários. (DEVMEDIA,2025)

A escolha das técnicas deve considerar o contexto, os recursos disponíveis e os objetivos do levantamento, sendo comum o uso combinado de diferentes abordagens para obter resultados mais completos e assertivos. (DEVMEDIA,2025)

Para isso existem várias técnicas diferentes, porém no desenvolvimento do levantamento foram utilizadas apenas entrevistas que ocorreram pessoalmente e em um formulário.

2.2.1 – Entrevista

As entrevistas presenciais foram uma etapa essencial para aprimorar o desenvolvimento do projeto, permitindo uma aproximação direta com a realidade das instituições envolvidas. Por meio de conversas com alunos, professores, e outras pessoas foi possível identificar necessidades específicas, sugestões de melhoria e pontos críticos que muitas vezes passariam despercebidos em abordagens mais

distantes ou padronizadas. Essa escuta ativa contribuiu para o fortalecimento do vínculo entre projeto e comunidade escolar, além de gerar dados qualitativos valiosos para o processo de refinamento técnico e pedagógico da solução proposta. Com base nessas interações, ajustes mais eficazes foram realizados, promovendo uma maior aderência à prática educacional e consolidando um processo participativo e colaborativo. No total 6 pessoas foram entrevistadas.

Perguntas:

1-Como você acha que a inclusão da Língua Brasileira de Sinais (Libras) na sociedade poderia impactar a comunicação entre pessoas surdas e ouvintes?

Justificativa: Essa pergunta busca entender como a inclusão da Libras pode melhorar a comunicação entre surdos e ouvintes, reforçando a importância do ensino dessa língua como ferramenta de inclusão social e acessibilidade.

2-Quais desafios você enxerga na difusão do ensino de Libras em escolas e ambientes de trabalho?

Justificativa: A pergunta busca entender os obstáculos para o ensino de Libras, reforçando a importância da inclusão nos espaços educacionais e profissionais.

3-Você já teve alguma experiência em que o conhecimento de Libras teria sido útil? Como foi essa situação?

Justificativa: A pergunta busca identificar o que desperta interesse pelo aprendizado de Libras, ajudando a pensar em formas mais eficazes de promover seu ensino.

4-O que te motivaria (ou pode motivar outra pessoa) a aprender Libras?

Justificativa :A pergunta permite identificar situações reais em que a Libras faria diferença, destacando sua importância prática no dia a dia e reforçando a relevância do seu ensino.

5- Na sua opinião, quais são as principais barreiras para que mais pessoas aprendam Libras?

Justificativa :A pergunta busca entender os fatores que dificultam o aprendizado de Libras, ajudando a refletir sobre como tornar seu ensino mais acessível e eficaz.

6-Na sua opinião, quais são as principais barreiras para que mais pessoas aprendam Libras?

Justificativa: Entender se as barreiras são individualizadas e se nossa aplicação poderia acabar com elas.

7- Você usaria um aplicativo que ensina libras que utiliza a câmera para avaliar movimentos? Como você acha que isso ajudaria no aprendizado?

Justificativa: Entender se o desenvolvimento da aplicação seria válido e ajudaria a comunidade e como nosso principal diferencial é avaliado.

2.2.2 – Questionário

O uso de questionários é recomendado em contextos com diversos grupos de usuários distribuídos geograficamente, onde seria inviável realizar entrevistas presenciais com todos. Nesses casos, pesquisas direcionadas com usuários selecionados são uma alternativa eficaz. Há diferentes formatos de questionários, como múltipla escolha, listas de verificação e questões abertas. Para garantir eficiência na resposta, o questionário deve ser objetivo e de rápida conclusão. Durante sua preparação, é essencial definir previamente quais informações se pretende obter. O analista deve elaborar perguntas claras e concisas, organizar os tópicos em blocos com títulos específicos e reservar espaço adequado para respostas descritivas. (DEV MEDIA,2025)

Foram formuladas 5 perguntas em formato de formulário via Google Docs e *apenas* uma dissertativa, as quais foram disponibilizadas para os alunos da escola ETEC Euro Albino de Souza, como forma de questionário. No total houve 17 respostas.

Perguntas:

1-Qual a sua idade?

Justificativa: Entender a idade do nosso público-alvo para direcionar o desenvolvimento do nosso sistema.

2-Você já teve contato com a Língua Brasileira de Sinais (Libras)?

Justificativa: Entender se o nosso público-alvo já teve algum contato com a Linguagem de sinal.

Opção 1: Sim

Opção 2: Não

3- Você acredita que Libras deveria ser ensinada nas escolas de forma obrigatória?

Justificativa: Entender se caso nosso projeto fosse implantado nas escolas ele seria usado e se iria ser importante implanta-lo nas escolas

Opção 1: Sim

Opção 2: Não

4- Você estaria disposto (a) a aprender Libras se houvesse um aplicativo acessível para você?

Justificativa: Saber se nossa aplicação seria utilizada pelos usuários

Opção 1: Sim

Opção 2: Não

5- Você já enfrentou dificuldades ao tentar se comunicar com uma pessoa surda?

Justificativa: Entender quantas pessoas já tiveram dificuldade e analisar o impacto da nossa aplicação nessa situação

Opção 1: Sim

Opção 2: Não

6- Você conhece alguma iniciativa ou projeto que promove o ensino de Libras na sua cidade ou comunidade?

Justificativa: Verificar a concorrência em Mogi-Guaçu é essencial para identificar oportunidades e ajustar nosso aplicativo às necessidades locais. Isso permite planejar estratégias eficazes para se destacar no mercado.

Opção 1: Sim

Opção 2: Não

2.2.3 - Análise de texto

A análise textual desempenha um papel essencial na interpretação de dados obtidos em pesquisas e estudos acadêmicos. Por meio da observação detalhada das respostas coletadas, é possível identificar padrões, compreender necessidades específicas e aperfeiçoar abordagens metodológicas. Esse processo permite uma visão ampla sobre os elementos presentes nos textos analisados, garantindo que a estrutura e a fundamentação estejam alinhadas com os objetivos propostos. Além disso, contribui para embasar decisões fundamentadas em evidências concretas, possibilitando ajustes estratégicos que aprimoram o desenvolvimento do trabalho. A análise textual também fortalece a compreensão do material estudado, permitindo uma interpretação mais precisa e aprofundada, essencial para garantir a coerência e qualidade dos resultados. (AWS,2025)

2.2.3.1 – Análise das respostas da entrevista

Entrevistados:

1- Cláudio José dos Santos - (Aprendiz de Libras)

“Como você acha que a inclusão de Libras ajuda a comunicação entre pessoas surdas e ouvintes? “

R: “Acho que Libras é, sim, uma das formas mais práticas de garantir a inclusão das pessoas surdas na sociedade. Como são uma minoria, é fundamental que existam leis e ações que facilitem essa inclusão”.

“Quais desafios você enxerga no ensino de Libras em escolas e ambientes de trabalho?”.

R: “Libras deveria estar presente desde cedo, como qualquer outra língua. É essencial para dar acesso igual a todos, principalmente no trabalho e na escola”.

“Você já teve alguma experiência em que o conhecimento de Libras teria sido útil? Como foi essa situação?”

R: “Sim, no meu trabalho tenho contato direto com pessoas surdas. Para poder me comunicar com elas, comecei a aprender Libras por conta própria”.

“O que te motivaria (ou pode motivar outra pessoa) a aprender Libras?”.

R: “Eu acredito que a maior motivação é promover a inclusão. Os surdos têm o direito de estar em todos os espaços, e saber Libras ajuda a garantir isso”.

“Na sua opinião, quais são as principais barreiras para que mais pessoas aprendam Libras?”

R: “As leis existem, mas não são colocadas em prática como deveriam. Sem incentivo e fiscalização, os surdos continuam sendo ignorados”.

“Você usaria um aplicativo que ensina Libras que utiliza a câmera para avaliar movimentos? Como você acha que isso ajudaria no aprendizado?”.

R: “Com certeza! Um *app* assim facilitaria demais, principalmente para quem é ouvinte e não tem contato direto com surdos, mas quer aprender”.

2- Juliana Torres Barros - (Aprendiz de Libras)

“Como você acha que a inclusão de Libras ajuda a comunicação entre pessoas surdas e ouvintes?”

R: “Ajudaria bastante, porque através da Libras você consegue entender o que a pessoa surda quer te dizer. É uma ponte de comunicação que aproxima.”.

“Quais desafios você enxerga no ensino de Libras em escolas e ambientes de trabalho?”.

R: “Um dos grandes desafios é a falta de professores qualificados. Nem toda escola ou local de trabalho tem alguém preparado para ensinar Libras.”

“Você já teve alguma experiência em que o conhecimento de Libras teria sido útil? Como foi essa situação?”.

R: “Sim, já aconteceu de uma pessoa surda passar pelo caixa no meu trabalho e a atendente conseguiu realizar toda a compra usando Libras. Foi incrível ver como isso fez a diferença”.

“O que te motivaria (ou pode motivar outra pessoa) a aprender Libras?”.

R: “No meu caso, é o trabalho. Quero estar preparado para quando for necessário me comunicar com alguém surdo, sem depender de outras pessoas”.

“Na sua opinião, quais são as principais barreiras para que mais pessoas aprendam Libras?”

R: “A falta de professores capacitados dificulta muito. Quem quer aprender não encontra apoio suficiente”.

“Você usaria um aplicativo que ensina Libras que utiliza a câmera para avaliar movimentos? Como você acha que isso ajudaria no aprendizado?”

R: “Sim, seria ótimo poder estudar Libras nas horas vagas com um aplicativo prático”.

3- Eduardo Freitas Moura - (Aprendiz de Libras)

“Como você acha que a inclusão de Libras ajuda a comunicação entre pessoas surdas e ouvintes?”

R: “Libras facilita muitas coisas. Além de ajudar na comunicação em geral, também pode ser útil em situações do dia a dia, como no comércio, por exemplo”.

“Quais desafios você enxerga no ensino de Libras em escolas e ambientes de trabalho?”

R: “A Libras ainda é deixada de lado. Falta incentivo e pessoas capacitadas. Nem sempre há alguém que saiba ou esteja disponível para ensinar”.

“Você já teve alguma experiência em que o conhecimento de Libras teria sido útil? Como foi essa situação?”

R: “Uma vez, no Burger King, chegou um cliente surdo e ninguém sabia Libras. A gente só conseguiu atender ele usando o cardápio com imagens. A comunicação ficou super limitada.”

“O que te motivaria (ou pode motivar outra pessoa) a aprender Libras?”

R: “A motivação pode vir de querer atuar em áreas específicas onde a Libras é essencial, como educação ou atendimento ao público”.

“Na sua opinião, quais são as principais barreiras para que mais pessoas aprendam Libras?”

R: “Falta gente falando sobre Libras. É uma língua que ainda não tem a visibilidade que realmente merece”.

“Você usaria um aplicativo que ensina Libras que utiliza a câmera para avaliar movimentos? Como você acha que isso ajudaria no aprendizado? ”

R: “Usaria sim! E o fato de a câmera avaliar os movimentos ajuda bastante a corrigir e aprender da maneira certa, tudo de casa, sem complicação”.

4- Renan Nogueira dos Santos - (Aprendiz de Libras)

“Como você acha que a inclusão de Libras ajuda a comunicação entre pessoas surdas e ouvintes?”

R: “Acho essencial. Imagina como seria confuso se a gente não tivesse nenhum recurso para se comunicar com quem precisa disso. Libras evita esse tipo de isolamento”.

“Quais desafios você enxerga no ensino de Libras em escolas e ambientes de trabalho?”

R: “O estudo de Libras ainda é muito restrito. Pouquíssimas escolas oferecem esse conteúdo e isso limita muito quem quer aprender”.

“Você já teve alguma experiência em que o conhecimento de Libras teria sido útil? Como foi essa situação?”

R: “Minha mãe já quis aprender Libras para ajudar uma aluna dela que era surda. Infelizmente, ela não conseguiu porque o acesso às aulas era muito difícil.”

“O que te motivaria (ou pode motivar outra pessoa) a aprender Libras? ”

R: “Assim como minha mãe teve interesse em aprender para ajudar uma aluna, eu também teria essa motivação. É um gesto de empatia e inclusão”.

“Na sua opinião, quais são as principais barreiras para que mais pessoas aprendam Libras?”

R: “Ainda é muito difícil ter acesso ao ensino de Libras. Não é algo que se aprende na escola regular e quase não tem cursos acessíveis”.

“Você usaria um aplicativo que ensina Libras que utiliza a câmera para avaliar movimentos? Como você acha que isso ajudaria no aprendizado?”

R: “ Sim! Isso melhoraria muito a comunicação dos surdos e permitiria que ouvintes se aproximassem mais. Todos saem ganhando”.

5- Eduarda Ribeiro da Silva - (Surda)

“Como você acha que a inclusão de Libras ajuda a comunicação entre pessoas surdas e ouvintes?”

R: “Ajuda muito mesmo. Você consegue conversar e entender melhor as pessoas, tudo se torna mais acessível e natural”.

“Quais desafios você enxerga no ensino de Libras em escolas e ambientes de trabalho?”

R: “Falta visibilidade. Libras precisa ser mais divulgada e valorizada, com mais profissionais capacitados para ensinar, especialmente no ambiente de trabalho”

“Você já teve alguma experiência em que o conhecimento de Libras teria sido útil? Como foi essa situação?”

R: “Sim, no meu trabalho (na Mahle) preciso falar em Libras todos os dias, e sempre tem colegas que ajudam. Libras virou uma necessidade real para mim”.

“O que te motivaria (ou pode motivar outra pessoa) a aprender Libras?”

R: “A maior motivação é fazer com que as pessoas surdas se sintam incluídas, seja no trabalho, na escola ou em qualquer outro lugar. Todos nós temos direitos iguais”.

“Na sua opinião, quais são as principais barreiras para que mais pessoas aprendam Libras?”

R: “Muita gente não tem acesso, seja por questões de distância, falta de informação ou mesmo custo”.

“Você usaria um aplicativo que ensina Libras que utiliza a câmera para avaliar movimentos? Como você acha que isso ajudaria no aprendizado?”

R: “Claro que sim. Um aplicativo desses tornaria o ensino muito mais acessível e rápido, além de ajudar bastante na hora de se comunicar com surdos”.

6- Adenilson Garcia da Cruz – (Professor do ensino fundamental, médio, técnico e superior)

“Como você acha que a inclusão de Libras ajuda a comunicação entre pessoas surdas e ouvintes?”

R: “Saber Libras é uma forma de respeito. A pessoa surda já faz parte da sociedade, então é a gente que precisa estar preparado para se comunicar. É como viajar para um país com outra língua, se você não sabe se comunicar também fica "surdo" naquele contexto.

“Quais desafios você enxerga no ensino de Libras em escolas e ambientes de trabalho?”

R: “Para mim, o único obstáculo é a falta de vontade política. Se há professores de inglês e de robótica nas escolas, por que não de Libras? É só colocar no currículo e contratar quem sabe”.

“Você já teve alguma experiência em que o conhecimento de Libras teria sido útil? Como foi essa situação?”

R: “Já tive aluno surdo na sala de aula e foi bem desafiador. Tive que adaptar a aula para algo mais visual, o que exige muito mais preparo e criatividade”.

“O que te motivaria (ou pode motivar outra pessoa) a aprender Libras? ”

R: “Melhoraria muito minha experiência em sala de aula. Seria uma ferramenta a mais para acolher e ensinar com mais eficiência”.

“Na sua opinião, quais são as principais barreiras para que mais pessoas aprendam Libras?”

R: “Existem várias barreiras. Primeiro, poucas escolas ensinam; segundo, quando existem, ficam longe; terceiro, os horários nem sempre ajudam; quarto, nem todo mundo tem interesse; e quinto, mesmo sendo algo importante, ainda é preciso pagar para aprender. Já que o governo fala tanto de inclusão, ele poderia oferecer o curso de graça”.

“Você usaria um aplicativo que ensina Libras que utiliza a câmera para avaliar movimentos? Como você acha que isso ajudaria no aprendizado?”

R: "Sem dúvidas. Isso seria muito útil tanto para mim quanto para outros professores. Um *app* assim deixaria o ensino mais simples e direto, o que é ótimo, principalmente porque muitos alunos não gostam de métodos mais tradicionais".

As entrevistas realizadas com aprendizes de Libras, uma pessoa surda e um professor da educação básica e superior evidenciam a relevância da Libras como ferramenta fundamental para a inclusão e a comunicação entre pessoas surdas e ouvintes. Há um consenso entre os participantes de que a Libras atua como uma ponte de integração social, escolar e profissional, garantindo direitos básicos de comunicação e acesso à informação.

Um dos pontos mais recorrentes nas falas é a carência de ensino estruturado de Libras em escolas e ambientes de trabalho. Muitos destacam a ausência de professores qualificados, a escassez de políticas públicas efetivas, e a falta de incentivo institucional para que Libras seja tratada como uma disciplina essencial, presente desde os primeiros anos da educação básica. Também foi apontado que, mesmo com leis que garantem o direito à inclusão, a falta de fiscalização e investimento concreto prejudica a efetividade dessas normas. A motivação para o aprendizado da Libras aparece, em sua maioria, associada à empatia, ao desejo de inclusão e à necessidade prática no ambiente profissional, especialmente no comércio, no setor educacional e em empresas com diversidade de funcionários. Relatos pessoais mostraram situações nas quais a comunicação com pessoas surdas foi dificultada por não haver alguém fluente em Libras, demonstrando a urgência em ampliar o número de pessoas capacitadas nessa linguagem.

Além disso, foram citadas barreiras significativas como a baixa visibilidade da Libras na sociedade, a distância física dos cursos, os custos envolvidos no aprendizado e a falta de interesse generalizado. A ausência de políticas públicas que garantam o acesso gratuito e acessível ao ensino de Libras também foi um ponto crítico levantado por alguns entrevistados.

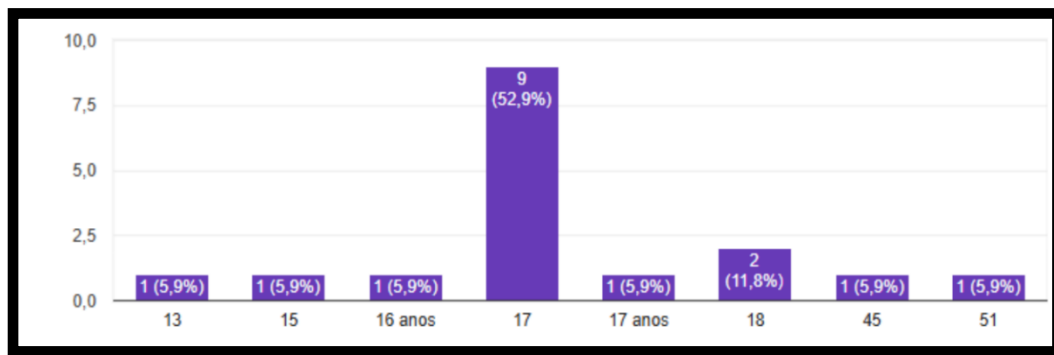
Quanto à proposta de um aplicativo com reconhecimento de movimentos via câmera, a recepção foi amplamente positiva. Os participantes consideraram a ideia prática e eficiente, destacando que o uso de tecnologia pode facilitar o processo de aprendizagem, principalmente para iniciantes que não têm contato direto com pessoas surdas. O uso de recursos visuais, correção de sinais e a possibilidade de aprender

em horários flexíveis foram apontados como diferenciais que podem ajudar a suprir as lacunas deixadas pelo ensino tradicional.

Por fim, a análise das entrevistas demonstra que o desenvolvimento de soluções acessíveis e tecnológicas voltadas ao ensino de Libras, como aplicativos interativos, pode contribuir significativamente para a disseminação da língua, promover maior inclusão social e reduzir o isolamento enfrentado por pessoas surdas em diversos contextos da sociedade.

2.2.3.2 – Análise das respostas do questionário

Figura 12- Gráfico de respostas para “Qual a sua idade?”

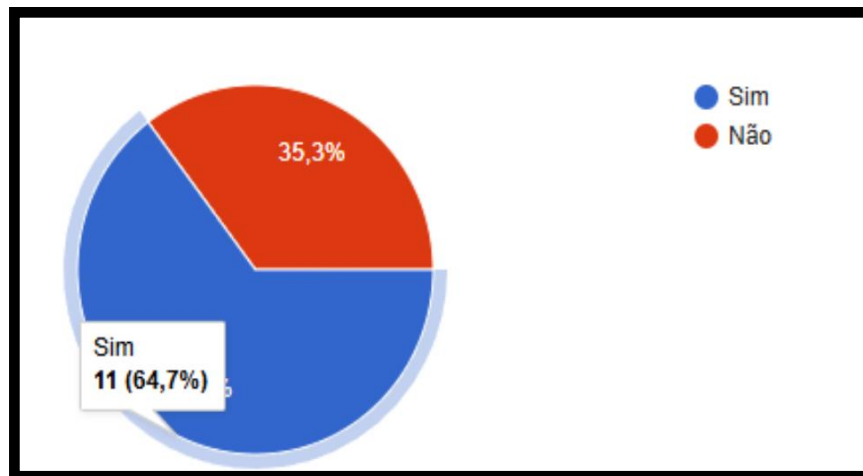


FONTE: AUTORIA PRÓPRIA, 2025

Resposta – A Figura 12 apresenta a distribuição etária dos participantes da pesquisa. A maior parte dos respondentes tem 17 anos, representando 52,9% do total. Outras idades também foram registradas, com menor frequência: 13, 15, 16, 17 (repetido), 18, 45 e 51 anos, cada uma correspondendo a 5,9%, exceto 18 anos, com 11,8%.

Esses dados indicam uma predominância de adolescentes, especialmente estudantes do ensino médio, mas também revelam a participação de adultos, o que contribui para uma amostra diversificada. Essa diversidade etária enriquece a pesquisa, permitindo uma análise mais ampla e abrangente das diferentes faixas etárias envolvidas. Além disso, a presença de diferentes grupos etários pode influenciar nas preferências e necessidades, o que deve ser considerado no desenvolvimento do projeto. A variação na idade dos participantes também reforça a importância de estratégias adaptáveis para atingir públicos distintos. Dessa forma, o projeto pode ser aprimorado para melhor atender às expectativas de cada grupo demográfico.

Figura 13 - Gráfico de respostas para “Você já teve contato com a Língua Brasileira de Sinais (Libras)?”



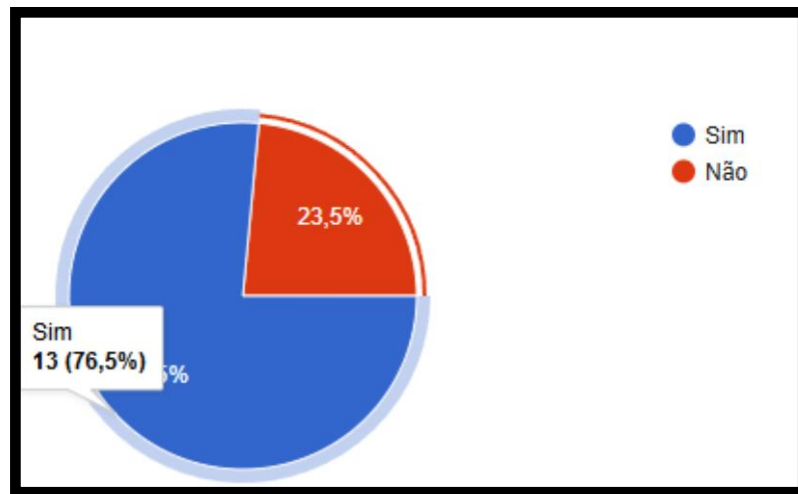
FONTE: AUTORIA PRÓPRIA, 2025

Resposta - Conforme mostra a Figura 13, a maioria dos participantes (64,7%) afirmou já ter tido contato com a Língua Brasileira de Sinais (Libras), o que demonstra um nível significativo de familiaridade com essa forma de comunicação. Por outro lado, 35,3% responderam que nunca tiveram esse contato, evidenciando que ainda há espaço para ampliar o conhecimento e a inclusão por meio da disseminação da Libras no ambiente escolar.

Esse dado reforça a importância de ações educativas e programas que promovam o aprendizado e a valorização da Libras, contribuindo para a construção de um ambiente mais inclusivo e acessível para pessoas surdas. Além disso, o aumento da familiaridade com a Libras entre estudantes pode facilitar a comunicação e a interação social, promovendo maior empatia e respeito às diferenças. Dessa forma, investir na capacitação em Libras pode impactar positivamente tanto o ambiente escolar quanto a comunidade em geral, fortalecendo a cultura da inclusão.

É fundamental que as escolas implementem estratégias contínuas para incentivar o contato e o aprendizado da Libras. Ademais, a presença de profissionais capacitados em Libras no contexto escolar amplia as possibilidades de inclusão e contribui para o desenvolvimento de práticas pedagógicas mais equitativas e representativas.

Figura 14 – Gráfico de respostas para “Você acredita que Libras deveria ser ensinada nas escolas de forma obrigatória?”



FONTE: AUTORIA PRÓPRIA, 2025

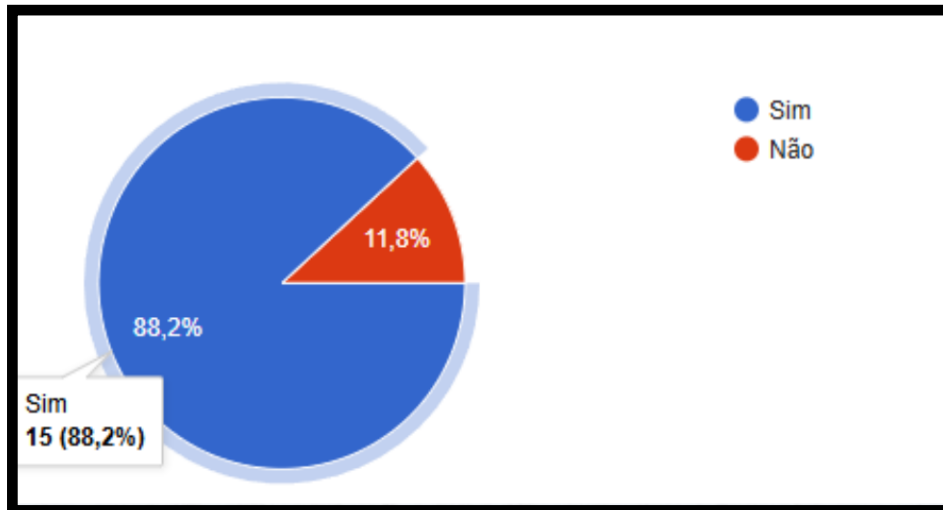
Resposta – A Figura 14 revela que a maioria dos participantes (76,5%) acredita que a Língua Brasileira de Sinais (Libras) deveria ser ensinada obrigatoriamente nas escolas. Esse dado reflete uma forte consciência sobre a importância da inclusão e da acessibilidade na educação. Apenas 23,5% responderam “Não”, o que indica que a resistência à proposta é relativamente baixa, reforçando a relevância de políticas educacionais que integrem Libras ao currículo escolar.

A ampla aceitação dessa ideia demonstra um crescente reconhecimento da Libras como um instrumento fundamental para garantir a equidade no processo de ensino-aprendizagem, especialmente para estudantes surdos. Ao incluir a Libras como disciplina obrigatória, promove-se não apenas o domínio de uma nova linguagem, mas também o desenvolvimento de valores como empatia, respeito e convivência com a diversidade.

Além disso, o ensino da Libras pode favorecer a formação de cidadãos mais preparados para lidar com diferentes realidades sociais e comunicativas, contribuindo para uma sociedade mais justa e inclusiva. Dessa forma, investir na presença da Libras nas escolas é um passo essencial para a construção de uma educação verdadeiramente democrática. Cabe ressaltar que a inserção da Libras no ambiente escolar também beneficia alunos ouvintes, ao ampliar suas habilidades comunicativas

e sociais. Assim, todos os estudantes se tornam agentes ativos na promoção da acessibilidade e da cidadania.

Figura 15 - Gráfico de respostas para “Você estaria disposto(a) a aprender Libras se houvesse um aplicativo acessível para você?”



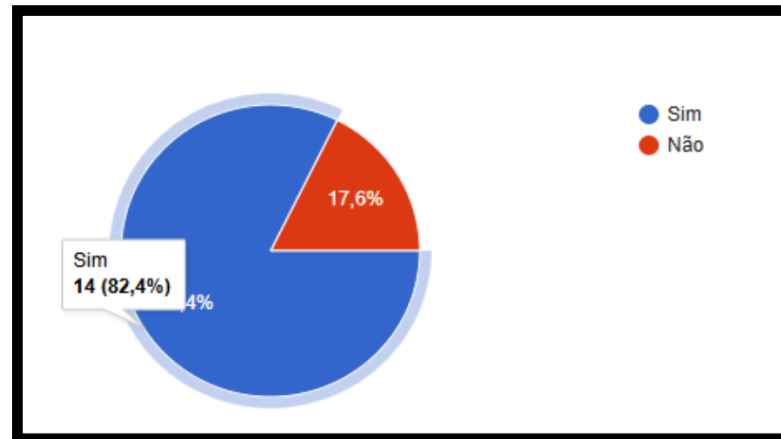
FONTE: AUTORIA PRÓPRIA, 2025

Resposta – A Figura 15 demonstra que a grande maioria dos participantes (88,2%) afirmou que estaria disposta a aprender Libras caso houvesse um aplicativo acessível. Esse dado reforça o interesse dos respondentes em adquirir conhecimento na área da acessibilidade, especialmente se houver recursos tecnológicos facilitadores.

Apenas 11,8% responderam “Não”, indicando uma baixa resistência à proposta. Esses resultados mostram que o uso de ferramentas digitais pode ser um caminho eficaz para promover a inclusão e o aprendizado da Libras. A tecnologia, nesse contexto, surge como uma aliada poderosa na disseminação do conhecimento, tornando o aprendizado mais prático, dinâmico e acessível a diferentes públicos. Aplicativos bem elaborados podem atender diversas faixas etárias e níveis de familiaridade com a Libras, contribuindo para a quebra de barreiras comunicacionais.

Além disso, o uso de recursos digitais pode motivar o engajamento contínuo dos usuários, tornando o processo de aprendizagem mais atrativo e personalizado. Essa abordagem representa um importante passo em direção a uma sociedade mais inclusiva e conectada com as necessidades da diversidade.

Figura 16 - Gráfico de respostas para “Você já enfrentou dificuldades ao tentar se comunicar com uma pessoa surda?”



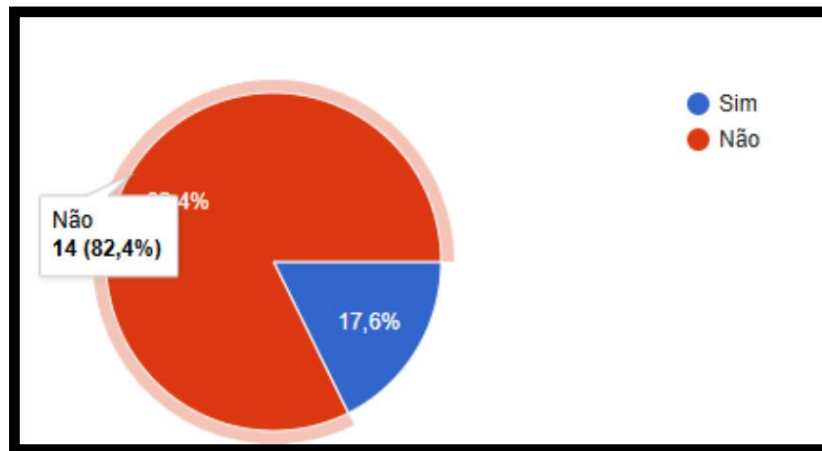
FONTE: AUTORIA PRÓPRIA, 2025

Resposta – A Figura 16 mostra que 82,4% dos participantes afirmaram já ter enfrentado dificuldades ao tentar se comunicar com uma pessoa surda. Esse dado evidencia uma barreira significativa na comunicação inclusiva, reforçando a importância de ampliar o ensino e a divulgação da Língua Brasileira de Sinais (Libras). Apenas 17,6% dos respondentes disseram não ter passado por essa situação, o que demonstra que, para a maioria, a falta de conhecimento em Libras é um obstáculo real no convívio social. Esse cenário revela a urgência de iniciativas que incentivem o aprendizado da Libras desde os primeiros anos escolares, promovendo uma convivência mais empática e acessível. Além disso, a presença de barreiras comunicacionais impacta diretamente a autonomia e a participação das pessoas surdas na sociedade.

Portanto, investir em ações que eliminem essas dificuldades, como capacitações, campanhas de conscientização e uso de tecnologias inclusivas, é essencial para fortalecer a cidadania e garantir o direito à comunicação para todos. A adoção de Libras como parte da formação continuada de professores e servidores escolares também pode ampliar o alcance das práticas inclusivas.

Projetos interdisciplinares envolvendo Libras podem ser incorporados ao currículo, aproximando os alunos da realidade das pessoas surdas e valorizando a diversidade linguística.

Figura 17 – Gráfico de respostas para “Você conhece alguma iniciativa ou projeto que promove o ensino de Libras na sua cidade ou comunidade?”



FONTE: AUTORIA PRÓPRIA, 2025

Resposta – A Figura 17 revela que 82,4% dos participantes afirmaram não conhecer nenhuma pessoa surda em seu convívio pessoal, enquanto apenas 17,6% disseram que sim. Esse dado demonstra um certo distanciamento da maioria dos respondentes em relação à comunidade surda, o que pode influenciar diretamente na percepção sobre a importância da acessibilidade e da inclusão comunicacional.

A ausência de contato com pessoas surdas no cotidiano pode resultar em menor sensibilização sobre as necessidades dessa parcela da população, contribuindo para a invisibilidade social que muitas vezes enfrentam. Essa realidade reforça a importância de aproximar a sociedade da cultura surda por meio de ações educativas, eventos culturais e espaços de diálogo, promovendo maior conscientização e empatia. Além disso, incluir a Libras e a temática da surdez no currículo escolar pode desempenhar um papel fundamental na construção de uma sociedade mais inclusiva, mesmo entre aqueles que não convivem diretamente com pessoas surdas.

Projetos de integração e vivências com a comunidade surda também podem enriquecer o processo educativo, combatendo estereótipos e incentivando a valorização da diversidade linguística e cultural. Portanto, é necessário ampliar oportunidades de interação, conhecimento e troca com a comunidade surda, tornando a inclusão uma realidade concreta e não apenas uma meta idealizada.

2.2.3.3 – Conclusão

As entrevistas evidenciam a importância da Libras como ferramenta de inclusão, ao mesmo tempo em que revelam desafios como a escassez de professores qualificados, a falta de incentivo e a dificuldade de acesso a cursos. Esses obstáculos limitam a difusão da língua em ambientes educacionais e profissionais.

A proposta de um aplicativo com uso de câmera para avaliação dos sinais foi bem recebida pelos participantes, que reconhecem na tecnologia uma aliada para facilitar o aprendizado, principalmente entre iniciantes. Ao oferecer uma solução acessível, prática e interativa, o projeto contribui para aproximar ouvintes e surdos, fortalecendo a comunicação e a inclusão social.

Investir nesse tipo de ferramenta é, portanto, um passo significativo para promover a equidade comunicacional e construir uma sociedade mais acessível e consciente da diversidade.

2.3 – PROTOTIPAGEM

A prototipagem é um processo dinâmico de criação de um modelo inicial ou protótipo, que serve como um rascunho funcional do produto final. Essa etapa permite aos desenvolvedores testar ideias, identificar problemas e realizar melhorias antes da produção em larga escala.

De maneira conceitual, a prototipagem é mais do que apenas um esboço; ela é uma exploração tangível do potencial de uma ideia, proporcionando uma visão realista de como o produto ou solução final funcionará. (QUERO BOLSA, 2023)

2.3.1 – Usuário comum

Após uma análise criteriosa das respostas fornecidas pelos usuários, as telas foram desenvolvidas com foco nas necessidades do usuário padrão. Esse processo começou com a coleta de *feedbacks* e sugestões dos usuários, o que permitiu identificar padrões e tendências essenciais para o desenvolvimento do *design*. A equipe de *design* utilizou essas informações para criar soluções que atendiam tanto aos requisitos funcionais quanto às expectativas visuais dos usuários.

Os protótipos foram criados com ênfase na usabilidade, buscando uma navegação intuitiva e agradável. A interface foi pensada para ser simples, mas eficiente, garantindo que os usuários pudessem acessar as funcionalidades de forma prática e sem dificuldades. Além disso, o apelo visual também foi cuidadosamente planejado, com a escolha de cores, tipografia e ícones que proporcionassem uma experiência agradável e harmoniosa.

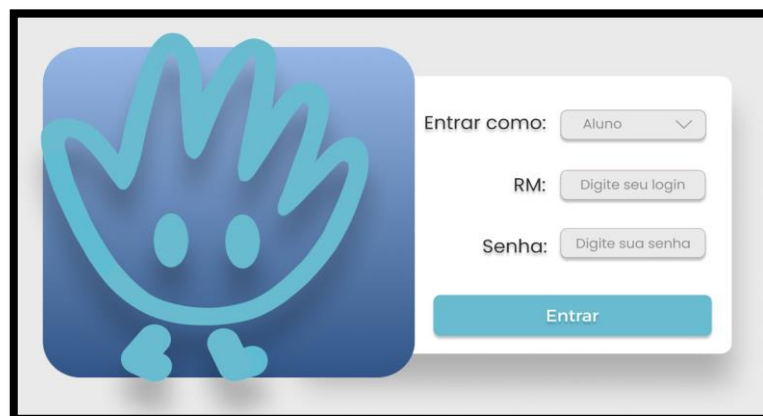
Testes com usuários reais foram conduzidos ao longo de todo o processo, permitindo validar as escolhas de *design* e realizar ajustes baseados no *feedback* obtido. Esses testes foram fundamentais para identificar pontos de melhoria, como a disposição dos elementos na tela e o fluxo de navegação, garantindo que o *design* fosse, de fato, funcional e acessível.

A cada ciclo de *feedback* e ajustes, o *design* se tornava mais refinado e a experiência do usuário mais fluida. O objetivo principal era garantir que o produto fosse não apenas funcional, mas também agradável de usar, com uma interface intuitiva e de fácil compreensão. O resultado final é uma interface moderna, com um equilíbrio entre usabilidade e estética, oferecendo uma navegação eficiente e uma experiência satisfatória ao usuário.

Em resumo, o processo de desenvolvimento foi iterativo, com constantes melhorias feitas com base no *feedback* dos usuários. Isso garantiu que, ao final, o produto atendesse de forma eficaz às necessidades do público-alvo, proporcionando uma experiência digital de alta qualidade.

2.3.1.1 – Tela inicial de login

Figura 18 - Tela de Login



FONTE: AUTORIA PRÓPRIA, 2025

Login: A tela de *login* apresentada é a interface inicial do sistema, onde o usuário realiza o processo de autenticação para acessar o ambiente interno da aplicação. Este é o ponto de entrada para todos os usuários, sendo uma etapa essencial para garantir a segurança e a personalização do acesso de acordo com a função de cada usuário.

Na interface, existe um menu suspenso (*dropdown*) que, quando clicado, permite ao usuário escolher o tipo de conta que deseja acessar. O usuário pode optar por entrar como "Aluno", "Professor" ou "Secretaria", dependendo de suas necessidades e permissões dentro do sistema. Essa funcionalidade de escolha é indicada por uma seta ao lado do menu, o que torna a navegação simples e intuitiva.

Ao lado do menu de seleção, o usuário deve inserir seu RM (Registro Matrícula) e a senha para concluir o processo de *login*. Essas informações são essenciais para a autenticação e permitem que o sistema verifique a identidade do usuário antes de conceder o acesso ao conteúdo apropriado, com base no tipo de conta escolhido.

O *design* da tela de *login* foi projetado para ser limpo e eficiente, com campos de entrada claros e uma interface visualmente agradável. O botão de "Entrar", ao ser clicado, realiza a validação das informações inseridas e direciona o usuário para a área correspondente do sistema, com base em seu perfil selecionado.

2.3.2 – Aluno

O usuário-aluno é o principal beneficiado pela aplicação, que oferece uma plataforma inovadora e acessível para o aprendizado da Língua Brasileira de Sinais (Libras). A aplicação foi projetada para atender a diferentes níveis de aprendizagem, permitindo que os alunos desenvolvam suas habilidades linguísticas de forma prática e eficiente, por meio de atividades didáticas interativas. Essas atividades são pensadas para serem dinâmicas e envolventes, o que torna o processo de ensino mais atrativo e eficaz, estimulando o aluno a se dedicar ao aprendizado de Libras.

Um dos principais diferenciais da plataforma é o uso de tecnologia de reconhecimento de gestos, que utiliza a câmera do dispositivo para identificar e avaliar a execução dos sinais feitos pelos alunos. Isso oferece um ambiente de aprendizado mais imersivo, pois o aluno pode praticar os sinais e receber *feedback* em tempo real sobre a forma como está realizando os gestos. Essa funcionalidade proporciona uma aprendizagem mais interativa, permitindo que o aluno se corrija e aprimore suas habilidades de maneira contínua.

A plataforma foi desenvolvida com um forte foco na acessibilidade. Ela oferece recursos que garantem que todos os alunos, independentemente de suas condições, possam utilizá-la. Isso inclui, por exemplo, funcionalidades de leitura em voz alta,

legendas e outras adaptações que tornam o processo de aprendizado mais inclusivo. A ideia é garantir que todos, sem exceção, possam aprender Libras de forma eficiente e sem limitações tecnológicas.

Além disso, o aplicativo visa proporcionar ao aluno autonomia em sua jornada de aprendizagem. A flexibilidade da plataforma permite que o aluno avance no seu próprio ritmo, com atividades que se ajustam ao seu nível de conhecimento. Isso é importante para manter o engajamento constante, pois o aluno tem controle sobre seu progresso e pode escolher as atividades que deseja realizar, conforme suas necessidades. Essa abordagem personalizada contribui para que o aluno se sinta mais confiante e motivado a continuar seu aprendizado.

2.3.2.1 – Tela inicial do aluno

Figura 19 - Tela inicial do aluno



FONTE: AUTORIA PRÓPRIA, 2025

Home do Aluno: A tela inicial, exibida quando o usuário acessa a plataforma como aluno, é projetada de forma simples e funcional. À esquerda, há um menu lateral fixo com o ícone do usuário, seu *e-mail* cadastrado e um botão de acesso ao perfil. Abaixo, estão os ícones de navegação para as principais seções do sistema: Início, Aulas, Atividades e *Chat*, facilitando o acesso rápido a essas áreas.

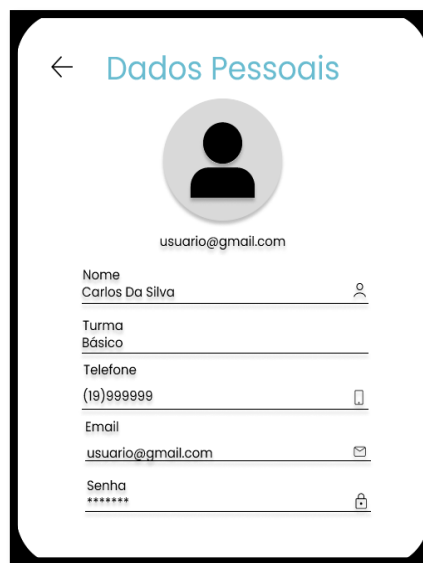
No centro da tela, o aluno encontra um painel de desempenho com quatro blocos que exibem métricas importantes sobre seu progresso: a quantidade de atividades resolvidas, aulas assistidas, capítulos concluídos e o tempo total de uso da plataforma. Esses dados são apresentados de forma clara e objetiva, utilizando barras

de progresso e números destacados, o que permite ao aluno acompanhar sua evolução no aprendizado de forma fácil e eficiente.

A interface foi projetada para ser limpa e acessível, com cores contrastantes e tipografia legível, garantindo que as informações sejam facilmente compreendidas. A disposição dos elementos e a navegação intuitiva garantem uma experiência de uso agradável e fluida.

2.3.2.2 – Tela de visualizar perfil do aluno

Figura 20 - Tela de visualizar perfil do aluno



FONTE: AUTORIA PRÓPRIA, 2025

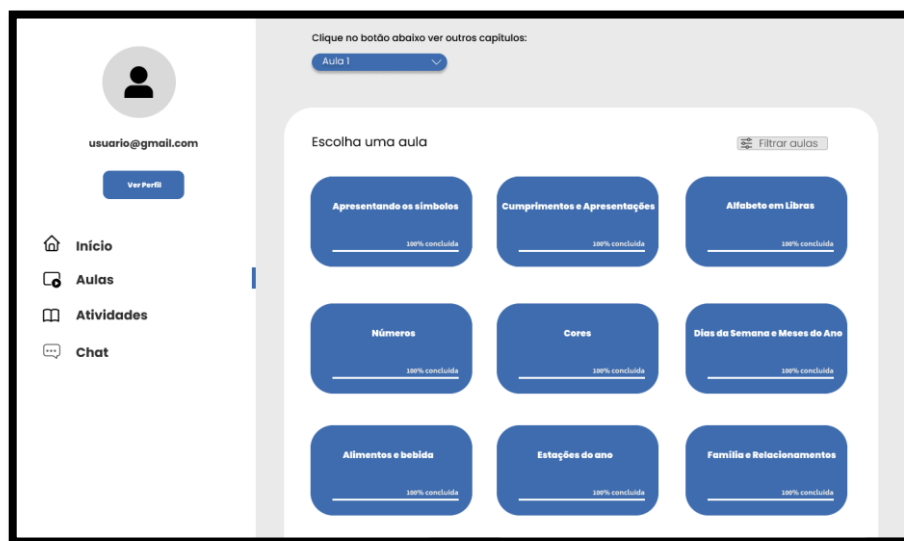
Perfil Aluno: A tela representada na figura 20 é a sobre o perfil do aluno, ele exibe as informações de forma simples e organizada. No topo da tela, encontra-se o título da seção, acompanhado de um ícone de seta, que permite ao usuário retornar à página anterior de forma rápida e intuitiva. Abaixo do título, aparece o avatar padrão do perfil do aluno, seguido do endereço de *e-mail*, que reforça a identificação do usuário e garante a personalização do acesso.

Os campos apresentados na tela reúnem as principais informações cadastradas do aluno, tais como Nome, Turma, Telefone, *E-mail* e Senha. Cada um desses campos é acompanhado por um pequeno ícone que facilita a compreensão do tipo de dado exibido, tornando a interface ainda mais intuitiva. A disposição dos campos segue um formato linear e direto, permitindo que o aluno visualize suas informações de maneira rápida e objetiva.

O *design* da tela segue a identidade visual da plataforma, priorizando simplicidade e funcionalidade. O aluno pode acessar facilmente seus dados pessoais, garantindo uma experiência prática e intuitiva. A interface é organizada e eficiente, proporcionando um ambiente amigável para o usuário. A plataforma assegura acesso rápido às informações e facilita atualizações de forma clara.

2.3.2.3 – Tela de aulas do aluno

Figura 21 - Tela aulas do aluno



FONTE: AUTORIA PRÓPRIA, 2025

Aulas do Aluno: A figura 21 representa a tela “Aulas” que apresenta ao aluno a lista de conteúdos disponíveis para estudo. No topo, há uma instrução acompanhada de um botão suspenso que permite selecionar o capítulo desejado, como “Aula 1”. Ao lado, encontra-se o recurso Filtrar aulas, que facilita a busca por conteúdos específicos.

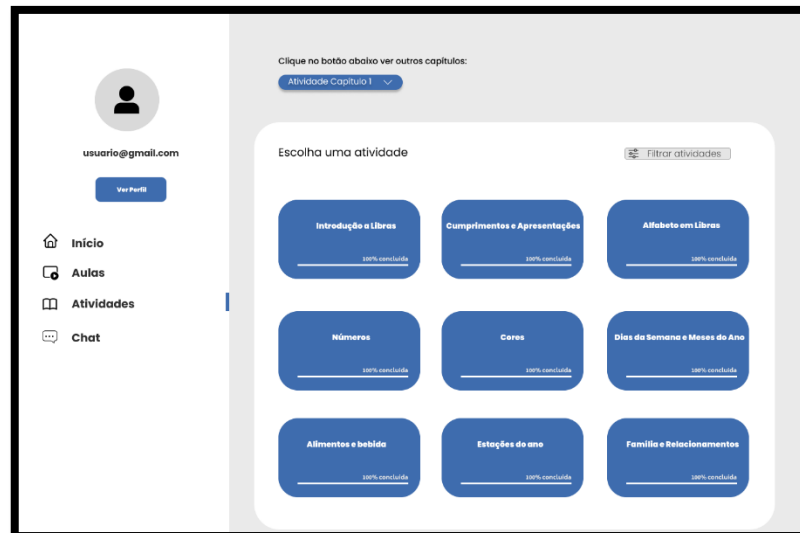
Na área central, as aulas são organizadas em cartões azuis arredondados, cada um com o título correspondente: “Apresentando os símbolos”, “Cumprimentos e Apresentações”, “Alfabeto em Libras”, “Números”, “Cores”, “Dias da Semana e Meses do Ano”, “Alimentos e bebida”, “Estações do ano” e “Família e Relacionamentos”. Abaixo de cada título, há uma barra de progresso acompanhada da informação de conclusão — neste caso, todas marcadas como 100% concluída.

No lado esquerdo da interface, o menu lateral fixo disponibiliza atalhos para Início, Aulas, Atividades e *Chat*, além do botão de “Ver Perfil”, posicionado logo abaixo do avatar e do *e-mail* do usuário. Priorizando a simplicidade e a clareza, dessa

maneira permitindo que o aluno acompanhe de forma prática seu progresso e tenha fácil acesso às aulas dentro da plataforma.

2.3.2.4 – Tela atividades do aluno

Figura 22 - Tela atividades do aluno



FONTE: AUTORIA PRÓPRIA, 2025

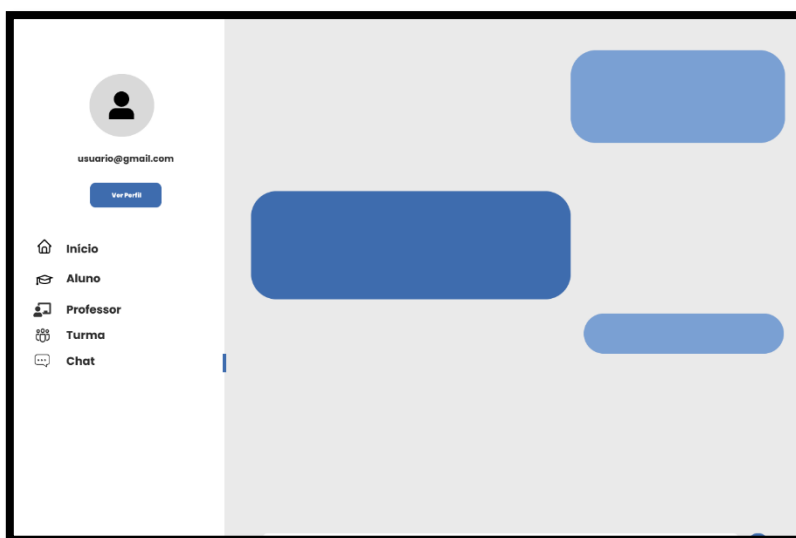
Atividade do Aluno: A figura 22 apresenta a tela “Atividades” e organiza de forma clara as tarefas disponíveis para os alunos. No topo, há uma instrução acompanhada de um botão suspenso que permite selecionar o capítulo desejado, facilitando a navegação entre diferentes conjuntos de atividades. Ao lado, encontra-se a opção “Filtrar atividades”, recurso que auxilia na busca por conteúdos específicos.

Na área principal, cada atividade é apresentada em cartões azuis arredondados, exibindo títulos como “Introdução a Libras”, “Cumprimentos e Apresentações”, “Alfabeto em Libras”, “Números”, “Cores”, “Dias da Semana e Meses do Ano”, “Alimentos e bebida”, “Estações do ano” e “Família e Relacionamentos”. Abaixo de cada título, há uma barra de progresso acompanhada da porcentagem de conclusão — neste caso, todas com 100% concluída.

À esquerda da interface, encontra-se o menu lateral fixo com ícones e opções como Início, Aulas, Atividades e *Chat*, além do botão de “Ver Perfil” abaixo do avatar e do *e-mail* do usuário. O foco é exclusivamente no consumo das atividades educacionais preparadas por docentes, o que garante uma experiência direcionada, simples e funcional para o estudante.

2.3.2.5 – Tela chat

Figura 23 - Tela chat



FONTE: AUTORIA PRÓPRIA, 2025

Chat: A figura 23 apresenta a tela *chat* que é mediada por um *chatbot* desenvolvido para responder dúvidas e fornecer orientações básicas, tanto sobre o uso da plataforma quanto sobre conteúdos relacionados à Libras. A proposta é oferecer um canal de comunicação ágil e acessível, capaz de atender rapidamente às necessidades dos usuários, promovendo maior autonomia no uso da ferramenta.

A interface do *chat* é simples e intuitiva, com uma área destinada às mensagens trocadas que permite acompanhar o histórico da conversa, além de um campo para digitação e um botão de envio, facilitando a interação direta e contínua. Esse *design* favorece a usabilidade, tornando o acesso à informação prático, mesmo para usuários com pouca familiaridade tecnológica.

Além disso, o *chatbot* funciona como um suporte inicial, orientando sobre funcionalidades básicas e esclarecendo dúvidas comuns, o que pode reduzir a necessidade de atendimento humano para questões simples, tornando o processo mais eficiente para todos os envolvidos. Essa solução também contribui para a inclusão digital, especialmente ao oferecer auxílio em um tema específico como a Libras, ampliando o alcance e a efetividade da plataforma.

2.3.3 – Professor

O professor desempenha um papel fundamental na estrutura e no funcionamento do aplicativo, atuando como mediador do processo de ensino e aprendizagem da Língua Brasileira de Sinais (Libras). Ele participa ativamente do acompanhamento do progresso dos alunos, garantindo que cada etapa do aprendizado seja compreendida e assimilada de maneira adequada. Por meio da plataforma, o docente tem a capacidade de monitorar o desempenho individual de cada usuário, identificando pontos fortes e dificuldades específicas. Com essas informações, é possível propor atividades didáticas direcionadas, exercícios práticos e estratégias pedagógicas que atendam às necessidades particulares de cada estudante, tornando o ensino mais personalizado e eficaz.

Além disso, a comunicação entre professor e aluno é facilitada por meio de recursos interativos presentes no aplicativo. Ferramentas de *feedback*, correção de exercícios, envio de mensagens e sugestões de aprimoramento permitem que o educador estabeleça um contato direto e contínuo com os alunos. Esse diálogo constante esclarece dúvidas, orienta o aprendizado e cria um vínculo mais próximo entre educador e aprendiz, promovendo um ambiente de confiança, incentivo e motivação para o estudo da Libras.

Outro ponto relevante é que o aplicativo proporciona um acesso mais democrático e inclusivo ao ensino da Libras. Usuários de diferentes regiões ou com horários variados podem se beneficiar de conteúdos adaptados e acompanhamento personalizado, o que seria difícil em um contexto exclusivamente presencial. Dessa forma, a plataforma contribui significativamente para a disseminação da Língua Brasileira de Sinais, garantindo que mais pessoas tenham a oportunidade de aprender e se comunicar de maneira eficaz.

Portanto, o papel do professor dentro do aplicativo vai muito além de ensinar: ele atua como facilitador, orientador e motivador, criando estratégias pedagógicas que promovem o aprendizado ativo e colaborativo. Ao mesmo tempo, fortalece a interação entre educador e aluno, tornando o processo de aprendizagem mais dinâmico, acessível e envolvente. O aplicativo, assim, potencializa o ensino da Libras e contribui para a construção de uma comunidade de aprendizado mais inclusiva, participativa e comprometida com o desenvolvimento integral dos estudantes.

2.3.3.1 – Tela inicial do professor

Figura 24 - Tela inicial do professor



FONTE: AUTORIA PRÓPRIA, 2025

Home professor: A figura 24 apresenta a tela inicial do professor. Ao acessar a plataforma como professor, o usuário é direcionado para uma tela inicial personalizada, desenvolvida com funcionalidades específicas voltadas à sua atuação docente, assim como ocorre na versão destinada ao aluno.

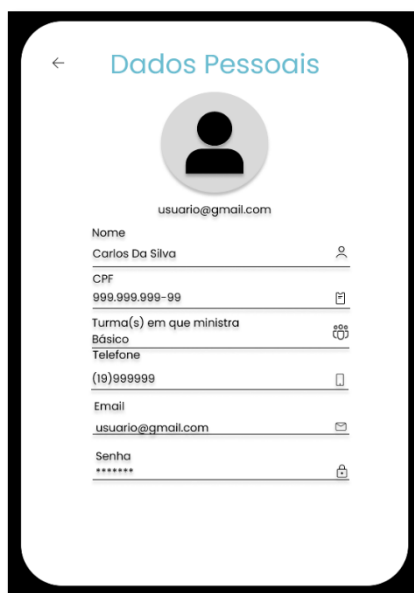
Essa tela inicial oferece uma navegação clara e objetiva, permitindo que o professor tenha acesso rápido às principais ferramentas de trabalho. No menu lateral esquerdo, o professor encontra seu ícone de usuário, o *e-mail* cadastrado e o botão “Ver Perfil”, que possibilita o acesso às informações pessoais e profissionais. Abaixo, estão dispostas as opções de navegação que incluem: Início, Aulas, Criar Aulas, Atividades, Criar Atividades, Turmas e *Chat*.

Esse menu foi projetado para facilitar o gerenciamento das turmas, a criação de conteúdos pedagógicos e a comunicação com os alunos de forma prática e eficiente. No painel principal da tela, são apresentados quatro cartões informativos que oferecem uma visão rápida e objetiva das atividades realizadas pelo docente na plataforma. Esses cartões exibem dados como o número de atividades lançadas, a quantidade de aulas criadas, o total de turmas gerenciadas e a quantidade de alunos sob sua supervisão.

Esses indicadores permitem que o professor acompanhe seu desempenho e organização dentro da plataforma, promovendo uma gestão mais eficiente do processo de ensino. Além disso, a interface intuitiva contribui para otimizar o tempo e facilitar a adaptação ao ambiente digital.

2.3.3.2 – Tela de visualizar perfil do Professor

Figura 25 - Tela de visualizar perfil do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Perfil do Professor: A figura 25 representa a tela de “Visualizar perfil do Professor” que apresenta as informações cadastrais do docente de forma organizada, clara e intuitiva, facilitando o acesso e a gestão dos dados pessoais. No topo da interface, há um ícone de seta que permite retornar à página anterior, além do título da seção em destaque, reforçando a estrutura visual da plataforma. Logo abaixo, encontra-se o avatar padrão do usuário, acompanhado do endereço de *e-mail*, compondo a identidade visual do perfil.

Na sequência, são exibidos os campos preenchidos com os dados do professor, como Nome, CPF, turma(s) em que ministra aulas, telefone, *e-mail* e senha. Cada campo é acompanhado de um pequeno ícone ilustrativo, que facilita a identificação rápida das informações apresentadas. Os dados são exibidos em campos editáveis, permitindo que o professor atualize suas informações sempre que necessário, de forma prática e segura.

Essa tela permite ao professor gerenciar suas informações pessoais de forma prática e segura. Com *design* moderno e elementos bem distribuídos, facilita a navegação e a identificação dos dados. Garante autonomia e controle sobre o perfil dentro da plataforma.

2.3.3.3 – Tela de aulas do professor

Figura 26 - Tela aulas do professor



FONTE: AUTORIA PRÓPRIA, 2025

Aulas do Professor: A figura 26 a apresenta a interface destinada às aulas do professor apresenta uma estrutura visual semelhante àquela acessada pelos alunos, mantendo os mesmos módulos temáticos voltados ao ensino de Libras — como “Cumprimentos e Apresentações”, “Alfabeto em Libras”, “Estações do Ano”, entre outros. Essa padronização facilita a navegação e a coerência entre os conteúdos disponíveis para os diferentes perfis de usuários da plataforma.

Na versão estudantil, cada aula exibe o progresso do aluno, como o indicador “100% concluído”. Já na tela do professor, essa funcionalidade não está presente, pois seu foco não é consumir o conteúdo. O docente atua como gestor, revisando e atualizando os materiais. A ausência da barra de progresso reforça seu papel organizador. Ele é responsável pela curadoria pedagógica da plataforma.

O menu lateral da plataforma disponibiliza funções exclusivas ao professor, como “Criar Aulas” e “Criar Atividades”. Essas ferramentas permitem inserir, editar e personalizar conteúdos didáticos. Diferenciam-se da versão estudantil por oferecer

recursos voltados à gestão pedagógica. Com isso, o docente ganha autonomia na organização do ensino.

2.3.3.4 – Tela de criar aulas do Professor

Figura 27 - Tela de criar aulas do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Criar Aulas: A tela da figura 27 representa a criação de aulas do professor apresenta um *layout* que reforça a função docente de gerenciar, inserir e personalizar conteúdos. Assim como na visão do aluno, há módulos temáticos organizados por categorias em Libras — como Verbos, Esportes, Alfabeto, Números, Objetos, Expressões Faciais, Animais, Lugares, Cumprimentos, Apresentações e Emoções — , promovendo uma navegação intuitiva e focada em temas visuais. No entanto, em vez de exibir progresso de aprendizagem, o professor tem acesso a ferramentas específicas para montar sua própria aula de maneira prática e direcionada.

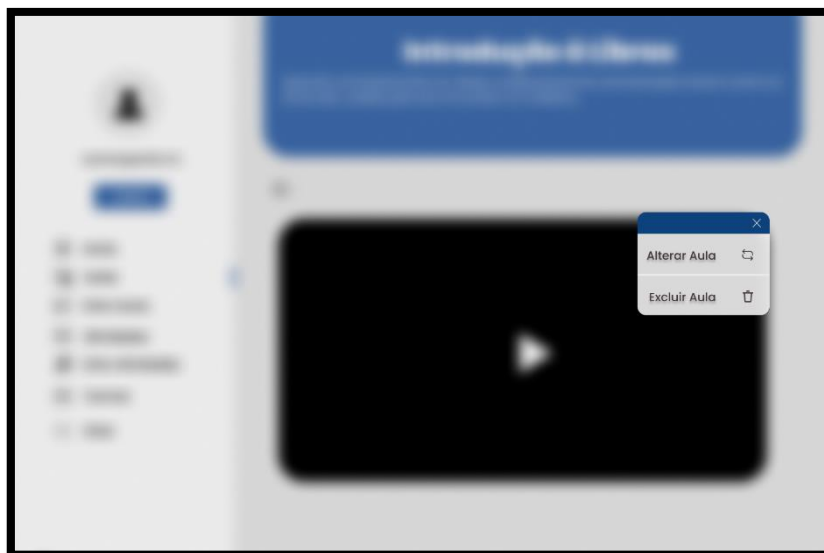
O menu lateral contém itens gerais como Início, Aulas, Atividades, Turmas e *Chat*, mas acrescenta opções exclusivas do professor: Criar Aulas e Criar Atividades, que permitem elaborar e publicar novos materiais didáticos. Na área central, a tela oferece campos para inserir o nome da aula, selecionar a turma e escolher categorias temáticas; além disso, há a funcionalidade de *upload* de vídeo, permitindo arrastar arquivos ou selecionar diretamente pelo botão, seguida de um espaço para a descrição da aula com observações pedagógicas.

Essa estrutura evidencia que, para o professor, o foco está na produção e gestão de conteúdos pedagógicos em Libras, com recursos que valorizam a

linguagem visual e facilitam o planejamento educacional, em contraste com o aluno, cujo foco é consumir, praticar e acompanhar o progresso do aprendizado.

2.3.3.5 – Tela de alterar e excluir aulas do Professor

Figura 28 – Tela de alterar e excluir aulas do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Alterar e Excluir Aula: A figura 28 representa a funcionalidade de alterar e excluir aulas representa um recurso essencial de gerenciamento disponível exclusivamente para professores na plataforma educacional, oferecendo autonomia total na curadoria dos conteúdos publicados. A interface apresenta um *player* de vídeo central que exibe o conteúdo previamente cadastrado, permitindo ao docente visualizar integralmente a aula antes de realizar qualquer modificação, o que garante maior precisão e segurança nas decisões pedagógicas.

No canto superior direito do vídeo, há um *card* flutuante com duas opções visuais bem definidas: “Alterar Aula”, com ícone de atualização, e “Excluir Aula”, com ícone de lixeira. A primeira opção permite substituir o vídeo, atualizar o nome da aula ou ajustar outras informações relevantes conforme o planejamento didático; já a segunda remove a aula da plataforma de forma definitiva, sendo recomendada apenas quando o conteúdo não for mais útil ou pertinente.

O *design* da interface é simples, direto e funcional, promovendo uma navegação fluida e intuitiva, com ênfase na clareza visual e na acessibilidade. Essa organização favorece o controle completo por parte do professor sobre os materiais

publicados, tornando o processo de edição e exclusão de aulas mais eficiente, prático e alinhado às necessidades do ambiente educacional.

Com isso, a plataforma reforça o papel ativo do professor na construção e manutenção do acervo didático, além de incentivar o aprimoramento contínuo dos conteúdos. A possibilidade de revisar e atualizar materiais garante que as aulas permaneçam coerentes com as metas de aprendizagem e as demandas dos alunos.

2.3.3.6 – Tela de alterar aulas do Professor

Figura 29 - Tela criar atividade do professor



FONTE: AUTORIA PRÓPRIA, 2025

Alterar Aula: A figura 29 representa a tela “Alterar Aula” que é uma funcionalidade destinada ao professor para editar conteúdos já cadastrados. No topo, encontra-se o título em destaque, seguido do campo Nome da Aula, onde é possível visualizar e modificar o título previamente inserido. Logo abaixo, há a seleção de turma, que organiza os conteúdos de acordo com o nível, como “Básico”.

A seção de Categorias aparece em formato de botões horizontais, destacando opções como “Verbos”, “Esportes”, “Alfabeto”, “Números”, “Objetos”, “Expressões Faciais”, “Animais”, “Lugares”, “Cumprimentos”, “Apresentações” e “Emoções”. Essa organização facilita a classificação da aula e garante uma curadoria eficiente.

Mais abaixo, há o campo de vídeo selecionado, exibindo o arquivo já associado, neste caso “IntroduçãoLibras.mp4”. Em seguida, a área de Descrição da Aula permite ao professor inserir instruções ou explicações complementares sobre o conteúdo, promovendo maior clareza e contextualização para os alunos. Na parte

inferior, encontra-se o botão de ação Alterar Aula, que finaliza o processo de edição. Essa tela oferece uma experiência prática e organizada, assegurando flexibilidade e controle ao professor na atualização de suas aulas dentro da plataforma.

2.3.3.7 – Tela de excluir aulas do Professor

Figura 30 - Tela de excluir aulas do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Excluir Aula: A tela representada na figura 30 foi projetada para oferecer ao professor um espaço claro e organizado para remover conteúdos cadastrados na plataforma. Logo no topo, o título em destaque indica a ação a ser realizada, trazendo clareza imediata ao usuário. Abaixo, encontram-se os campos com informações da aula, como o nome e a turma, que permitem confirmar se o conteúdo selecionado está correto antes da exclusão.

Um menu de categorias é exibido em formato de botões arredondados, facilitando a identificação da área de conhecimento relacionada à aula, como “Verbos”, “Alfabeto” ou “Esportes”. Esse recurso contribui para a navegação e organização dos conteúdos, ajudando o professor a localizar rapidamente o material.

Mais abaixo, a interface mostra o vídeo selecionado e a descrição da aula, onde são apresentados os objetivos e a proposta pedagógica do conteúdo. Isso garante que o usuário tenha todas as informações necessárias para validar sua decisão de exclusão.

Por fim, o botão Excluir Aula, destacado na parte inferior, oferece uma ação direta e acessível, mantendo a simplicidade do *design* e evitando confusões.

A presença do menu lateral, com atalhos como Início, Aulas, Atividades, Turmas e *Chat*, garante acesso rápido a outras funções, favorecendo a usabilidade e a continuidade do trabalho dentro da plataforma.

2.3.3.8 – Tela criar Atividades do Professor

Figura 31 - Tela criar atividade do professor



FONTE: AUTORIA PRÓPRIA, 2025

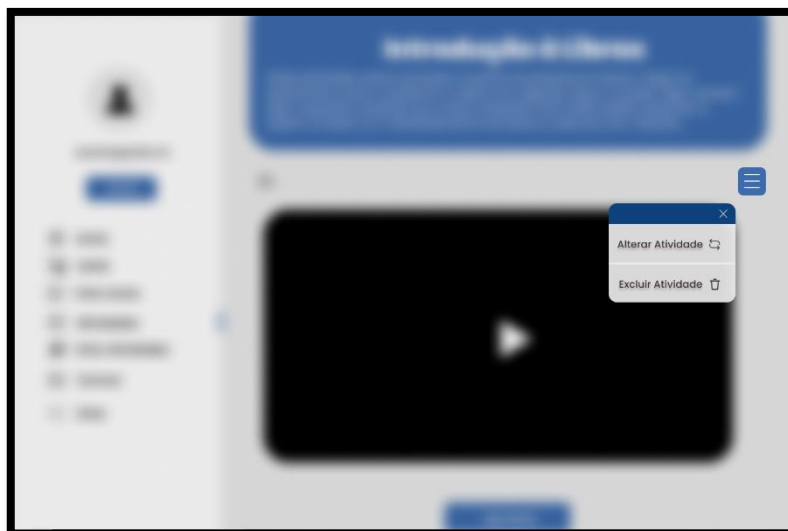
Criar atividade: A tela “Criar Atividade” representada na figura 31 é uma funcionalidade exclusiva para professores, destinada à criação de conteúdos interativos que estejam alinhados aos objetivos pedagógicos do ensino de Libras. No topo da interface, há uma barra horizontal com várias categorias temáticas, como “Verbos”, “Alfabeto”, “Números”, “Objetos”, “Expressões Faciais” e “Emoções”, que auxiliam na organização e classificação dos conteúdos, facilitando a curadoria e a localização posterior pelos alunos. Logo abaixo, o professor pode inserir o nome da atividade no campo apropriado, garantindo uma identificação clara do conteúdo produzido. Além disso, há um menu suspenso para selecionar a turma que receberá a atividade, o que torna o gerenciamento mais eficiente e direcionado.

A página é organizada para priorizar a usabilidade e a clareza, com áreas bem definidas e fáceis de navegar. No centro, existe um espaço dedicado à descrição da atividade, onde o docente pode adicionar explicações detalhadas, orientações para o uso dos sinais, ou contextualizações práticas que enriquecem o material. Embora esta tela não permita o envio direto de vídeos, como acontece na criação de aulas, o foco está na elaboração textual e pedagógica das tarefas. Ao final da página, o botão “Criar

Atividade” permite concluir o processo de forma rápida e intuitiva, promovendo a autonomia do professor na construção e personalização do conteúdo didático.

2.3.3.9 – Tela de alterar e excluir atividades do Professor

Figura 32 - Tela de alterar e excluir atividades do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Alterar e Excluir Atividade: A figura 32 representa as telas destinadas ao professor para alterar e excluir atividades são essenciais para a gestão eficiente dos conteúdos em Libras dentro da plataforma. Diferente da visão do aluno, que foca em indicadores de progresso, o professor conta com ferramentas específicas que permitem modificar ou remover atividades já criadas, garantindo assim um controle completo sobre o material pedagógico disponível. Na interface, o docente pode assistir ao vídeo associado à atividade, o que facilita a análise detalhada do conteúdo antes de realizar qualquer alteração, assegurando que as mudanças estejam alinhadas com os objetivos pedagógicos. Além disso, um menu lateral apresenta as opções “Alterar Atividade” e “Excluir Atividade”, proporcionando flexibilidade para atualizar ou eliminar tarefas conforme as necessidades da turma ou do planejamento.

Essa funcionalidade permite ao professor manter o acervo didático atualizado e alinhado às necessidades dos alunos, tornando o ensino mais dinâmico. O *design* simples e intuitivo facilita a navegação e reforça a autonomia docente. Com opções claras, o professor gerencia os recursos educacionais de forma prática e ágil. Isso valoriza a curadoria pedagógica, garantindo conteúdos relevantes. Assim, a criação e revisão das atividades ficam mais acessíveis e eficientes.

2.3.3.10 – Tela de alterar atividades do Professor

Figura 33 - Tela de alterar atividades do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Alterar Atividade: A tela da figura 33 representa a alteração de atividades oferece ao professor um ambiente completo e organizado para editar conteúdos já criados, garantindo uma experiência de uso intuitiva e eficiente. No menu lateral esquerdo, a navegação permanece padronizada, com opções essenciais como Início, Aulas, Criar Aulas, Atividades, Criar Atividades, Turmas e *Chat*, além do ícone de usuário, endereço de *e-mail* e o botão “Ver Perfil”, o que facilita a movimentação pelo sistema sem perder o contexto.

No painel principal, o docente encontra campos específicos para modificar informações importantes da atividade, como o nome, a turma e o nível de dificuldade, além das categorias temáticas que incluem Verbos, Esportes, Números, Alfabeto, Objetos, Expressões Faciais, Animais, Lugares, Cumprimentos, Apresentações e Emoções, organizadas para otimizar a curadoria e a seleção dos conteúdos. Logo abaixo dessas opções, há um espaço destinado à descrição detalhada da atividade, onde o professor pode revisar, complementar ou modificar as instruções e orientações destinadas aos alunos, assegurando que o material esteja sempre claro, completo e alinhado com os objetivos pedagógicos.

Para finalizar, um botão de ação intitulado “Alterar Atividade” permite que o docente salve todas as alterações feitas de maneira rápida e segura, garantindo a atualização constante do conteúdo. Essa funcionalidade reforça o papel do professor

como gestor e mediador do conhecimento, possibilitando que ele tenha total autonomia para adaptar, aprimorar e personalizar o material pedagógico em Libras, tornando o processo educacional mais dinâmico, eficaz e alinhado às necessidades específicas de cada turma.

2.3.3.11 – Tela de excluir atividades do Professor

Figura 34 - Tela de excluir atividades do Professor



FONTE: AUTORIA PRÓPRIA, 2025

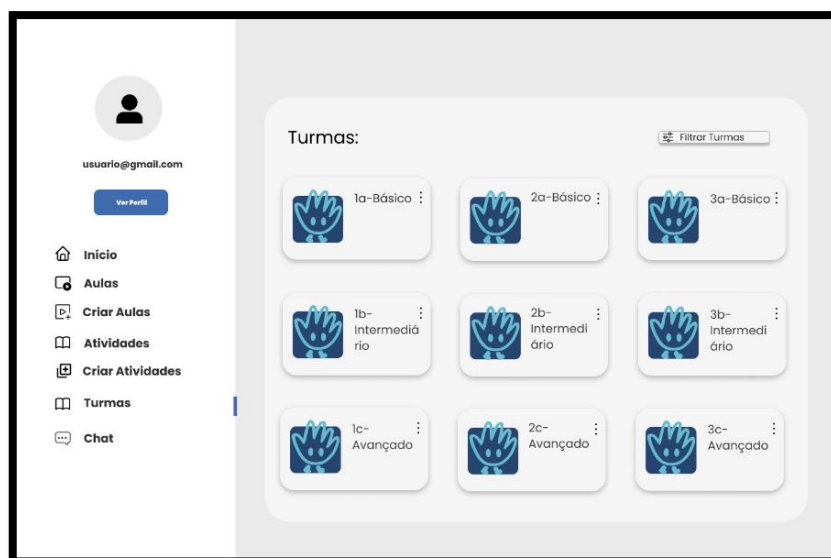
Excluir Atividade: Na tela da figura 34 é representado a exclusão de atividades, o professor tem acesso às informações principais da atividade cadastrada antes de confirmar sua remoção. No menu lateral esquerdo, permanecem o ícone de usuário, *e-mail*, botão “Ver Perfil” e as opções de navegação — Início, Aulas, Criar Aulas, Atividades, Criar Atividades, Turmas e *Chat* — organizadas para facilitar a gestão.

No painel central, são exibidos os dados da atividade selecionada: nome da atividade, turma e categoria temática (neste caso, Números). Abaixo, encontra-se a descrição da atividade, que explica os objetivos pedagógicos e instruções destinadas aos alunos. Ao final, o botão de ação “Excluir Atividade” permite ao docente remover definitivamente o conteúdo da plataforma.

Essa funcionalidade garante que o professor possa manter o ambiente sempre atualizado, eliminando materiais que não sejam mais necessários ou que precisem ser substituídos.

2.3.3.12 – Tela de turmas do Professor

Figura 35 - Tela de turmas do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Turmas: A tela da figura 35 apresenta de forma clara e organizada os grupos de ensino disponíveis na plataforma. Na parte superior, o título “Turmas” é acompanhado por um botão suspenso com a opção “Filtrar Turmas”, permitindo ao usuário refinar a visualização conforme critérios específicos, como nível ou disciplina. A área principal exibe os cartões das turmas em um *layout* de grade, com formato arredondado e fundo branco, cada um contendo um ícone ilustrativo e o nome da turma.

Os títulos seguem uma progressão por níveis: “1a-Básico”, “2a-Básico” e “3a-Básico” representam o nível inicial; “1b-Intermediário”, “2b-Intermediário” e “3b-Intermediário” indicam o nível médio; e “1c-Avançado”, “2c-Avançado” e “3c-Avançado” correspondem ao nível mais avançado. Essa estrutura facilita a identificação rápida das turmas e promove uma navegação eficiente entre os diferentes estágios de aprendizado.

À esquerda da interface, o menu lateral fixo exibe o avatar do usuário, seguido pelo e-mail “usuario@gmail.com” e o botão azul “Ver Perfil”. Abaixo, estão as opções de navegação: Início, Aulas, Criar Aulas, Atividades, Criar Atividades, Turmas e *Chat*. O foco da tela está na gestão das turmas disponíveis, oferecendo uma experiência funcional e intuitiva tanto para alunos quanto para professores, com acesso direto às informações essenciais para o acompanhamento das atividades educacionais.

2.3.4 – Secretaria

A secretaria possui uma função administrativa fundamental dentro da plataforma, sendo responsável pela gestão e organização dos cadastros de usuários, como alunos e professores. Por meio de um painel específico, é possível realizar o controle das turmas, atribuir professores, acompanhar a frequência de uso e manter atualizadas as informações institucionais relacionadas ao sistema.

A atuação da secretaria contribui para o bom funcionamento do ambiente virtual, garantindo que os dados estejam organizados, seguros e acessíveis, o que reflete diretamente na eficiência do processo educacional. Assim, a secretaria desempenha um papel essencial na estruturação e manutenção da plataforma, assegurando suporte operacional aos demais usuários.

2.3.4.1 – Tela inicial da secretaria

Figura 36 - Tela inicial da secretaria



FONTE: AUTORIA PRÓPRIA, 2025

Home Secretaria: A tela da figura 36 representa o *home* ao acessar a plataforma com o perfil de secretaria, o usuário visualiza uma tela inicial voltada à gestão e acompanhamento geral do sistema educacional. À esquerda, o menu de navegação reúne as principais seções: Início, Aluno, Professor, Turma e *Chat*, facilitando o gerenciamento integrado das informações institucionais.

No painel principal, são exibidos quatro cartões com os dados estatísticos mais relevantes para a administração: o número total de turmas criadas, professores cadastrados, alunos registrados e um resumo com os dados gerais, que neste caso

totalizam 95 entradas no sistema. Essa visualização oferece à equipe administrativa uma visão clara do funcionamento e da evolução da plataforma, permitindo tomada de decisões mais rápida e eficiente.

2.3.4.2 – Tela de visualizar perfil da secretaria

Figura 37 - Tela aluno da secretaria



FONTE: AUTORIA PRÓPRIA, 2025

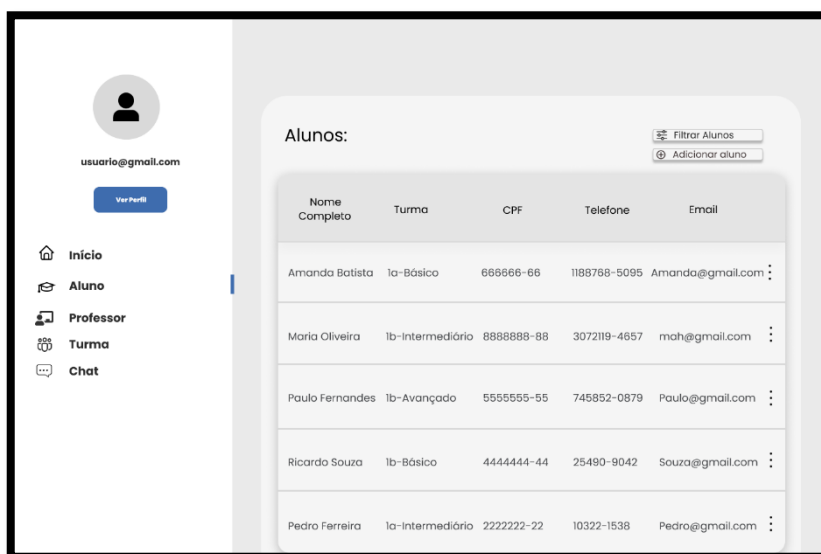
Perfil Secretaria: Ao acessar a plataforma na seção “Dados Pessoais” que é representado na figura 37, o usuário é direcionado a uma tela voltada à administração de suas informações pessoais. No topo da interface, uma seta de retorno posicionada à esquerda permite a navegação para a tela anterior, reforçando a fluidez da experiência.

O painel principal apresenta de forma centralizada o avatar genérico do usuário, seguido pelo *e-mail* cadastrado: *usuario@gmail.com*. Abaixo, três campos organizam os dados editáveis: telefone, *e-mail* e senha. O número de telefone é exibido no formato nacional (19)999999, enquanto o campo de *e-mail* repete o endereço principal do usuário. A senha aparece mascarada com asteriscos (*****), garantindo a privacidade das credenciais. Ícones de ação ao lado dos campos de *e-mail* e senha indicam funcionalidades de edição ou visualização, permitindo ajustes rápidos diretamente na interface.

Essa tela oferece ao usuário controle direto sobre suas informações, promovendo segurança e autonomia na gestão do perfil. A disposição limpa e objetiva dos elementos contribui para uma navegação intuitiva, essencial para uma experiência digital eficiente e personalizada.

2.3.4.3 – Tela alunos secretaria

Figura 38 - Tela aluno da secretaria



Nome Completo	Turma	CPF	Telefone	Email
Amanda Batista	1a-Básico	666666-66	1188768-5095	Amanda@gmail.com
Maria Oliveira	1b-Intermediário	8888888-88	3072119-4657	mah@gmail.com
Paulo Fernandes	1b-Avançado	5555555-55	745852-0879	Paulo@gmail.com
Ricardo Souza	1b-Básico	4444444-44	25490-9042	Souza@gmail.com
Pedro Ferreira	1a-Intermediário	2222222-22	10322-1538	Pedro@gmail.com

FONTE: AUTORIA PRÓPRIA, 2025

Visualizar Alunos: A tela da figura 38 é destinada à secretária para o gerenciamento de alunos tem como principal objetivo facilitar o controle e a organização das informações dos estudantes cadastrados no sistema. Nessa interface, é apresentada uma tabela que exibe os dados essenciais de cada aluno, como nome completo, turma, CPF, telefone e *e-mail*. Essas informações são organizadas de forma clara, permitindo uma visualização rápida e eficiente.

Na parte superior da tela, estão disponíveis dois botões funcionais: "Filtrar Alunos", que possibilita aplicar filtros específicos para localizar alunos com mais agilidade (por nome, turma ou outros critérios), e "Adicionar aluno", que direciona a secretária para um formulário de cadastro, onde é possível inserir os dados de novos estudantes no sistema.

Cada linha da tabela de alunos possui um botão de opções (representado por três pontos), que permite realizar ações específicas, como editar ou excluir o cadastro

do aluno, ou ainda visualizar mais detalhes do seu perfil, garantindo assim um controle completo e eficiente das informações.

2.3.4.3 – Tela secretaria professor

Figura 39 - Tela professor da secretaria

Nome Completo	Número de matrícula	CPF	Turma(s) que ministra	Telefone	Email
Amanda Batista	23254	666666-66	1a-Básico	1188768-5095	Amanda@gmail.com
Maria Oliveira	45232	8888888-88	2a-Básico	3072119-4657	mah@gmail.com
Paulo Fernandes	22345	5555555-55	3a-Básico	745852-0879	Paulo@gmail.com
Ricardo Souza	34252	4444444-44	1b-Básico	25490-9042	Souza@gmail.com
Pedro Ferreira	52243	2222222-22	2b-Básico	10322-1538	Pedro@gmail.com

FONTE: AUTORIA PRÓPRIA, 2025

Visualizar Professor: Na tela de professores representada na figura 39, o sistema apresenta uma tabela com estrutura semelhante à utilizada na tela de alunos, garantindo padronização e facilidade de navegação. A tabela exibe campos importantes para a gestão docente, como nome completo, número de matrícula, CPF, turma(s) que ministra, telefone e *e-mail*, permitindo uma visão clara e objetiva das informações de cada professor. Na parte superior da interface, estão disponíveis os botões “Filtrar Professor” e “Adicionar Professor”.

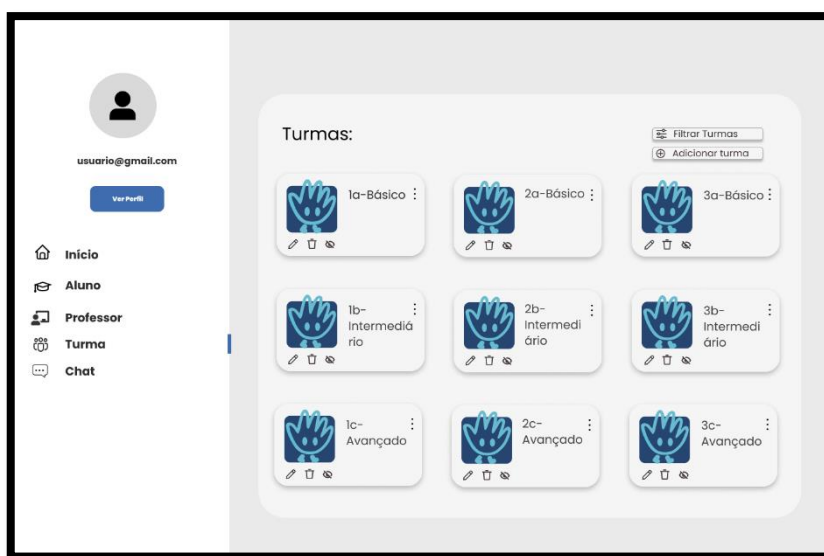
O primeiro permite refinar a busca por critérios específicos, facilitando a localização de docentes em meio a um grande número de registros. Já o segundo possibilita o cadastro de novos profissionais de forma rápida e eficiente, agilizando os processos administrativos. Cada linha da tabela possui um botão de opções (representado por três pontos), que possibilita ações como editar ou excluir o cadastro, ou ainda visualizar informações adicionais do professor, proporcionando um controle mais detalhado.

Essa organização contribui para uma gestão mais prática e centralizada dos dados dos professores, otimizando o trabalho da secretaria escolar e garantindo maior

eficiência no gerenciamento. Além disso, a interface foi desenvolvida pensando na usabilidade, com um *layout* intuitivo que facilita a navegação mesmo para usuários com pouca familiaridade tecnológica.

2.3.4.4 – Tela secretaria turmas

Figura 40 – Tela secretaria turmas



FONTE: AUTORIA PRÓPRIA, 2025

Visualizar Turmas: Na tela de turmas representada na figura 40, o sistema apresenta um painel visual organizado, composto por cartões que representam cada turma cadastrada. Esses cartões são distribuídos conforme o nível da turma — básico, intermediário e avançado — e, dentro de cada nível, subdivididos por séries (1^a, 2^a e 3^a).

Cada cartão exibe informações essenciais da turma e conta com ícones de ação visíveis e intuitivos, que permitem à secretária realizar operações como editar os dados da turma, excluí-la do sistema ou acessar seus detalhes completos com facilidade.

Na parte superior da tela, estão localizados os botões “Adicionar turma” e “Filtrar turmas”. O primeiro possibilita o cadastro de novas turmas de forma rápida e direta, enquanto o segundo permite aplicar filtros para refinar a visualização, facilitando a navegação entre as turmas já existentes e otimizando a gestão acadêmica. Essa organização visual contribui para uma experiência de uso mais

eficiente, especialmente em instituições com grande número de turmas e níveis de ensino.

2.3.4.5 – Filtro

Figura 41 – Tela filtro



FONTE: AUTORIA PRÓPRIA, 2025

Filtro: A interface representada na figura 41 oferece diversas ferramentas que facilitam a busca por turmas específicas. À esquerda, destaca-se uma caixa de pesquisa contendo um campo para digitar o nome da turma, acompanhado por um ícone de lupa, que simboliza a função de busca. Abaixo desse campo, estão posicionados dois botões em destaque na cor azul: “Limpar”, que permite ao usuário apagar rapidamente o texto digitado para iniciar uma nova pesquisa, e “Buscar”, que executa a procura pelo termo inserido, agilizando o acesso às informações desejadas.

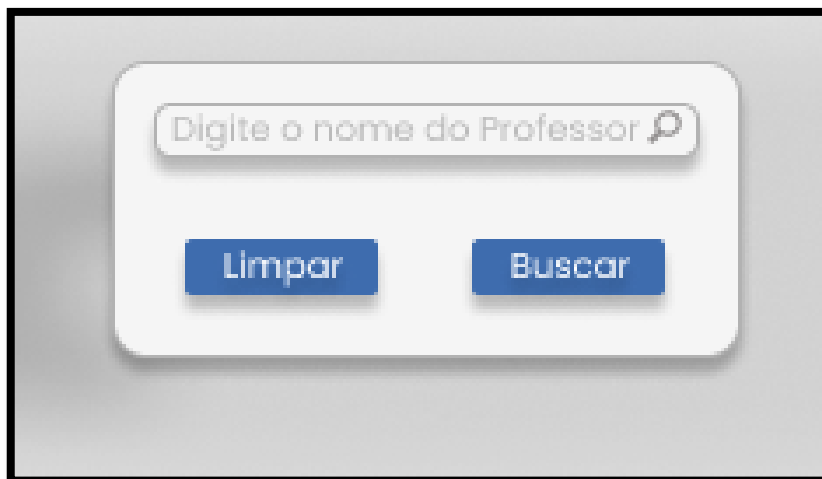
Do lado direito, um painel suspenso de filtragem traz o título “Filtrar Turma Por:”, oferecendo opções para refinar os resultados com base em critérios importantes, como o Nome da turma ou o Nível de ensino. A interface apresenta ainda os botões “Sair” e “Filtrar”, localizados na parte inferior desse painel e exibidos em tons de cinza, que possibilitam ao usuário cancelar a operação ou aplicar o filtro selecionado, respectivamente.

Essa combinação de elementos visuais e funcionais torna a navegação mais eficiente e prática, facilitando tanto para estudantes quanto para professores a tarefa de localizar turmas específicas. A usabilidade intuitiva garante que o processo seja rápido, sem necessidade de passos complexos, promovendo uma experiência agradável, fluida e direta dentro da plataforma. Além disso, o *design* claro e

organizado contribui para reduzir erros e aumentar a satisfação do usuário, tornando o uso da ferramenta acessível e eficiente para todos os perfis.

2.3.4.6 – Tela Barra de pesquisa

Figura 42 – Tela barra de pesquisa



FONTE: AUTORIA PRÓPRIA, 2025

Barra de Pesquisa: A interface representada na figura 42 apresenta uma barra de pesquisa simples e eficiente, especialmente desenvolvida para facilitar a localização de professores dentro da plataforma. Na parte superior, há um campo de texto com o *placeholder* “Digite o nome do Professor”, que pode variar conforme o perfil do usuário, acompanhado por um ícone de lupa que simboliza a função de busca, tornando a ação intuitiva e facilmente reconhecível.

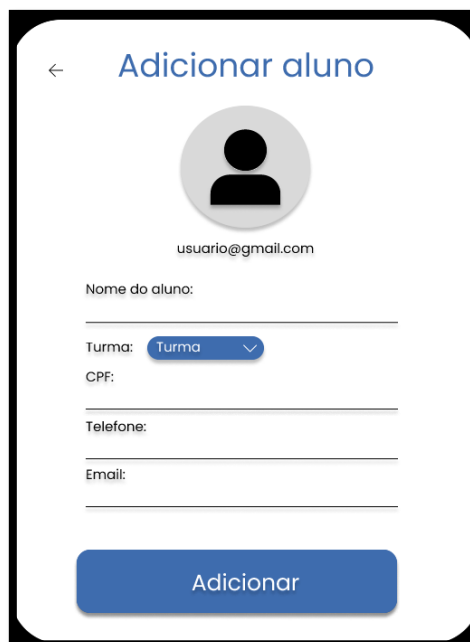
Abaixo desse campo, estão posicionados dois botões destacados em azul: “Limpar” e “Buscar”. O botão “Limpar” permite ao usuário apagar rapidamente qualquer texto digitado, reiniciando a pesquisa para um novo começo, enquanto o botão “Buscar” confirma a consulta, ativando a filtragem dos dados para exibir os resultados correspondentes.

Esse conjunto de elementos oferece uma experiência prática e ágil, garantindo que a navegação seja fluida e organizada. Além disso, a organização visual clara contribui para a usabilidade, tornando o processo de busca simples, rápido e acessível para todos os usuários da plataforma. Essa facilidade na pesquisa ajuda a economizar tempo e evita frustrações, especialmente em ambientes com grande volume de dados. Por fim, o *design* minimalista e funcional garante que o recurso seja intuitivo, mesmo para usuários com pouca familiaridade tecnológica. Dessa forma, a

barra de pesquisa reforça a eficiência da plataforma como ferramenta de gestão educacional.

2.3.4.7 – Tela de Adicionar Aluno Secretaria

Figura 43 – Tela de adicionar aluno secretaria



A imagem mostra a tela de "Adicionar aluno" de uma interface de usuário. No topo, há um ícone de seta para trás e o título "Adicionar aluno". Abaixo, um avatar de usuário é exibido com o e-mail "usuario@gmail.com". O formulário contém os seguintes campos: "Nome do aluno:" com uma linha de texto; "Turma:" com um menu suspenso que mostra "Turma" e um ícone de seta para baixo; "CPF:" com uma linha de texto; "Telefone:" com uma linha de texto; e "Email:" com uma linha de texto. No final da tela, há um botão azul com o texto "Adicionar".

FONTE: AUTORIA PRÓPRIA, 2025

Adicionar Aluno: A tela “Adicionar Aluno” representada na figura 43 foi projetada para facilitar o registro de novos estudantes na plataforma, garantindo um processo ágil e organizado. No topo da interface, o avatar do usuário é exibido junto ao *e-mail* principal (*usuario@gmail.com*), reforçando a identidade de quem está realizando o cadastro e proporcionando um ambiente personalizado.

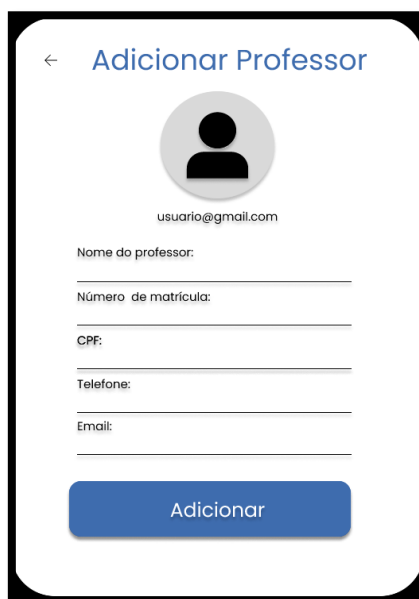
Logo abaixo, são apresentados os campos de preenchimento obrigatórios: “Nome do aluno”, “Turma” (com menu suspenso para seleção do grupo), “CPF”, “Telefone” e “*E-mail*”. Cada campo é claramente identificado, facilitando o preenchimento e garantindo a padronização dos dados inseridos. A utilização de menus suspensos e campos de texto contribui para uma experiência de usuário mais fluida e sem ambiguidades.

Ao final da página, um botão destacado “Adicionar” permite concluir o registro de forma prática e rápida, promovendo uma experiência intuitiva para o usuário. Essa organização contribui para evitar erros e agiliza o processo de cadastro, essencial para instituições que necessitam de eficiência na gestão de alunos.

Além disso, a interface pode incluir validações em tempo real, como a verificação do formato correto do CPF e do *e-mail*, garantindo que os dados inseridos estejam corretos antes de serem salvos. Essa funcionalidade reduz a necessidade de correções posteriores e assegura a integridade das informações no sistema.

2.3.4.8 – Tela de Adicionar Professor Secretaria

Figura 44 – Tela de adicionar professor secretaria



A imagem mostra uma tela de formulário com o título "Adicionar Professor" e uma seta para voltar. Abaixo do título há um ícone de perfil de usuário e o e-mail "usuario@gmail.com". O formulário possui cinco campos de entrada: "Nome do professor:", "Número de matrícula:", "CPF:", "Telefone:" e "Email:". Um botão azul arredondado com o texto "Adicionar" está na base da tela.

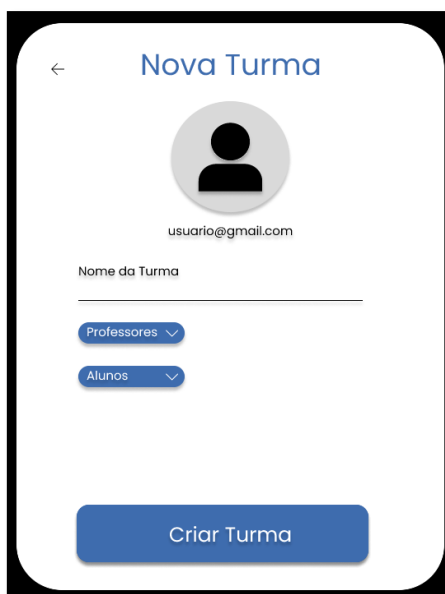
FONTE: AUTORIA PRÓPRIA, 2025

Adicionar Professor: A interface representada na figura 44 organiza de forma simples o processo de cadastro de novos docentes no sistema. Na parte superior, o título “Adicionar Professor” aparece em destaque, acompanhado de uma seta para retornar à página anterior. Logo abaixo, há um avatar genérico com um *e-mail* exibido como identificação do usuário. Na área central, são disponibilizados campos para preenchimento de dados importantes: nome do professor, número de matrícula, CPF, telefone e *e-mail*. Esses campos estão alinhados verticalmente, proporcionando clareza e facilitando o fluxo de inserção das informações.

Ao final da tela, encontra-se um botão azul arredondado com a palavra “Adicionar”, que confirma a ação e registra o professor no sistema. O *design* mantém a objetividade e garante uma experiência simples e intuitiva, favorecendo a usabilidade no processo de cadastro.

2.3.4.9 – Tela de Adicionar Turma

Figura 45 – Tela de adicionar turma



FONTE: AUTORIA PRÓPRIA, 2025

Adicionar Turma: A interface representada na figura 45 foi projetada para a criação de novas turmas no sistema de forma clara e objetiva. Na parte superior, o título “Nova Turma” aparece em destaque, acompanhado de uma seta que possibilita retornar à tela anterior, facilitando a navegação do usuário. Logo abaixo, há a exibição de um avatar genérico com o *e-mail* de identificação do usuário, reforçando a personalização da experiência.

Na área central, o campo “Nome da Turma” permite inserir a identificação da nova turma a ser criada, essencial para a organização e localização futura dentro da plataforma. Abaixo, encontram-se dois menus suspensos: o primeiro destinado à seleção de professores, permitindo a atribuição rápida e eficiente dos docentes responsáveis, e o segundo para a escolha dos alunos, possibilitando compor a turma de maneira organizada e prática.

Na parte inferior da tela, destaca-se o botão azul “Criar Turma”, que representa a ação principal da página, finalizando o processo de cadastro. A interface entrega uma experiência funcional, direta e intuitiva ao usuário, simplificando tarefas administrativas e promovendo agilidade na gestão escolar. Além disso, a disposição visual equilibrada e o uso de elementos interativos contribuem para uma interação fluida e agradável dentro do sistema.

2.3.4.10 – Tela de Alterar Aluno

Figura 46 – Tela de alterar aluno

A captura de tela mostra uma interface de usuário para a edição de dados de um aluno. No topo, há um ícone de seta para voltar e o título 'Alterar aluno'. Abaixo, um avatar genérico e o e-mail 'usuario@gmail.com'. O formulário contém campos para: Nome do aluno (Amanda Batista), Turma (1a-Básico), CPF (666666-66), Telefone (1188768-5095) e Email (Amanda@gmail.com). Um botão azul 'Alterar' está na base.

FONTE: AUTORIA PRÓPRIA, 2025

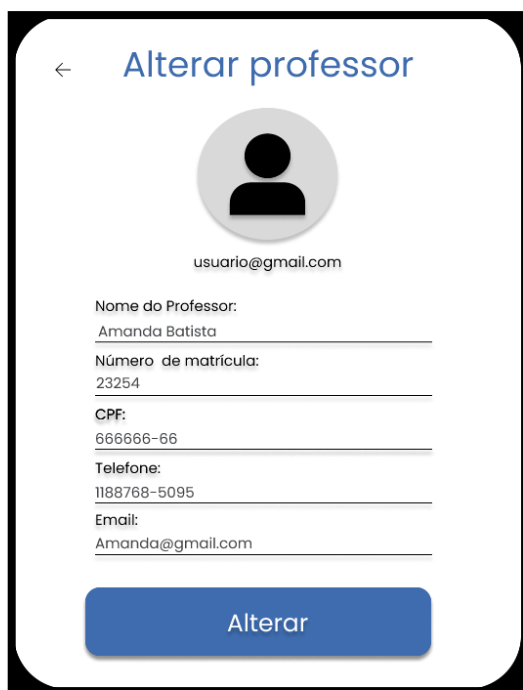
Alterar Aluno: A tela representada na figura 46 apresenta todos os recursos necessários para modificar as informações de um estudante já registrado no sistema, facilitando a gestão dos dados acadêmicos. No topo da interface, o título “Alterar aluno” aparece em destaque, acompanhado de uma seta que possibilita o retorno rápido e intuitivo à tela anterior, garantindo uma navegação eficiente.

Logo abaixo, encontra-se a representação de um avatar genérico junto ao e-mail do usuário, reforçando a identidade do responsável pela alteração. Na área principal, os campos já vêm preenchidos com dados atualizados do aluno, como nome, turma, CPF, telefone e e-mail, permitindo que sejam modificados de forma clara e organizada conforme a necessidade.

O *layout* simples e funcional facilita a compreensão e evita erros durante o processo de edição. Na parte inferior, destaca-se o botão azul “Alterar”, que, ao ser acionado, confirma e registra todas as mudanças realizadas, assegurando a atualização correta dos dados no sistema. Essa tela oferece uma experiência prática e objetiva, essencial para manter as informações dos alunos sempre atuais, colaborando para a organização administrativa e o acompanhamento pedagógico de forma eficiente.

2.3.4.11 – Tela de Alterar Professor

Figura 47 – Tela de alterar professor



← Alterar professor

usuario@gmail.com

Nome do Professor:
Amanda Batista

Número de matrícula:
23254

CPF:
666666-66

Telefone:
1188768-5095

Email:
Amanda@gmail.com

Alterar

FONTE: AUTORIA PRÓPRIA, 2025

Alterar Professor: Ao acessar a tela representada na figura 47 que é dedicada à edição de dados docentes, o usuário encontra uma interface voltada para a atualização das informações cadastrais de um professor já registrado no sistema. No topo da tela, o título “Alterar Professor” é exibido em destaque, acompanhado do avatar do usuário e do *e-mail* (usuario@gmail.com), identificando claramente quem está realizando a ação e proporcionando uma sensação de personalização e segurança.

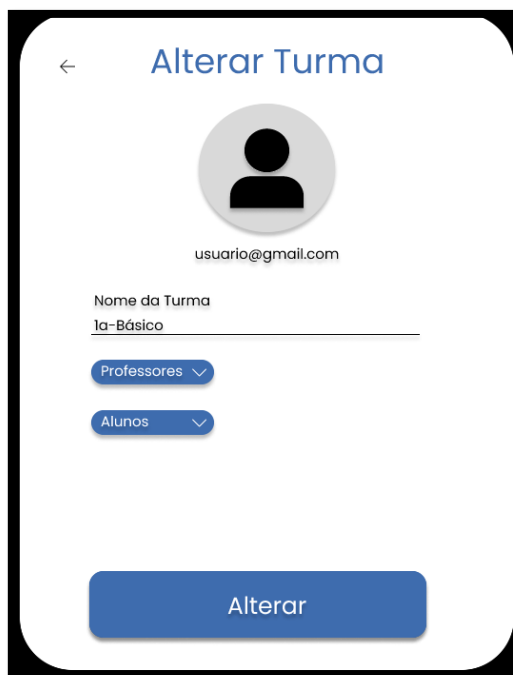
Na área principal, os campos apresentam os dados atuais do professor, como nome completo “Amanda Batista”, número de matrícula “23254”, CPF “666666-66”, telefone “1188768-5095” e *e-mail* “Amanda@gmail.com”. Esses campos são preenchidos automaticamente com as informações existentes, permitindo que sejam modificados conforme necessário. A estrutura organizada facilita a visualização e edição dos dados, contribuindo para um processo eficiente e sem erros.

O botão azul “Alterar”, localizado ao final da tela, confirma e registra mudanças realizadas, garantindo a atualização segura das informações no sistema. Essa funcionalidade é essencial para manter registros docentes precisos, permitindo

ajustes conforme necessário. Além disso, contribui para uma gestão acadêmica eficiente, organizada e alinhada às necessidades administrativas da plataforma.

2.3.4.12 – Tela de Alterar Turma

Figura 48 – Tela de alterar turma



FONTE: AUTORIA PRÓPRIA, 2025

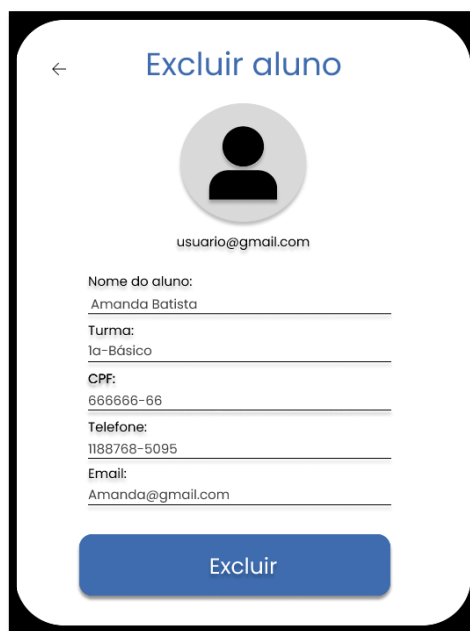
Alterar Turma: A tela representada na figura 48 oferece uma interface clara e funcional para a edição de dados de turmas já cadastradas no sistema acadêmico. No topo, o título “Alterar Turma” aparece em destaque, acompanhado de uma seta que permite ao usuário retornar à tela anterior com facilidade, promovendo uma navegação intuitiva. Logo abaixo, é exibido um avatar padrão seguido do *e-mail* do usuário logado, o que reforça a identificação e segurança durante o processo de edição.

Na área central da tela, encontra-se o campo “Nome da Turma”, já preenchido com o nome atual, como por exemplo “1a-Básico”. Esse campo pode ser modificado conforme necessário, permitindo ajustes em nomenclaturas, correções ou atualizações administrativas. Abaixo desse campo, dois menus suspensos possibilitam a seleção de professores e alunos vinculados à turma. Esses menus oferecem flexibilidade para incluir, excluir ou substituir os participantes da turma, garantindo que os registros reflitam com precisão a composição atual.

Ao final da tela, destaca-se o botão azul “Alterar”, que tem a função de confirmar e registrar todas as mudanças realizadas. Ao ser clicado, ele salva as alterações de forma segura e objetiva, atualizando os dados no sistema. Essa funcionalidade é essencial para manter os registros acadêmicos organizados e atualizados, permitindo que gestores, coordenadores e docentes realizem ajustes com agilidade e confiança. A tela foi desenvolvida para atender às demandas administrativas de forma eficaz, promovendo uma gestão escolar moderna, precisa e alinhada às necessidades institucionais.

2.3.4.13 – Tela de Excluir Aluno

Figura 49 – Tela de excluir aluno



FONTE: AUTORIA PRÓPRIA, 2025

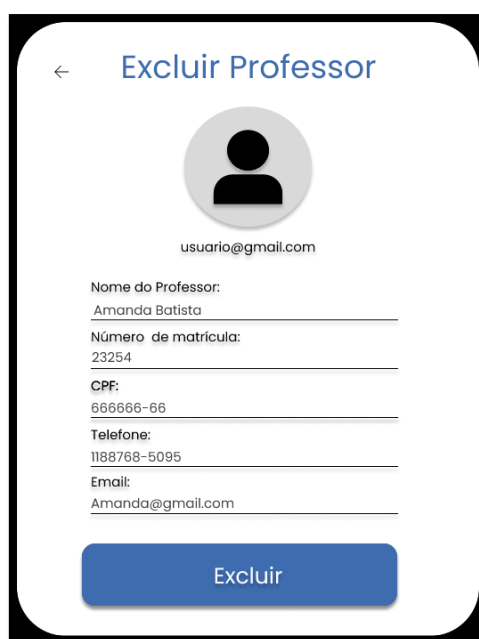
Excluir Aluno: Nesta tela representada na figura 49, o usuário com perfil administrativo acessa a opção de remover um aluno do sistema. O título “Excluir aluno” aparece no topo, deixando claro o objetivo da ação. Logo abaixo, um ícone de perfil genérico acompanha o e-mail “usuario@gmail.com”, identificando o usuário que está realizando a operação.

As informações do aluno são exibidas de forma direta: nome (Amanda Batista), turma vinculada (1a-Básico), CPF (666666-66), telefone (1188768-5095) e e-mail (Amanda@gmail.com). Esses dados ajudam a confirmar que o registro selecionado é o correto antes de seguir com a exclusão. Na parte inferior da tela, o

botão azul “Excluir” finaliza o processo. A interface é simples e objetiva, pensada para agilizar tarefas administrativas sem abrir mão da segurança e da precisão no gerenciamento dos dados escolares.

2.3.4.14 – Tela de Excluir Professor

Figura 50 – Tela de excluir professor



FONTE: AUTORIA PRÓPRIA, 2025

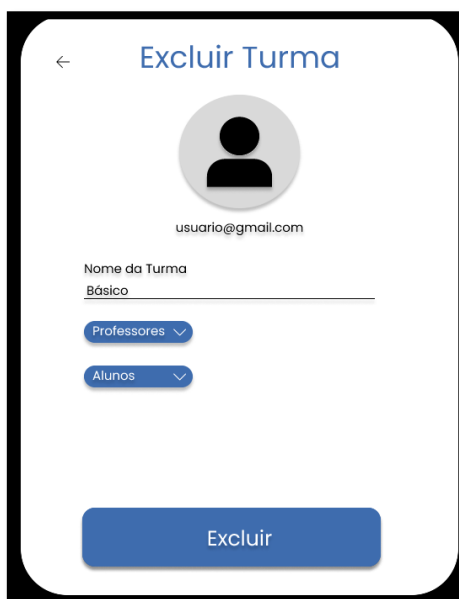
Excluir Professor: Ao acessar essa tela representada na figura 50 o usuário é direcionado a uma tela específica para remoção definitiva de perfis cadastrados no sistema. No topo da interface, o título “Excluir Professor” é exibido de forma destacada, acompanhado por um ícone de seta para retorno à tela anterior. Logo abaixo, o avatar genérico do usuário aparece centralizado, seguido pelo *e-mail* “usuario@gmail.com”, identificando quem está realizando a ação.

A área principal da tela apresenta os dados do professor que será excluído, organizados de forma clara e objetiva. São exibidos o nome completo “Amanda Batista”, o número de matrícula “23254”, o CPF “666666-66”, o telefone “1188768-5095” e o *e-mail* “Amanda@gmail.com”. Esses dados são apresentados em campos fixos, permitindo ao usuário revisar cuidadosamente as informações antes de confirmar a exclusão. Na parte inferior da tela, o botão azul “Excluir” é posicionado com destaque, indicando a ação final. Ao clicar, o sistema realiza a remoção do registro do professor, eliminando-o da base de dados da plataforma.

Essa tela foi projetada para garantir segurança e precisão no processo de exclusão, oferecendo ao usuário uma visualização clara das informações e evitando erros operacionais. A interface mantém a consistência visual da plataforma, com foco na funcionalidade e na experiência administrativa.

2.3.4.15 – Tela de Excluir Turma

Figura 51 - Tela de excluir turma



FONTE: AUTORIA PRÓPRIA, 2025

Excluir Turma: Ao acessar essa interface representada na figura 51, o usuário com perfil de secretaria é direcionado a uma tela específica para a remoção de registros de turmas. O título “Excluir Turma”, destacado no topo da tela, orienta claramente a ação em curso. Logo abaixo, um avatar genérico é exibido junto ao e-mail “usuario@gmail.com”, representando o usuário responsável pela operação.

O painel principal apresenta os dados essenciais da turma selecionada. O campo “Nome da Turma” exibe o título “Básico”, identificando a turma a ser excluída. Abaixo, dois menus suspensos — “Professores” e “Alunos” — indicam a possibilidade de visualizar ou selecionar os docentes e discentes vinculados à turma, oferecendo à equipe administrativa uma visão rápida das relações institucionais envolvidas.

Na parte inferior da tela, um botão azul com o rótulo “Excluir” permite concluir o processo de remoção. Essa interface proporciona uma experiência objetiva e segura para a gestão de turmas, garantindo que ações administrativas sejam realizadas com precisão e responsabilidade.

2.4 – CLIENTE

O cliente, no contexto deste projeto, refere-se ao conjunto de pessoas que possuem interesse em adquirir conhecimentos sobre a Língua Brasileira de Sinais (Libras) por meio de uma ferramenta tecnológica intuitiva, acessível e eficiente. Este público é composto por indivíduos ouvintes e surdos, familiares de pessoas surdas, estudantes, profissionais da educação, profissionais da área da saúde, empresas, instituições e qualquer pessoa que compreenda a importância da inclusão e da comunicação acessível. A definição clara do cliente é essencial para orientar o desenvolvimento das funcionalidades, da interface e do conteúdo do aplicativo, assegurando que ele atenda às demandas reais da sociedade.

2.4.1 – Cliente Interno

O cliente interno corresponde aos membros da própria organização ou equipe responsável pelo desenvolvimento do projeto. Incluem-se nesse grupo os desenvolvedores, programadores, *designers*, gestores de projeto, além de outros colaboradores envolvidos diretamente no processo de concepção, desenvolvimento, manutenção e aprimoramento da plataforma. Este público tem interesse direto no sucesso do projeto, visto que seus resultados refletem não apenas em ganhos profissionais, mas também na consolidação da missão da equipe de promover a acessibilidade e a inclusão social.

2.4.2 – Cliente Externo

O cliente externo é composto por todos os indivíduos, empresas, organizações e instituições que não fazem parte da equipe de desenvolvimento, mas que utilizam ou se beneficiam dos serviços oferecidos pelo aplicativo. Este público busca, principalmente, ferramentas que possibilitem o aprendizado de Libras de forma prática, dinâmica e eficiente. A compreensão das necessidades do cliente externo é essencial para que o aplicativo consiga proporcionar uma experiência satisfatória, contribuindo para o fortalecimento da inclusão de pessoas surdas na sociedade.

2.4.3 – Cliente Ativo

O cliente ativo é caracterizado pelo usuário que faz uso constante e frequente do aplicativo. Este público acessa regularmente os conteúdos, realiza os exercícios, acompanha os módulos e interage com as funcionalidades disponibilizadas. O cliente ativo não apenas consome o serviço, mas também serve como fonte importante de dados para melhorias, por meio de *feedbacks*, avaliações, sugestões e

comportamentos de uso. A manutenção e o crescimento desse grupo são fundamentais para a sustentabilidade do projeto e para seu aprimoramento contínuo.

2.4.4 – Cliente Inativo

O cliente inativo compreende aqueles usuários que, embora tenham realizado o *download* do aplicativo ou criado uma conta, deixaram de utilizá-lo com o passar do tempo, seja por motivos pessoais, por não terem suas expectativas atendidas ou por dificuldades encontradas na navegação ou no conteúdo oferecido. A análise desse grupo permite identificar pontos de melhoria e desenvolver estratégias para reconquistar esses usuários, como atualizações, melhorias na usabilidade, notificações de incentivo, novos conteúdos e campanhas de engajamento.

2.4.5 – Cliente Satisfeito

O cliente satisfeito é aquele que percebe que suas expectativas foram atendidas ou superadas pelo uso do aplicativo. Este perfil reconhece o valor da plataforma na facilitação do aprendizado de Libras, demonstrando apreço pela qualidade do conteúdo, pela facilidade de uso, pela acessibilidade e pela proposta de promover a inclusão. A satisfação do cliente gera impactos positivos como o aumento da taxa de retenção, melhores avaliações nas plataformas de *download*, crescimento orgânico por meio de indicações e fortalecimento da reputação do aplicativo no mercado.

2.4.6 – Cliente Insatisfeito

O cliente insatisfeito, por sua vez, é aquele que encontrou dificuldades ou frustrações no uso do aplicativo. As razões para a insatisfação podem variar desde limitações no conteúdo oferecido, problemas técnicos, lentidão, falta de recursos pedagógicos complementares, dificuldade na navegação, até ausência de suporte adequado. A identificação desse perfil é de extrema relevância para o ciclo de melhoria contínua, possibilitando que a equipe desenvolvedora adote ações corretivas, implemente atualizações necessárias e, conseqüentemente, transforme experiências negativas em oportunidades de evolução do produto.

2.5 – ÁREA DE ATUAÇÃO

A área de atuação representa o setor ou conjunto de atividades em que um projeto, empresa ou instituição desenvolve suas ações, buscando alcançar determinados objetivos. O estudo é relacionado à organização espacial e territorial,

com ênfase em como essas divisões influenciam a administração pública, o planejamento urbano e regional, bem como os impactos socioeconômicos e culturais nas diferentes áreas. A definição de uma área bem estruturada permite uma expansão estratégica, no caso do projeto SCAN HANDS primeiramente terá início em uma microrregião e posteriormente será ampliado para macrorregiões, garantindo sustentabilidade e reconhecimento no setor.

2.5.1 – Microrregião

Foram escolhidas como cidades iniciais para a implementação do projeto Mogi Mirim, Mogi Guaçu e Itapira devido à presença de ETECs que oferecem diversos cursos técnicos e à sua boa infraestrutura educacional. Além disso, há um interesse crescente por tecnologia educacional e inclusão, tornando essas localidades um ambiente propício para o desenvolvimento de iniciativas inovadoras na área da educação.

As unidades do Centro Paula Souza (CPS) na região incluem a ETEC Pedro Ferreira Alves, em Mogi Mirim; a ETEC Euro Albino de Souza, em Mogi Guaçu; e a ETEC João Maria Stevanatto, em Itapira. A meta inicial do projeto é a implantação do SCAN HANDS nessas três ETECs, envolvendo duas turmas por escola, com uma média de 150 a 200 alunos no total (CPS, 2025).

A partir dessa implementação, serão avaliados os resultados pedagógicos e técnicos com base no uso real da tecnologia. A análise considerará o nível de engajamento dos estudantes nas atividades desenvolvidas, a adaptação dos docentes ao recurso tecnológico e os impactos observados no processo de ensino-aprendizagem. Também serão levantados dados relacionados à estabilidade da ferramenta, à frequência de utilização e à experiência de uso por parte da comunidade escolar. Com base nesses indicadores, será possível verificar a eficácia do SCAN HANDS e sua viabilidade para aplicação em larga escala, contribuindo para o fortalecimento de práticas educacionais inovadoras e inclusivas na rede pública de ensino técnico.

2.5.2 – Macrorregião

As próximas regiões-alvo para expansão incluem Campinas, que conta com diversas FATECs e ETECs; Limeira, onde estão a ETEC Trajano Camargo e a FATEC

(Faculdade de Tecnologia do Estado de São Paulo) local; e São João da Boa Vista, que possui ETECs ativas e uma FATEC. A escolha dessas macrorregiões foi baseada em critérios estratégicos, como a forte presença do Centro Paula Souza, o interesse por ferramentas de ensino inclusivas, a infraestrutura de *internet* adequada e a proximidade com a microrregião inicial, garantindo uma transição eficiente na expansão do projeto.

A meta quantitativa inicial do projeto prevê a participação de 9 unidades entre ETECs e FATECs. Em cada escola, serão envolvidas duas turmas, totalizando 18 turmas no conjunto das instituições. Com essa estrutura, estima-se que entre 450 e 600 alunos participem da iniciativa, permitindo uma avaliação ampla dos impactos pedagógicos e técnicos do SCAN HANDS (CPS, 2025).

O cronograma de expansão do projeto segue três fases estratégicas:

- 0 a 6 meses: Fase inicial de implantação e validação em Mogi Mirim, Mogi Guaçu e Itapira, garantindo a eficiência do SCAN HANDS e analisando seus impactos.
- 6 a 12 meses: Expansão para Campinas e Limeira, ampliando a cobertura para novas instituições e aumentando o número de alunos beneficiados.
- 24 meses: Desenvolvimento de uma proposta formal de parceria com a administração central do Centro Paula Souza (CPS), viabilizando a expansão estadual do projeto e sua implementação em novas unidades.

2.6 – CONCORRENTES

Para construir credibilidade no mercado e fortalecer a presença do projeto, é essencial definir uma estratégia de *marketing* bem estruturada. A abordagem inicial visa conectar-se diretamente com instituições de ensino e seus gestores, além de utilizar recursos digitais para ampliar o alcance entre os alunos e professores.

O plano de *marketing* principal será focado em reuniões presenciais com diretores e coordenadores das ETECs, garantindo que os responsáveis pelas instituições compreendam a proposta e possam apoiar sua implementação. Além disso, serão distribuídos cartazes e *folders* físicos nas escolas, contendo QR Codes que direcionam ao site ou aplicativo do projeto, facilitando o acesso às informações e adesão dos estudantes.

As redes sociais segmentadas, como Instagram e Facebook, serão utilizadas para alcançar especificamente os alunos e professores do CPS, criando conteúdos direcionados para esse público. A participação em feiras e eventos educacionais locais, como a Semana da Tecnologia, Fatec Portas Abertas e Fecctec, será uma estratégia importante para ampliar a visibilidade e gerar interação direta com potenciais usuários.

Complementando essas ações, serão produzidos vídeos curtos de demonstração do uso do SCAN HANDS, garantindo que tanto professores quanto alunos possam entender os benefícios do sistema de maneira clara e prática. Essa combinação de *marketing* presencial e digital permitirá que o projeto ganhe força no meio educacional, aumentando sua reputação e a adesão por parte das escolas.

2.6.1 – Diretos

Um concorrente é qualquer pessoa ou entidade que disputa com outra pelo mesmo objetivo. No contexto dos negócios, refere-se à competição entre empresas que atuam no mesmo setor e oferecem produtos ou serviços semelhantes. Essa rivalidade é considerada saudável e essencial para a sustentabilidade do mercado, pois impulsiona as empresas a aumentarem sua eficiência, investirem em inovação e aprimorarem o atendimento ao cliente. Sem a presença da concorrência, haveria menos motivação para que as organizações se diferenciassem e buscassem melhorias constantes. (FONSECA, 2024)

2.6.1.1 – Teams

O Microsoft Teams é uma plataforma de comunicação colaborativa desenvolvida para facilitar a interação remota. Com ele, é possível enviar não somente mensagens privadas, mas criar grupos e canais com diversas pessoas, além de outros recursos. (TECNOBLOG,2024)

Vantagens:

Integração com Microsoft 365: Permite acessar, editar e compartilhar arquivos do Word, Excel, PowerPoint e OneDrive diretamente no Teams, facilitando o trabalho colaborativo. (BAZZARA CARLOS,2025)

Colaboração em tempo real: Múltiplos usuários podem editar o mesmo documento ao mesmo tempo, com atualizações imediatas, o que agiliza as tarefas em equipe. (BAZZARA CARLOS,2025)

Bate-papo e mensagens: Oferece mensagens diretas, em grupo e organizadas por tópicos, facilitando a comunicação e o acompanhamento das conversas. (BAZZARA CARLOS,2025)

Videoconferência integrada: Permite reuniões com vídeo, áudio, gravação, compartilhamento de tela e integração com o Outlook para agendamento automático. (BAZZARA CARLOS,2025)

Organização por canais: Estrutura flexível que agrupa arquivos, tarefas e discussões por tema ou projeto, melhorando a organização do ambiente de trabalho. (BAZZARA CARLOS,2025)

Desvantagens:

Complexidade: O excesso de recursos pode tornar o Teams confuso, especialmente para iniciantes, exigindo treinamento para uso eficiente. (BAZZARA CARLOS,2025)

Uso intensivo de recursos: O consumo elevado de memória e processamento, principalmente em chamadas e ao compartilhar arquivos grandes, pode comprometer o desempenho. (BAZZARA CARLOS,2025)

Sobrecarga de notificações: A grande quantidade de alertas pode gerar distrações e reduzir a produtividade, sendo necessário ajustar as configurações. (BAZZARA CARLOS,2025)

Curva de aprendizado: Usuários novos ou acostumados a outras ferramentas podem ter dificuldades para se adaptar à interface e funcionalidades. (BAZZARA CARLOS,2025)

Colaboração externa: O acesso de convidados e a integração com parceiros externos pode ser complexo, exigindo configurações adicionais. (BAZZARA CARLOS,2025)

Como o SCAN HANDS supera:

O sistema é dedicado exclusivamente à Libras, com reconhecimento corporal via câmera que o Teams não oferece, e agrega um suporte pedagógico completo para escolas, indo além da simples comunicação, com ferramentas pensadas para inclusão.

2.6.1.2 – Hand Talk:

É um aplicativo tradutor automático de Língua de Sinais. Ele funciona como um dicionário de bolso, traduzindo texto e voz automaticamente para Libras (Língua Brasileira de Sinais) e ASL (*American Sign Language* – Língua de Sinais Americana), reunindo tudo o que o cliente precisa em um só lugar: tradutor, trilha de aprendizado e muito mais. Feito para diferentes públicos e objetivos (seja para alguém fluente, estudante curioso, um profissional buscando mais conexões ou uma empresa que deseja promover a inclusão).

Vantagens:

Acessibilidade digital: Traduz textos e áudios automaticamente para Libras e ASL, facilitando a comunicação entre surdos e ouvintes. (PCDEMFOCO,2025)

Interface intuitiva: Fácil de usar, com avatar 3D (Hugo) que realiza as traduções de forma visual e didática. (PCDEMFOCO,2025)

Gratuito e multiplataforma: Disponível para Android, iOS e como *plugin* para *sites*, ampliando o alcance da inclusão. (PCDEMFOCO,2025)

Reconhecimento internacional: Premiado por sua inovação em acessibilidade e impacto social. (PCDEMFOCO,2025)

Uso educacional e profissional: Pode ser usado em salas de aula, entrevistas de emprego e ambientes corporativos para promover inclusão. (PCDEMFOCO,2025)

Desvantagens:

Tradução limitada: A conversão é feita para o português sinalizado, o que pode não respeitar totalmente a gramática da Libras, dificultando a compreensão por usuários surdos nativos. (SINGLIBRAS,2025)

Dependência de *internet*: Para funcionar corretamente, o aplicativo precisa estar conectado, o que pode limitar o uso em locais com baixa conectividade. (SINGLIBRAS,2025)

Não substitui o ensino formal: Apesar de útil como apoio, não é indicado como ferramenta principal para aprender Libras de forma aprofundada. (SINGLIBRAS,2025)

Português sinalizado em vez de Libras autêntica: O aplicativo traduz para uma forma adaptada da língua de sinais, o que pode dificultar a compreensão por surdos nativos que usam a gramática completa da Libras. (SINGLIBRAS,2025)

Limitações no vocabulário: Apesar de amplo, o banco de sinais ainda não cobre todas as expressões e regionalismos da Libras, o que pode limitar a comunicação em situações mais complexas. (SINGLIBRAS,2025)

Como o SCAN HANDS supera:

O projeto é único ao integrar o reconhecimento corporal em Libras pelo uso da câmera, tornando a interação direta e natural. Além disso, o sistema entrega suporte para escolas, com ferramentas e *chatbot*, criando um ambiente educativo e inclusivo que HandTalk ainda não oferece.

2.6.1.3 – NSA:

O NSA (Novo Sistema Acadêmico) é um sistema acadêmico desenvolvido para facilitar o trabalho da área acadêmica das Etecs do Centro Paula Souza. Através do NSA, o responsável e os alunos da ETESP (Escola Técnica Estadual de São Paulo) têm acesso às menções, frequências, plano de trabalho docente (PTD – Plano de Trabalho Docente), ficha individual do aluno e ao horário escolar. Além disso, o sistema também oferece ferramentas para o acompanhamento do desempenho acadêmico e possibilita uma comunicação mais eficiente entre docentes, alunos e responsáveis. Dessa forma, busca-se otimizar a gestão e a organização dentro das instituições de ensino técnico. (NORMASABNT, 2025)

Vantagens:

Acesso rápido às informações escolares: Alunos, professores e responsáveis podem consultar notas, faltas, horários e planos de aula em um só lugar. (NORMASABNT,2025)

Transparência e comunicação: Facilita o acompanhamento do desempenho acadêmico e melhora a comunicação entre escola e comunidade escolar. (NORMASABNT,2025)

Plataforma *online* e centralizada: Pode ser acessada de qualquer lugar com *internet*, agilizando processos que antes dependiam da secretaria presencial. (NORMASABNT,2025)

Desvantagens:

Dependência de *internet*: O acesso ao sistema exige conexão estável, o que pode ser um obstáculo para alguns alunos. (NORMASABNT,2025)

Interface pouco intuitiva: Alguns usuários relatam dificuldades de navegação, especialmente em dispositivos móveis. (NORMASABNT,2025)

Limitações técnicas: Em períodos de alta demanda (como fechamento de notas), o sistema pode apresentar lentidão ou instabilidade. (NORMASABNT,2025)

Como o SCAN HANDS supera:

O projeto é pioneiro na união dessas duas frentes — reconhecimento de Libras via movimentos capturados em tempo real e apoio educacional, com *chatbot* interativo que facilita o aprendizado e comunicação.

2.6.2 – Indiretos

Um concorrente indireto é uma empresa que, mesmo não oferecendo exatamente os mesmos produtos ou serviços, atua sobre o mesmo público-alvo ao disponibilizar alternativas que podem substituir o que outra empresa oferece. Esse tipo de concorrência se dá quando diferentes empresas disputam a atenção do consumidor ao suprir necessidades parecidas, ainda que com soluções distintas. Nesse cenário, cada uma busca conquistar espaço no mercado por meio de estratégias que apresentem suas ofertas como opções viáveis e atrativas para as mesmas demandas atendidas pelos concorrentes diretos. (MOREIRA, 2025)

2.6.2.1 – VLibras

O VLibras é um conjunto de ferramentas específicas para traduzir textos em português para a Língua Brasileira de Sinais. O objetivo da ferramenta é tornar

computadores, celulares e navegadores mais acessíveis para todas as pessoas. (UNIFIQUE,2025)

Vantagens:

Gratuito e de código aberto: Qualquer pessoa pode usar e até contribuir com melhorias. (GOVERNODIGITAL,2025)

Multiplataforma: Funciona em computadores, celulares e navegadores, com versões para Windows, Linux, Android e iOS. (GOVERNODIGITAL,2025)

Inclusão digital: Traduz textos, áudios e vídeos do português para Libras, promovendo acessibilidade para pessoas surdas. (GOVERNODIGITAL,2025)

Avatar 3D interativo: Um intérprete virtual realiza os sinais, facilitando a compreensão visual. (GOVERNODIGITAL,2025)

Ampla adoção: Já é utilizado em mais de 120 mil sites e realiza mais de 100 mil traduções por dia. (GOVERNODIGITAL,2025)

Desvantagens:

Não substitui intérpretes humanos: O VLibras não deve ser usado em situações complexas como aulas, audiências ou eventos ao vivo. (GOVERNODIGITAL,2025)

Tradução automática com limitações: Pode haver dificuldades com expressões idiomáticas ou frases muito longas, o que afeta a fluidez da Libras. (GOVERNODIGITAL,2025)

Como o SCAN HANDS supera:

O projeto oferece captura e reconhecimento em tempo real dos sinais feitos pelo usuário através da câmera, tornando a interação mais dinâmica e natural, além de suportar a escola com ferramentas e *chatbot* integrados, ampliando o uso pedagógico.

2.6.2.2 – ProDeaf

ProDeaf é um aplicativo para Android e iOS que traduz frases de português para Libras (Língua Brasileira de Sinais), a língua usada pelos surdos. A proposta do *app* é ajudar na comunicação entre as pessoas e fazer com que os usuários conheçam um pouco mais sobre essa linguagem. Por meio do reconhecimento de voz, o serviço consegue, com ajuda de um avatar, transformar a fala em sinais.

Vantagens:

Reconhecimento de voz e texto: O *app* converte fala e escrita em Libras por meio de um avatar 3D, facilitando a comunicação com pessoas surdas. (TECHTUDO,2025)

Gratuito e acessível: Está disponível para dispositivos Android e pode ser usado também no computador. (TECHTUDO,2025)

Dicionário integrado: Possui um glossário organizado por categorias, o que ajuda no aprendizado da Libras. (TECHTUDO,2025)

Interface intuitiva: O avatar torna a experiência mais visual e compreensível, especialmente para iniciantes. (TECHTUDO,2025)

Desvantagens:

Limitações no reconhecimento de voz: Nem sempre a fala é interpretada corretamente, o que pode comprometer a tradução. (TECHTUDO,2025)

Disponibilidade restrita: Em algumas versões, o *app* estava disponível apenas para Android, o que limitava o acesso para usuários de outras plataformas. (TECHTUDO,2025)

Como o SCAN HANDS supera:

A solução alia reconhecimento corporal via câmera para identificar sinais em Libras, o que permite uma comunicação mais natural e eficiente. Além disso, o suporte para escolas integrado ao projeto ajuda na inclusão educacional, além de contar com um *chatbot* que interage e auxilia alunos e professores, algo que ProfDeaf não oferece. Com essa combinação de recursos, o sistema promove acessibilidade ativa e fortalece o vínculo entre tecnologia e educação inclusiva. Sua aplicação prática

contribui diretamente para a autonomia da comunidade surda em ambientes escolares.

2.6.2.3 – EduLibras

EduLibras é um aplicativo bilíngue que visa ensinar tanto a Língua Brasileira de Sinais (Libras) quanto o português escrito, com foco na interação individual ou em grupo e no desenvolvimento da leitura e escrita dos usuários. Ele oferece conteúdo interativo para ampliar o domínio desses idiomas, permitindo que pessoas surdas e ouvintes se comuniquem e interajam mais efetivamente. (EDULIBRAS,2025)

Vantagens:

Equipe especializada: A EduLibras é formada por profissionais certificados pelo MEC (Ministério da Educação), o que garante qualidade técnica nos serviços. (EDULIBRAS,2025)

Atuação nacional: Oferece serviços em todo o Brasil, incluindo empresas, escolas, eventos e órgãos públicos. (EDULIBRAS,2025)

Diversidade de serviços: Vai além da tradução, oferecendo cursos de Libras, português para surdos, audiodescrição, intérpretes e guia-intérpretes. (EDULIBRAS,2025)

Compromisso com a inclusão: A missão da empresa é promover o acesso à informação para pessoas com deficiência auditiva, valorizando suas competências e experiências. (EDULIBRAS,2025)

Desvantagens:

Foco institucional: A maior parte dos serviços parece voltada para empresas e instituições, o que pode limitar o acesso individual gratuito. (EDULIBRAS,2025)

Ausência de *app* ou ferramenta automatizada: Diferente de VLibras ou ProDeaf, a EduLibras não oferece um aplicativo de tradução automática, sendo mais voltada à prestação de serviços humanos. (EDULIBRAS,2025)

Como o SCAN HANDS supera:

O projeto utiliza a câmera para capturar os movimentos corporais em tempo real, permitindo um reconhecimento dinâmico dos sinais em Libras, tornando o

aprendizado mais prático e interativo. Além disso, o sistema oferece suporte específico para o ambiente escolar, facilitando a inclusão e comunicação dentro da escola, algo que o EduLibras não entrega.

2.6.3 – Conclusão

Para garantir a inserção efetiva do SCAN HANDS no mercado educacional e fortalecer sua credibilidade, foi desenvolvida uma estratégia de *marketing* que une ações presenciais, como reuniões com gestores das ETECs e distribuição de materiais físicos com QR Codes, a uma forte presença digital em redes sociais e participação em eventos como a Semana da Tecnologia. O diferencial do projeto está no uso da câmera para reconhecimento corporal em Libras, oferecendo uma experiência interativa e acessível, voltada especialmente ao ambiente escolar.

Durante a análise de concorrência, foi possível identificar que, embora existam soluções relevantes como Microsoft Teams, HandTalk, NSA, VLibras, ProDeaf e EduLibras, nenhuma delas combina reconhecimento em tempo real por câmera, suporte pedagógico e foco exclusivo na inclusão educacional como o SCAN HANDS. Enquanto os concorrentes diretos se limitam à tradução ou comunicação, o nosso projeto vai além, proporcionando aprendizado prático da Libras com o auxílio de *chatbot* e ferramentas integradas para professores e alunos. Assim, o SCAN HANDS se destaca como uma solução inovadora, acessível e alinhada

2.7 – LEIS E REGRAS CONTRATUAIS

No Brasil, os contratos são regulamentados por leis que definem os direitos e deveres das partes envolvidas. A principal base legal é o Código Civil, que trata da formação, validade, execução e extinção dos contratos, além de estabelecer princípios como boa-fé, autonomia da vontade e função social. Em relações de consumo, o Código de Defesa do Consumidor (CDC) também se aplica, garantindo transparência, proteção contra abusos e equilíbrio nas relações entre fornecedores e consumidores. Além dessas normas gerais, há leis específicas para diferentes áreas, como a CLT (Consolidação das Leis do Trabalho), que regula contratos trabalhistas. A jurisprudência — decisões dos tribunais — também orienta a interpretação e aplicação dessas leis, e os princípios jurídicos (como equidade e segurança jurídica) complementam esse processo. (SARAIVAEDUCACAO, 2023)

2.7.1 – Termos e Condições de Uso do Sistema SCAN HANDS

Para garantir a conformidade legal e a segurança jurídica na utilização do sistema SCAN HANDS, é fundamental observar as legislações aplicáveis em âmbito federal, estadual e municipal. A seguir, apresentam-se as principais normas que regem os direitos e deveres da empresa desenvolvedora, da instituição contratante e dos usuários do sistema.

Legislação Federal

Lei nº 13.709/2018 – Lei Geral de Proteção de Dados (LGPD)

Trecho: “Dados pessoais são quaisquer informações relacionadas a pessoa natural identificada ou identificável” (BRASIL, 2018).

A Lei Geral de Proteção de Dados (LGPD) assegura direitos aos titulares de dados e impõe responsabilidades aos responsáveis pelo tratamento dessas informações. No projeto SCAN HANDS, que faz uso de câmeras e imagens biométricas, a lei exige o consentimento explícito dos usuários, uma política de privacidade clara, medidas de segurança (como criptografia) e um canal de atendimento sobre dados. O descumprimento pode gerar multas de até 2% do faturamento, limitadas a R\$ 50 milhões. a Lei nº 13.709/2018, que estabelece diretrizes para proteger os direitos dos titulares desses dados.

Lei nº 13.146/2015 – Lei de Estatuto da Pessoa com Deficiência (LBI)

Trecho: “A pessoa com deficiência tem direito à igualdade de oportunidades com as demais pessoas e não sofrerá nenhuma espécie de discriminação” (BRASIL, 2015).

A LBI assegura direitos fundamentais à acessibilidade comunicacional, educacional e tecnológica, promovendo a inclusão plena de pessoas com deficiência. O SCAN HANDS, ao empregar reconhecimento por gestos, torna-se uma ferramenta poderosa de inclusão digital, especialmente para indivíduos com deficiência auditiva ou motora, ampliando sua autonomia e participação social. Além disso, o projeto possui potencial para captar recursos por meio de editais públicos, incentivos fiscais e parcerias estratégicas com instituições governamentais, fortalecendo sua implementação em larga escala. Sua adoção pode transformar ambientes educacionais e sociais, tornando-os mais acessíveis, interativos e inclusivos.

Lei nº 9.394/1996 – Lei de Diretrizes e Bases da Educação (LDB)

Trecho: “A educação, direito de todos, será promovida e incentivada com a colaboração da família, da comunidade, da sociedade em geral e do poder público” (BRASIL, 1996).

A LDB regulamenta a educação nacional e incentiva o uso de tecnologias assistivas que promovam a inclusão de alunos com deficiência nos processos de ensino e aprendizagem. No contexto do SCAN HANDS, a utilização de reconhecimento de gestos contribui para a acessibilidade comunicacional, favorecendo a participação ativa de estudantes com deficiência auditiva, motora ou de fala. Dessa forma, o projeto está alinhado aos princípios legais de uma educação inclusiva, equitativa e de qualidade.

Decreto nº 5.626/2005

Trecho: Estabelece a Política Nacional de Proteção dos Direitos da Pessoa com Deficiência e cria um comitê para promover acessibilidade” (BRASIL, 2005).

Esse decreto regulamenta o uso e o ensino da Língua Brasileira de Sinais (Libras) nas instituições de ensino, promovendo a inclusão de pessoas surdas. O projeto SCAN HANDS, ao interpretar gestos e apoiar a comunicação em Libras, se alinha diretamente à legislação, podendo ser aplicado como ferramenta de apoio à educação inclusiva e à comunicação acessível.

Lei nº 12.965/2014 – Marco Civil da *Internet*

Trecho: “Define princípios e direitos para o uso da *internet*, garantindo liberdade de expressão e proteção da privacidade” (BRASIL, 2014).

O Marco Civil da *Internet* estabelece os princípios, garantias e deveres para o uso da internet no Brasil, com ênfase na liberdade de expressão, proteção à privacidade e à segurança dos dados pessoais. No contexto do SCAN HANDS, que opera em ambiente digital e lida com informações sensíveis (como imagens e comportamentos), a lei exige que o sistema adote medidas que assegurem a confidencialidade, integridade e proteção dos dados dos usuários, reforçando a necessidade de transparência e responsabilidade no uso da tecnologia. Essa adequação legal é essencial para garantir a confiança dos usuários e viabilizar parcerias estratégicas.

Lei nº 10.436/2002

Trecho: “Reconhece a Língua Brasileira de Sinais (Libras) como meio legal de comunicação e expressão das pessoas surdas” (BRASIL, 2002).

Essa lei reconhece oficialmente a Língua Brasileira de Sinais (Libras) como meio legítimo de comunicação e expressão da comunidade surda no Brasil. No âmbito do SCAN HANDS, essa legislação fortalece o valor social e educacional do sistema, que pode atuar como ponte comunicacional acessível ao interpretar ou facilitar a interação por meio de gestos. Assim, o projeto contribui para a promoção da inclusão e da cidadania linguística, em consonância com os direitos reconhecidos por essa lei.

Decreto nº 9.854/2019 – Estratégia Brasileira de Governo Digital

Trecho: “Estabelece diretrizes para a transformação digital no setor público, visando simplificar serviços e ampliar o acesso dos cidadãos” (BRASIL, 2019).

Esse decreto institui a Estratégia Brasileira de Governo Digital, que orienta a transformação dos serviços públicos para o formato digital, acessível e inclusivo. No contexto do SCAN HANDS, a norma incentiva parcerias com soluções tecnológicas acessíveis, especialmente aquelas que promovem a inclusão de pessoas com deficiência por meio da inovação. Assim, o projeto se posiciona como uma ferramenta estratégica alinhada aos objetivos do governo digital e à modernização do atendimento público.

Lei nº 13.005/2014 – Plano Nacional de Educação (PNE)

Trecho: “Define as diretrizes, metas e estratégias para a política educacional no Brasil, com vigência de dez anos” (BRASIL, 2014).

A Meta 4 do PNE prevê a universalização do acesso à educação básica para pessoas com deficiência, com garantia de atendimento educacional especializado e inclusão. O SCAN HANDS, ao utilizar tecnologias de leitura de gestos e comunicação acessível, contribui diretamente para romper barreiras comunicacionais e promover a participação de alunos com deficiência, especialmente aqueles com surdez ou dificuldades motoras. Isso o torna um projeto compatível com os objetivos do PNE, podendo ser inclusive elegível para financiamentos ou parcerias com base nos recursos previstos na lei.

Lei nº 12.764/2012 – Política Nacional de Proteção dos Direitos da Pessoa com Transtorno do Espectro Autista

Trecho: “Institui a política nacional para garantir direitos e inclusão das pessoas com transtorno do espectro autista” (BRASIL, 2012).

A lei garante o direito das pessoas com Transtorno do Espectro Autista (TEA) ao acesso a tecnologias de comunicação alternativa, promovendo inclusão e autonomia. O SCAN HANDS pode futuramente ser adaptado para atender esse público, ampliando seu impacto social como recurso acessível também para autistas com dificuldades de fala ou interação verbal.

Lei nº 9.279/1996 – Lei da Propriedade Industrial

Trecho: “Regula direitos e obrigações relativos à propriedade industrial, incluindo patentes, marcas e registros industriais” (BRASIL, 1996).

Essa lei trata da proteção de ativos intelectuais, como marcas, patentes e desenhos industriais. No contexto do SCAN HANDS, ela possibilita o registro da marca do sistema, bem como a proteção das inovações tecnológicas e funcionais envolvidas (como algoritmos, mecanismos de leitura gestual ou *design* exclusivo). Esse registro impede cópias não autorizadas, confere segurança jurídica ao projeto e pode valorizar a proposta perante investidores, editais e parceiros institucionais.

Lei nº 9.610/1998 – Lei de Direitos Autorais

Trecho: “Protege os direitos dos autores sobre suas obras literárias, artísticas e científicas” (BRASIL, 1998).

Essa lei protege os elementos intelectuais de um *software*, como o código-fonte, *design* da interface, estrutura de banco de dados e identidade visual. No caso do SCAN HANDS, ela permite resguardar juridicamente essas criações contra uso indevido ou cópias não autorizadas, garantindo direitos exclusivos de uso e reprodução ao autor do sistema.

Lei nº 8.078/1990 – Código de Defesa do Consumidor

Trecho: “Estabelece normas de proteção e defesa do consumidor, garantindo seus direitos nas relações de consumo” (BRASIL, 1990).

Caso o SCAN HANDS seja ofertado como produto ou serviço, o CDC exige transparência nas informações, segurança na utilização e suporte adequado ao usuário. A legislação garante os direitos do consumidor frente a falhas técnicas, informações insuficientes ou mau funcionamento, contribuindo para fortalecer a confiança no projeto no ambiente comercial.

Lei nº 14.133/2021 – Nova Lei de Licitações e Contratos Administrativos

Trecho: “Regula licitações e contratos da administração pública visando transparência e eficiência” (BRASIL, 2021).

Essa lei regula as licitações e contratações públicas no Brasil. Se o SCAN HANDS for ofertado a instituições governamentais, será necessário seguir seus procedimentos, garantindo transparência, concorrência justa, responsabilidade contratual e cumprimento de requisitos legais para fornecimento de bens e serviços à administração pública.

LEIS ESTADUAIS (São Paulo)

Lei nº 12.907/2008

Trecho: “Dispõe sobre a obrigatoriedade do ensino da Língua Brasileira de Sinais (Libras) e dá outras providências” (BRASIL, 2008).

Essa lei exige que órgãos públicos do Estado de São Paulo garantam acessibilidade comunicacional para pessoas com deficiência. O SCAN HANDS, ao facilitar a comunicação por gestos, pode ser adotado em escolas estaduais e serviços públicos, promovendo inclusão e acesso igualitário à informação.

Lei nº 17.469/2021

Trecho: “Altera dispositivos referentes à administração do sistema prisional e segurança pública no Estado de São Paulo” (BRASIL, 2021).

Essa lei complementa a LGPD (Lei Geral de Proteção de Dados) no Estado de São Paulo e define regras para o tratamento de dados pessoais por órgãos públicos estaduais. Caso o SCAN HANDS seja implementado em instituições públicas paulistas, ele deve assegurar o respeito à privacidade, à segurança da informação e à transparência, fortalecendo sua legalidade e credibilidade no setor público.

LEIS MUNICIPAIS (Mogi Guaçu)

Lei Municipal nº 5.095/2015 – Plano Municipal de Educação

Trecho: “Estabelece metas e diretrizes para a melhoria da educação municipal, promovendo inclusão e qualidade” (ITAPIRA, 2015).

Essa legislação municipal tem como foco a inclusão de estudantes com deficiência e a promoção de tecnologias educacionais acessíveis. O SCAN HANDS, ao facilitar a comunicação por meio de gestos, contribui para reduzir barreiras de aprendizagem e ampliar a participação de alunos na rede pública, estando alinhado às metas do PME (Pequenas e Médias Empresas) de Itapira.

Leis Municipais sobre Acessibilidade Urbana

Trecho: “Leis municipais de acessibilidade garantem a eliminação de barreiras para pessoas com deficiência em espaços públicos” (BRASIL, 2025)

Essas leis definem padrões que garantem acessibilidade nos serviços públicos e espaços municipais, exigindo que soluções tecnológicas utilizadas em escolas ou repartições públicas estejam adequadas às necessidades de pessoas com deficiência. O SCAN HANDS, ao facilitar a comunicação acessível, pode ser incorporado nesses ambientes como ferramenta de inclusão, desde que respeite os requisitos técnicos e funcionais estabelecidos pelas normas locais.

Lei Orgânica do Município de Mogi Guaçu

Trecho: “Dispõe sobre a autonomia política, administrativa e financeira do município, regulando sua organização e funcionamento” (MOGI GUAÇU, 2025)

Essa lei segue os princípios constitucionais de inclusão, igualdade e acessibilidade, servindo como base legal para o apoio institucional a iniciativas voltadas à cidadania e à justiça social. O SCAN HANDS, ao promover acessibilidade comunicacional para pessoas com deficiência, está alinhado aos valores da Lei Orgânica, podendo ser adotado ou apoiado pelo poder público local como instrumento de inclusão. Sua aplicação fortalece políticas públicas voltadas à equidade e contribui diretamente para a construção de ambientes mais justos e participativos. Além disso, reforça o compromisso das instituições com a democratização da tecnologia e o respeito à diversidade.

NORMAS TÉCNICAS E PADRÕES DE ACESSIBILIDADE

ABNT NBR 9050:2020:

Trecho “Define critérios para garantir acessibilidade em edificações e espaços urbanos para pessoas com deficiência ou mobilidade reduzida” (ABNT, 2020).

Acessibilidade em Espaços Físicos Essa norma estabelece critérios técnicos para garantir a acessibilidade em edificações, mobiliário, espaços e equipamentos urbanos. Embora voltada ao ambiente físico, ela serve como base cultural e conceitual para projetos inclusivos, reforçando o compromisso com a acessibilidade universal e inspirando soluções digitais que respeitem os princípios de usabilidade e autonomia.

ABNT NBR 17060:2022:

Trecho: “Estabelece requisitos para garantir a acessibilidade de aplicativos móveis, facilitando o uso por pessoas com diferentes tipos de deficiência” (ABNT, 2022).

Acessibilidade na Comunicação Digital Norma brasileira que define requisitos de acessibilidade para aplicativos móveis e interfaces digitais, incluindo *apps* nativos, híbridos e *web*. Aplica-se diretamente ao SCAN HANDS, exigindo critérios como contraste adequado, navegação por teclado, uso de Libras, legendas e descrição de elementos não textuais. É a primeira norma nacional voltada à acessibilidade digital e fortalece a conformidade com a Lei Brasileira de Inclusão.

WCAG – *Web Content Accessibility Guidelines*:

Trecho: “Diretrizes que orientam a criação de conteúdos web acessíveis para pessoas com deficiência” (W3C, 2025)

Conjunto de diretrizes internacionais desenvolvidas pelo W3C para tornar conteúdos digitais acessíveis a pessoas com deficiência. Baseadas nos princípios Perceptível, Operável, Compreensível e Robusto (POUR), as WCAG (*Web Content Accessibility Guidelines* – Diretrizes de Acessibilidade para Conteúdo da Web) são amplamente utilizadas como referência em editais públicos, certificações e políticas de acessibilidade digital. Seguir essas diretrizes amplia o alcance e a inclusão do SCAN HANDS em ambientes educacionais, institucionais e governamentais

2.7.2 – Contrato de Prestação de Serviços

Neste ato, apresenta-se formalmente o Contrato de Prestação de Serviços, elaborado em conformidade com os dispositivos do Código Civil Brasileiro, que representa o principal referencial normativo aplicável aos contratos de natureza empresarial.

O presente instrumento foi cuidadosamente redigido com base nas legislações vigentes, assegurando o pleno respeito aos direitos e deveres das partes contratantes. Ressalta-se que o Código Civil oferece o fundamento jurídico essencial para a validade e execução deste contrato, conferindo segurança, equilíbrio e respaldo legal às obrigações aqui estipuladas.

Dessa forma, as Partes declaram estar cientes e de acordo com todas as cláusulas ora pactuadas, comprometendo-se a cumpri-las integralmente, em observância ao ordenamento jurídico brasileiro, garantindo a eficácia plena e legítima deste instrumento contratual.

CONTRATO DE PRESTAÇÃO DE SERVIÇOS DE SOFTWARE – SCAN HANDS

Entre:

Contratante: ETEC de Mogi Guaçu, com sede na Rua [Endereço completo], Mogi Guaçu – SP, inscrita no CNPJ sob nº [inserir], doravante denominada “Contratante”;

Contratada: Isabela Leme Cruz, representante legal do projeto SCAN HANDS, residente em [Endereço completo], inscrita no CPF sob nº [inserir], doravante denominada “Contratada”;

Cláusula 1 – Objeto do Contrato

1.1. O presente contrato tem por objeto a prestação de serviços pela Contratada, responsável pelo projeto SCAN HANDS, para o fornecimento, implantação, manutenção, suporte técnico e eventual adequação pedagógica de um sistema de tradução gestual com base na Língua Brasileira de Sinais (LIBRAS), visando à inclusão comunicacional e educacional de pessoas surdas.

1.2. O sistema será entregue conforme as especificações técnicas descritas no Anexo I.

1.3. Esta prestação de serviços fundamenta-se nas seguintes normas e legislações:

- Lei nº 9.394/1996 – LDB, que regulamenta a educação nacional e incentiva inclusão (BRASIL, 1996);
- Lei nº 13.146/2015 – LBI, garantindo acessibilidade e igualdade de oportunidades (BRASIL, 2015);
- Decreto nº 5.626/2005, que regulamenta o uso da Libras em instituições de ensino (BRASIL, 2005);
- Lei nº 10.436/2002, reconhecendo Libras como meio legal de comunicação (BRASIL, 2002);
- Lei nº 12.907/2008, que exige o ensino da Libras em órgãos públicos (BRASIL, 2008);
- Plano Nacional de Educação – PNE (Lei nº 13.005/2014) e Plano Municipal de Educação – PME (Lei Municipal nº 5.095/2015), para inclusão de alunos com deficiência (BRASIL, 2014; ITAPIRA, 2015);
- ABNT NBR 9050:2020, ABNT NBR 17060:2022 e WCAG – *Web Content Accessibility Guidelines*, garantindo acessibilidade em espaços físicos e digitais (ABNT, 2020; ABNT, 2022; W3C,2025).

Cláusula 2 – Representação e Equipe Responsável

2.1. A Contratada atua como representante oficial do projeto SCAN HANDS, desenvolvido em colaboração com os seguintes integrantes:

- Giovanna Gabrielle Cirino dos Santos
- Isabela Leme Cruz
- Kimberly Moura Batista
- Laura Gonçalves Flausino
- Mayra Vitória Carlos

- Miguel Martins Maia
- Náthali Pasquim Ribeiro
- Patrick Heloi da Silva Júnior

2.2. A Contratada possui plenos poderes para firmar este contrato e tomar as decisões necessárias à sua execução.

Cláusula 3 – Tratamento de Dados Pessoais

3.1. A Contratada se compromete a coletar, tratar e armazenar dados pessoais dos usuários do SCAN HANDS em conformidade com a Lei nº 13.709/2018 – LGPD (BRASIL, 2018) e Lei nº 17.469/2021 (SP,2021), garantindo segurança, transparência e respeito à privacidade.

Trecho: "Os agentes de tratamento devem adotar medidas de segurança, técnicas e administrativas aptas a proteger os dados pessoais de acessos não autorizados e de situações acidentais ou ilícitas de destruição, perda, alteração, comunicação ou difusão." (BRASIL, 2018)

3.2. Os dados incluem imagem captada por câmera, gestos corporais, *logs* de acesso e, se necessário, nome e *e-mail* para cadastro e suporte, jamais sendo utilizados para fins comerciais.

Cláusula 4 – Finalidade do Tratamento de Dados

4.1. Os dados coletados terão como finalidades:

- Reconhecimento de sinais em LIBRAS;
- Promoção da inclusão digital e educacional;
- Feedback* de aprendizagem ao usuário;
- Melhorias técnicas do sistema com base em análises.

4.2. 4.2. Todas as atividades de tratamento de dados obedecem à LGPD, ao Código de Defesa do Consumidor – Lei nº 8.078/1990 (BRASIL, 1990) e às normas de segurança da informação (ABNT NBR 17060:2022, WCAG). O sistema adota práticas que garantem a confidencialidade, integridade e disponibilidade das informações, respeitando os direitos dos usuários. Essa conformidade reforça o

compromisso com a proteção de dados e a construção de um ambiente digital ético, seguro e acessível.

Cláusula 5 – Direitos do Usuário

5.1. Conforme previsto na LGPD, o usuário tem direito de:

- Confirmar existência de tratamento;
- Acessar, corrigir ou excluir seus dados;
- Revogar o consentimento;
- Solicitar portabilidade ou anonimização;
- Obter informações sobre uso e compartilhamento.

5.2. As solicitações deverão ser enviadas para: [inserir *e-mail* de contato].

Cláusula 6 – Segurança e Armazenamento

6.1. Os dados serão armazenados em ambiente seguro, com criptografia, controle de acesso e monitoramento.

6.2. O período de retenção estará vinculado à necessidade técnica do projeto, conforme previsto na legislação.

6.3. As práticas de segurança também seguem recomendações da ABNT NBR 17060:2022 e princípios das WCAG, garantindo proteção integral aos usuários.

Cláusula 7 – Obrigações da Contratada

7.1. Fornecer o sistema conforme especificações técnicas, obedecendo às leis de inclusão, acessibilidade e proteção de dados:

- LGPD, LBI, LDB, PNE, PME, Decreto nº 5.626/2005;
- ABNT NBR 9050:2020, ABNT NBR 17060:2022, WCAG;
- Lei nº 9.610/1998 – Direitos Autorais;
- Lei nº 9.279/1996 – Propriedade Industrial.

7.2. Garantir conformidade com LGPD, LBI, CDC, PME, e demais normas educacionais e de acessibilidade;

7.3. Manter confidencialidade das informações;

7.4. Prover suporte técnico e atualizações conforme o Anexo II;

7.5. Implementar medidas de cibersegurança e comunicar falhas ou incidentes.

Cláusula 8 – Obrigações da Contratante

8.1. Utilizar o sistema apenas para fins educacionais e institucionais;

8.2. Proteger credenciais de acesso;

8.3. Fornecer dados e *feedbacks* relevantes para adequação do sistema;

8.4. Zelar pela ética e segurança no uso da plataforma.

Cláusula 9 – Propriedade Intelectual

9.1. Todo código, interface, banco de dados e documentação do SCAN HANDS é de titularidade da equipe desenvolvedora.

9.2. A Contratante recebe licença de uso não exclusiva, limitada à vigência contratual.

9.3. Esta cláusula é respaldada por: Lei de Direitos Autorais – Lei nº 9.610/1998 e Lei da Propriedade Industrial – Lei nº 9.279/1996.

Cláusula 10 – Confidencialidade

10.1. Ambas as partes se comprometem a manter sigilo sobre dados, metodologias e demais informações trocadas no âmbito deste contrato.

Cláusula 11 – Vigência e Rescisão

11.1. Este contrato terá vigência de [inserir prazo], podendo ser prorrogado por acordo escrito entre as partes.

11.2. A rescisão pode ocorrer por inadimplemento, desinteresse mútuo ou revogação dos direitos de uso, mediante aviso prévio de 30 dias.

Cláusula 12 – Penalidades

12.1. O descumprimento das cláusulas contratuais ensejará a responsabilização civil, inclusive com possibilidade de reparação por perdas e danos.

Cláusula 13 – Foro

13.1. Fica eleito o Foro da Comarca de Mogi Guaçu/SP para dirimir conflitos decorrentes deste contrato.

Mogi Guaçu, ____ de _____ de 20__.

Isabela Leme Cruz

Representante legal do projeto SCAN HANDS – Contratada

[Nome do representante da ETEC]

Representante da ETEC de Mogi Guaçu – Contratante

Testemunhas:

Nome: _____ | CPF: _____

Nome: _____ | CPF: _____

3 – DOCUMENTAÇÃO

A documentação é um componente essencial no gerenciamento e desenvolvimento de projetos, desempenhando um papel estratégico na comunicação, organização e continuidade das atividades. Ela consiste em um conjunto de registros detalhados que descrevem especificações, processos, decisões e procedimentos relacionados ao projeto ou sistema, funcionando como uma fonte confiável de consulta e orientação para todos os envolvidos (SOMMERVILLE, 2011).

Sua relevância é evidenciada pela capacidade de proporcionar clareza e precisão, garantindo que os membros da equipe compartilhem uma visão comum dos objetivos, das etapas e dos critérios de execução (PRESSMAN, 2014)

A documentação bem elaborada contribui diretamente para o alinhamento entre os participantes, reduzindo ambiguidades e promovendo maior eficiência na tomada de decisões. Além disso, ela atua como uma ferramenta indispensável para a manutenção e evolução dos sistemas, oferecendo uma base sólida para futuras modificações, correções e aprimoramentos, ao mesmo tempo em que preserva o conhecimento acumulado ao longo do projeto (KROLL & MACISAAC, 2012).

Normas e padrões internacionais, como os definidos pela ISO/IEC 15288, reforçam a importância da documentação como parte integrante dos processos técnicos e gerenciais, promovendo conformidade regulatória, padronização e qualidade nos resultados entregues (ISO/IEC, 2018).

A padronização e a documentação eficaz também facilitam a comunicação entre as partes interessadas e a integração de novos membros da equipe, ao fornecer uma referência clara e acessível para todos os aspectos do projeto. Além disso, a documentação bem estruturada contribui para a rastreabilidade das decisões tomadas ao longo do ciclo de vida do projeto, permitindo revisões, auditorias e ajustes com base em registros confiáveis e justificativas técnicas (BOOCH, RUMBAUGH & JACOBSON, 1999).

3.1 – OMG

O OMG (*Object Management Group* – Grupo de Gerenciamento de Objetos) é um consórcio internacional sem fins lucrativos fundado em 1989 por um grupo de empresas multinacionais com o objetivo de estabelecer padrões abertos que

promovam a interoperabilidade entre sistemas e aplicações em ambientes distribuídos e heterogêneos (OMG,2025).

Desde então, o OMG tem desempenhado um papel essencial na padronização de tecnologias da informação, reunindo empresas, instituições acadêmicas e órgãos governamentais para colaborar na criação e manutenção de especificações técnicas que sustentam a integração eficaz de soluções tecnológicas.

As Forças-Tarefa do OMG são responsáveis por desenvolver e atualizar esses padrões, garantindo que eles acompanhem a evolução das necessidades do mercado e das tecnologias emergentes. Entre os padrões mais conhecidos estão a UML (*Unified Modeling Language* – Linguagem de Modelagem Unificada), BPMN (*Business Process Model and Notation* – Modelo e Notação de Processos de Negócio) CORBA (*Common Object Request Broker Architecture* – Arquitetura Comum do Agente de Solicitação de Objetos) e DDS (*Data Distribution Service* – Serviço de Distribuição de Dados) todos amplamente utilizados em setores como engenharia de *software*, sistemas embarcados, automação industrial e saúde digital.

Dentro desse contexto, surgiu a OMA (*Object Management Architecture* – Arquitetura de Gerenciamento de Objetos), uma iniciativa do OMG que fornece uma estrutura conceitual robusta para o desenvolvimento de aplicações orientadas a objetos. A OMA define componentes como serviços de objetos, objetos de aplicação, facilidades comuns e o ORB (*Object Request Broker* – Corretor de Solicitação de Objetos) que atua como intermediário na comunicação entre objetos distribuídos.

Essa arquitetura visa garantir interoperabilidade, reutilização e portabilidade entre sistemas, promovendo uma base uniforme para a criação de soluções tecnológicas escaláveis e integradas. A missão do OMG permanece centrada na promoção da interoperabilidade e da integração entre ferramentas e sistemas de *software*, por meio da definição de padrões abertos que possam ser adotados globalmente. Essa abordagem tem permitido que organizações de diferentes setores colaborem e inovem com base em uma linguagem comum e em *frameworks* compartilhados, fortalecendo a coesão tecnológica em escala internacional.

3.1.1 – Utilização no Sistema

Para garantir que a aplicação SCAN HANDS operasse de forma eficiente e consistente em diferentes plataformas e sistemas operacionais, foi essencial seguir

as normas e diretrizes estabelecidas pela *Object Management Group* (OMG). A integração das especificações da OMG com a BPMN (*Business Process Model and Notation* – Modelo e Notação de Processos de Negócio) no projeto permitiu estruturar de maneira clara e funcional os processos envolvidos na interação entre usuários e a plataforma.

A utilização da BPMN possibilitou a definição visual e compreensível dos fluxos de atividades, como o reconhecimento de gestos, a realização de exercícios didáticos e a comunicação com professores. Isso facilitou a colaboração entre os envolvidos no desenvolvimento e aprimoramento do sistema, além de tornar os processos mais acessíveis e gerenciáveis.

A adoção dessas práticas também contribuiu para a criação de uma base flexível e adaptável, permitindo ajustes contínuos conforme as necessidades dos usuários evoluem. Dessa forma, o SCAN HANDS não apenas atende às demandas atuais de ensino e prática de Libras, mas também está preparado para incorporar futuras inovações, promovendo um ambiente de aprendizado inclusivo, interativo e em constante evolução.

3.1.2 – Instalação de Meta-Objeto

O MOF (*Meta-Object Facility* – Instalação de Meta-Objeto) é um padrão estabelecido pelo *Object Management Group* (OMG) que serve como base para a definição de metamodelos, ou seja, modelos que descrevem a estrutura, o comportamento e a semântica de outros modelos. Ele é amplamente utilizado em sistemas complexos para organizar e padronizar dados, como perfis de usuários, organizar os elementos principais da plataforma — como os tipos de usuários (alunos, professores), os gestos reconhecidos e os critérios de avaliação.

Por meio do MOF, é possível criar linguagens de modelagem específicas e consistentes, que facilitam a interoperabilidade entre ferramentas e plataformas. Essa abordagem é essencial em ambientes que exigem precisão na modelagem de dados e flexibilidade na adaptação de estruturas conforme as necessidades do negócio.

3.2 – BPMN

A BPMN (*Business Process Model and Notation* – Modelo e Notação de Processos de Negócio) é uma notação gráfica desenvolvida pela OMG (*Object Management Group* – Grupo de Gestão de Objetos) com o objetivo de representar

processos de negócios de forma clara e padronizada. Ela permite que profissionais de diferentes áreas, como analistas, gestores e desenvolvedores, compreendam e colaborem na modelagem de processos, eliminando ambiguidades e facilitando a comunicação entre todos os envolvidos.

Desde sua adoção pela OMG em 2005, a BPMN tem evoluído para atender às demandas de transformação digital, sendo atualmente utilizada em sua versão 2.0.2. (IPROCESS, 2025)

A principal vantagem da BPMN está na sua capacidade de traduzir fluxos de trabalho complexos em diagramas visuais intuitivos, que podem ser utilizados tanto para documentação quanto para automação de processos. Isso contribui diretamente para a melhoria da eficiência operacional, redução de custos e aumento da agilidade organizacional. Além disso, a notação é compatível com diversas ferramentas de modelagem, como Bizagi, HEFLO, Pipefy e Microsoft Visio, que permitem criar, documentar e até simular processos de forma colaborativa e segura.

A BPMN também se destaca por sua flexibilidade, podendo ser aplicada em diferentes setores e tipos de processos, desde os mais simples até os mais sofisticados. Essa adaptabilidade torna a notação uma aliada estratégica na busca por inovação, padronização e excelência na gestão de processos.

Versões:

1.0 (2004): A BPMN 1.0 lançada em 2004 pela BPMI, apresentando uma notação básica para modelar processos de negócios. Seu objetivo era oferecer uma linguagem visual clara e acessível para analistas e técnicos. (TRISOTECH, 2024)

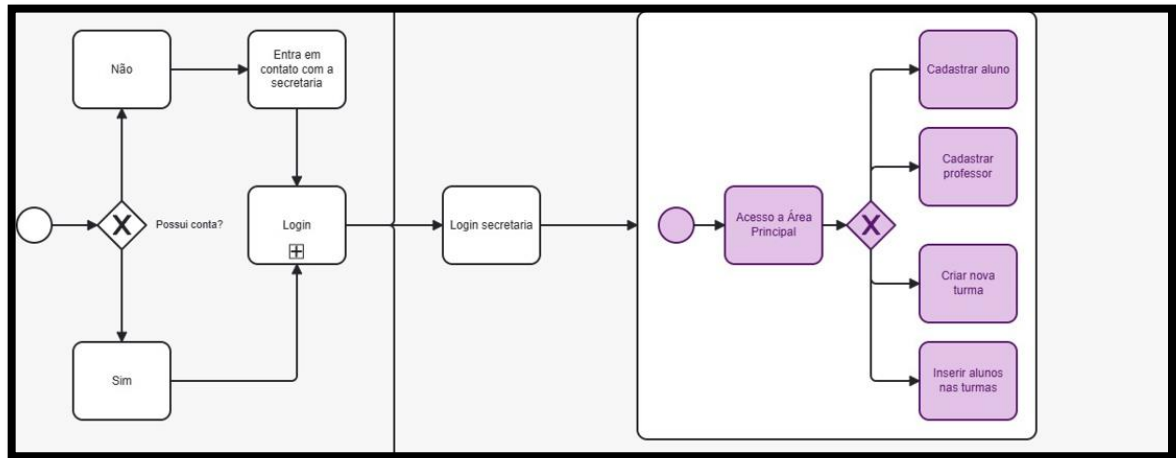
1.1 (2006): Após a fusão da BPMI com o OMG, foi lançada a versão 1.1 da BPMN. Essa atualização trouxe melhorias nos símbolos e conceitos, aumentando a precisão e a usabilidade da notação. (OMG, 2025)

2.0 (2011): BPMN 2.0, oficializada em 2013, aprimorou a execução dos processos, refinou eventos e adicionou novos diagramas para representar interações e tarefas humanas. (IPROCESS, 2025)

2.0.2 (2013): Publicada como padrão internacional ISO/IEC 19510, essa versão trouxe ajustes técnicos e refinamentos na notação, garantindo maior compatibilidade com outras normas e práticas de modelagem (NEOMIND,2024).

3.2.1 – BPMN Secretaria

Figura 52 - Tela BPMN secretaria



FONTE: AUTORIA PRÓPRIA, 2025

A figura 52 representa o processo de entrada e navegação dentro de um sistema escolar voltado à área administrativa. A estrutura do fluxo é organizada para guiar o usuário desde o primeiro contato com o sistema até a execução de funções específicas. Ele apresenta decisões claras e caminhos bem definidos, facilitando o entendimento das etapas envolvidas e promovendo uma experiência mais intuitiva para o usuário.

Verificação de Conta e *Login*:

O ponto inicial do fluxo é uma decisão: o usuário já possui conta no sistema. Caso não tenha, o processo é interrompido com a orientação para entrar em contato com a secretaria, responsável pelo cadastro. Se o usuário já for registrado, ele é direcionado à etapa de *login*. Ambas as rotas convergem para o acesso como secretaria, que é o perfil administrativo com permissões ampliadas.

Área Principal e Funcionalidades:

Após o *login*, o sistema libera o acesso à área principal, onde estão concentradas as ferramentas de gestão. O diagrama apresenta quatro funcionalidades centrais:

Cadastrar aluno:

Permite o registro de novos estudantes, com inserção de dados pessoais e vinculação a turmas específicas.

Cadastrar professor:

Funcionalidade voltada à inclusão de docentes no sistema, garantindo que estejam habilitados para atuar nas turmas.

Criar nova turma:

Ferramenta para estruturar novas turmas, definindo nome, nível de ensino e demais parâmetros pedagógicos.

Inserir alunos nas turmas:

Processo de alocação dos estudantes nas turmas criadas, essencial para o controle acadêmico e organização das aulas.

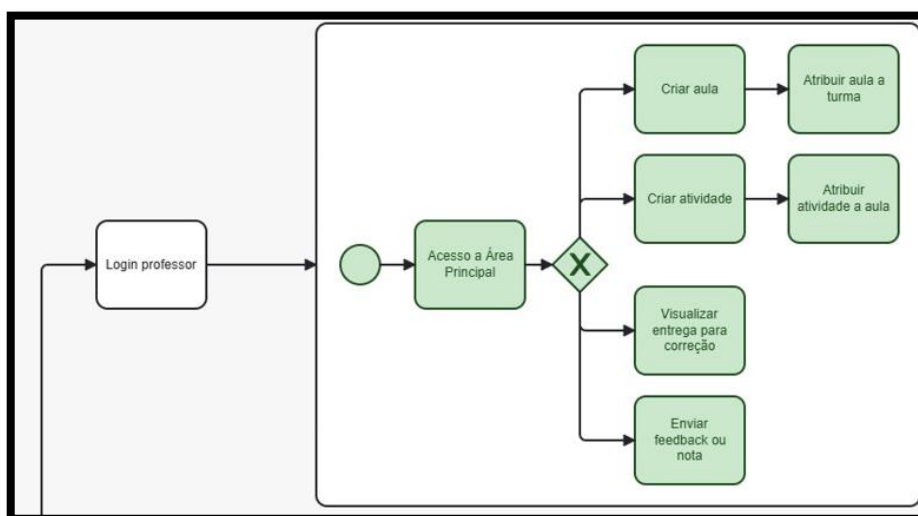
Integração e Eficiência Operacional:

O fluxo apresentado no diagrama reforça a lógica sequencial e a clareza na navegação entre etapas. A separação entre verificação de conta e funcionalidades administrativas garante segurança no acesso e precisão na execução das tarefas. A estrutura visual facilita o entendimento do processo, contribuindo para uma gestão escolar mais eficiente e organizada. Além disso, o uso desse tipo de diagrama proporciona uma visão sistematizada do funcionamento da aplicação, permitindo que desenvolvedores e gestores compreendam facilmente o comportamento esperado do sistema. A clareza no mapeamento de decisões e ações contribui também para a identificação de melhorias, redução de erros e suporte à documentação do sistema como um todo.

Esse tipo de abordagem ainda favorece a comunicação entre diferentes setores da instituição, permitindo que todos os envolvidos compartilhem uma mesma compreensão do fluxo de trabalho. A padronização visual auxilia no treinamento de novos colaboradores, tornando o processo de adaptação mais rápido e menos suscetível a falhas. Outro ponto importante é a possibilidade de atualização e manutenção contínua do sistema, já que o diagrama facilita a identificação de gargalos ou redundâncias. Com uma visão clara das etapas, torna-se mais viável implementar novas funcionalidades sem comprometer a lógica já existente. Por fim, essa representação gráfica serve também como um instrumento de auditoria e controle, garantindo rastreabilidade e transparência em todas as ações realizadas no ambiente escolar.

3.2.2 – BPMN Professor

Figura 53 – Tela BPMN professor



FONTE: AUTORIA PRÓPRIA, 2025

O diagrama da figura 53 apresenta o percurso funcional do usuário com perfil de professor, detalhando desde o processo de autenticação até a execução das principais tarefas pedagógicas dentro do sistema. A estrutura é organizada de forma sequencial e lógica, permitindo uma visualização clara das etapas envolvidas na rotina docente e favorecendo a compreensão de como o sistema apoia o processo educacional.

Login e Acesso à Área Principal:

O fluxo tem início com o processo de autenticação do professor, representado pelo bloco “Login professor”. Após a validação das credenciais, o usuário é direcionado à “Área Principal”, ambiente central onde estão concentradas as ferramentas de trabalho voltadas à gestão das aulas, atividades e interações com os alunos.

Criar aula:

Permite ao docente estruturar uma nova aula, definindo conteúdo, objetivos, metodologia e recursos didáticos.

Atribuir aula à turma:

Após a criação, a aula pode ser vinculada a uma ou mais turmas específicas, garantindo que os alunos certos tenham acesso ao conteúdo planejado.

Criar atividade:

Funcionalidade voltada à elaboração de exercícios, tarefas ou avaliações, que complementam o conteúdo trabalhado em aula.

Atribuir atividade à aula:

As atividades são associadas às respectivas aulas, mantendo a organização pedagógica e o alinhamento com o planejamento de ensino.

Visualizar entrega para correção: O professor pode acessar as atividades enviadas pelos alunos, avaliando individualmente cada resposta ou arquivo submetido.

Enviar *feedback* ou nota:

Após a correção, é possível registrar comentários, observações ou notas, promovendo um retorno construtivo e alinhado às práticas avaliativas da instituição.

Organização e Eficiência:

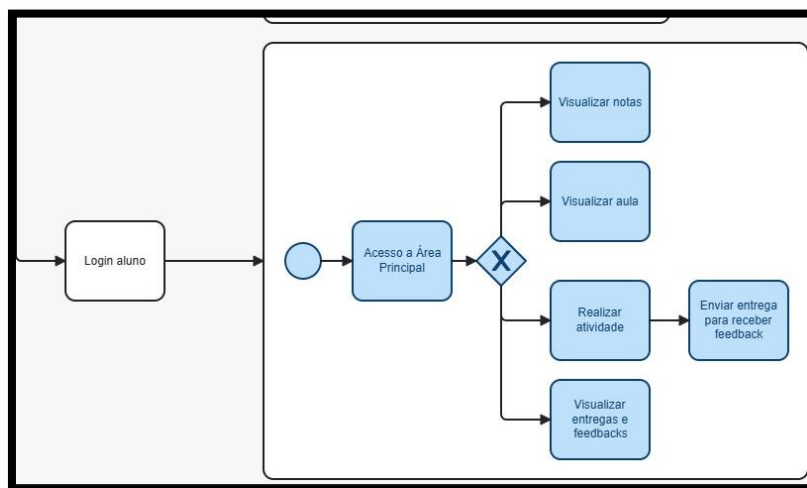
O diagrama destaca a fluidez do processo, com caminhos bem definidos que orientam o docente do início ao fim de cada tarefa. A organização das ações por blocos funcionais reforça a autonomia do professor na gestão de suas atividades, ao mesmo tempo em que assegura a integração com o restante do sistema. Essa abordagem contribui significativamente para a produtividade docente, fortalecendo a gestão educacional e a qualidade do acompanhamento pedagógico oferecido aos alunos. Além disso, o modelo apresentado serve como uma base visual para entender a lógica operacional do sistema, sendo útil tanto para fins de documentação técnica quanto para treinamentos e manuais de uso da plataforma.

Ao permitir uma visão clara das etapas e interações, o diagrama favorece a comunicação entre professores, gestores e desenvolvedores, garantindo que todos compreendam de forma uniforme os processos implementados. Essa representação gráfica também facilita a identificação de possíveis gargalos ou melhorias, promovendo ajustes ágeis e eficientes. Com a padronização visual, novos usuários podem ser capacitados de forma mais rápida e segura, reduzindo erros e aumentando a confiabilidade do sistema. Além disso, a manutenção e evolução das funcionalidades tornam-se mais assertivas, uma vez que o mapeamento das etapas

orienta futuras atualizações. Por fim, o uso desse tipo de recurso reforça a transparência e a rastreabilidade das ações, contribuindo para uma gestão educacional mais estratégica e baseada em evidências.

3.2.3 – BPMN Aluno

Figura 54 - Tela do BPMN aluno



FONTE: AUTORIA PRÓPRIA, 2025

O diagrama da figura 54 apresenta o percurso funcional do aluno dentro do sistema educacional, desde o momento do *login* até a realização de atividades acadêmicas. A estrutura do fluxo é organizada para garantir uma navegação intuitiva e eficiente, permitindo ao estudante acessar os principais recursos disponíveis na plataforma.

Login e Acesso à Área Principal:

O fluxo tem início com o processo de autenticação do aluno, representado pelo bloco “*Login* aluno”. Após a validação das credenciais, o usuário é direcionado à “Área Principal”, ambiente central onde estão concentradas as funcionalidades voltadas à rotina acadêmica.

Funcionalidades Disponíveis:

A partir da área principal, o aluno pode executar diversas ações, organizadas em blocos que representam tarefas de acompanhamento e participação nas atividades escolares:

Visualizar notas:

Permite ao estudante acompanhar seu desempenho acadêmico por meio das avaliações registradas no sistema.

Visualizar aula:

Funcionalidade que oferece acesso ao conteúdo das aulas disponibilizadas pelos professores.

Realizar atividade:

Ferramenta que possibilita a execução de tarefas propostas, como exercícios, trabalhos ou avaliações.

Visualizar entregas e *feedbacks*:

O aluno pode consultar as atividades já enviadas e os retornos dados pelos professores, como comentários ou notas.

Enviar entrega para receber *feedback*:

Processo de envio de atividades para correção, permitindo que o professor avalie e registre observações.

Organização e Clareza no Fluxo:

O diagrama utiliza símbolos padronizados para representar ações e decisões, facilitando a compreensão do percurso lógico. A estrutura favorece a autonomia do aluno na gestão de suas atividades, promovendo uma experiência de uso eficiente, intuitiva e alinhada às demandas pedagógicas da instituição.

3.3 – UML

UML (*Unified Modeling Language* – Linguagem de Modelagem Unificada) é uma linguagem de modelagem visual amplamente utilizada na engenharia de *software*. Sua principal função é auxiliar no planejamento, desenvolvimento e documentação de sistemas, facilitando a visualização da estrutura e do comportamento de um *software* antes mesmo de sua implementação. Ao oferecer uma representação gráfica dos componentes e suas interações, a UML contribui para a identificação precoce de erros, melhora a comunicação entre os membros da equipe e promove maior organização no desenvolvimento.

Essa linguagem permite representar tanto aspectos estáticos quanto dinâmicos de um sistema por meio de diversos tipos de diagramas. Os diagramas estruturais descrevem a estrutura interna do sistema, como classes, objetos, componentes e suas inter-relações. Já os diagramas comportamentais representam o comportamento do sistema em tempo de execução, demonstrando como os elementos interagem entre si e com usuários ou outros sistemas externos.

A UML surgiu da necessidade de padronização das técnicas de modelagem orientadas a objetos. Durante as décadas de 1980 e 1990, diferentes metodologias eram utilizadas, como a metodologia de Booch, a OMT (*Object Modeling Technique – Técnica de Modelagem de Objetos*) e a OOSE (*Object-Oriented Software Engineering – Engenharia de Software Orientada a Objetos*). Cada uma possuía suas próprias notações, o que dificultava a integração entre equipes que utilizavam abordagens diferentes (LUCIDCHART,2025).

Para resolver esse problema, em 1994, a empresa Rational *Software* reuniu os criadores dessas metodologias — Grady Booch, Ivar Jacobson e James Rumbaugh, conhecidos como os “Três Amigos” com o objetivo de unificar as abordagens existentes. Como resultado dessa colaboração, surgiu a UML, cuja primeira versão preliminar foi lançada em 1995. A proposta era criar uma linguagem padrão, clara e eficiente, capaz de ser compreendida e adotada por profissionais em toda a indústria de *software*. (LUCIDCHART,2025).

Versões:

1.1 (1997): A primeira versão oficial da UML, adotada pela OMG, teve como objetivo padronizar a notação para modelagem de sistemas orientados a objetos, unificando diversas abordagens existentes. (IBM,2021)

1.3 (1999): Essa versão aprimorou a notação e estabilizou os diagramas, tornando a linguagem mais consistente e adequada para aplicação em projetos reais. (IBM,2021)

1.5 (2003): Introduziu o conceito de perfis UML, permitindo que a linguagem fosse adaptada a diferentes domínios específicos. Também trouxe avanços na extensibilidade, tornando a UML mais flexível. (IBM,2021)

2.0 (2005): A versão 2.0, reformulou a linguagem com novos diagramas e melhorias na semântica, oferecendo suporte mais completo para modelar sistemas complexos. (IBM,2021)

2.4.1 (2011):Focou em ajustes técnicos e refinamentos na padronização. Não trouxe grandes mudanças visuais, mas melhorou a clareza e a consistência da linguagem. (IBM,2021)

2.5 (2015): Essa versão simplificou a estrutura interna da linguagem e tornou a documentação mais clara, visando facilitar a compreensão por desenvolvedores e modeladores. (IBM,2021)

2.5.1 (2017): Versão mais recente e estável, consolidou melhorias anteriores e foi reconhecida como padrão ISO, aumentando sua confiabilidade e aceitação global. (IBM,2021)

3.3.1 – Aplicação no Sistema

O projeto SCAN HANDS adota a UML como linguagem de modelagem unificada, utilizando diagramas para representar graficamente as funcionalidades e interações do sistema. Diagramas são representações visuais que descrevem ações entre componentes e usuários, sendo os principais utilizados o diagrama de caso de uso e o diagrama de atividade.

O diagrama de caso de uso é empregado para ilustrar as ações realizadas pelos usuários em relação ao sistema. No contexto do SCAN HANDS, esse diagrama representa funcionalidades como iniciar a captura de gestos, processar imagens, traduzir sinais e exibir resultados. Para organizar essas ações, são utilizados os relacionamentos “*Include*” e “*Extend*”, que direcionam o fluxo de execução, indicando quais ações são obrigatórias e quais são opcionais.

O relacionamento *Include* é utilizado para garantir a continuidade das ações do sistema, representando funcionalidades que devem ser executadas obrigatoriamente dentro de um processo. Por exemplo, ao realizar a tradução de um gesto, é necessário que o sistema execute a etapa de reconhecimento da imagem. O relacionamento seria: Traduzir gesto <*include*> Reconhecer imagem

O relacionamento *Extend* representa ações que podem ser executadas opcionalmente, geralmente associadas a funcionalidades complementares. No projeto

SCAN HANDS, após a exibição da tradução, o sistema pode oferecer a opção de salvar o resultado em um histórico. O relacionamento seria: Exibir tradução <extend> Salvar no histórico

Esses relacionamentos são fundamentais para definir a lógica entre os casos de uso, contribuindo para uma modelagem mais precisa e compreensível. O diagrama de atividade também é utilizado no projeto, sendo responsável por detalhar o fluxo interno das operações do sistema. Esse diagrama pertence ao grupo de diagramas de comportamento e descreve com clareza o que deve ocorrer em cada etapa do processo, desde a captura do gesto até a entrega da tradução.

Para sua construção, são utilizados símbolos específicos que representam o início, o encerramento, decisões, fusões de processos e ações paralelas, organizando as etapas de forma lógica e acessível. A aplicação de diagramas UML no projeto SCAN HANDS contribui para a estruturação da lógica do sistema, além de funcionar como ferramenta de comunicação entre os membros da equipe.

3.4 – DIAGRAMAS

A criação de diagramas UML permite representar visualmente sistemas complexos, facilitando a análise, o desenvolvimento e a documentação de projetos de *software*. A UML (*Unified Modeling Language*) é uma linguagem de modelagem padronizada que descreve aspectos estruturais e comportamentais de sistemas orientados a objetos. Por meio de notações gráficas, é possível ilustrar classes, objetos, interações, atividades e estados, promovendo maior clareza na comunicação entre equipes técnicas e não técnicas. (LUCIDCHART,2025).

A linguagem UML foi desenvolvida com o objetivo de unificar os diversos métodos de modelagem existentes, sendo posteriormente adotada como padrão pelo consórcio internacional OMG (*Object Management Group* – Grupo de Gestão de Objetos). Desde sua padronização, a UML passou por diversas revisões, consolidando-se como uma ferramenta essencial na engenharia de *software*. Diagramas UML podem ser classificados em dois grandes grupos: diagramas estruturais, como o diagrama de classes e o diagrama de componentes, e diagramas comportamentais, como o diagrama de casos de uso e o diagrama de sequência.

Cada tipo de diagrama possui uma finalidade específica, contribuindo para diferentes etapas do ciclo de vida do sistema. Além de facilitar a comunicação entre equipes técnicas e não técnicas, a UML promove padronização na documentação de sistemas. Seu uso é amplamente adotado em projetos de *software* orientado a objetos, desde a fase de análise até a implementação.

A clareza visual dos diagramas permite identificar falhas de lógica, dependências excessivas e oportunidades de refatoração. Com isso, a UML se torna uma aliada estratégica na construção de sistemas robustos, escaláveis e bem estruturados.

3.4.1– Diagrama de Classe

O diagrama de classe é essencial na modelagem de sistemas orientados a objetos porque ele define a estrutura estática do sistema. Isso significa que ele mostra como os dados são organizados e como os objetos se relacionam entre si, antes mesmo de o sistema ser implementado. Ele serve como uma espécie de *blueprint* para os desenvolvedores.

Cada classe representa um conceito ou entidade do sistema. Por exemplo, em um sistema educacional, o usuário pode ter classes como Professor, Aluno, Aula, Categoria, Vídeo. Cada uma dessas classes terá atributos (como nome, *email*, *dataDeNascimento*) e métodos como *cadastrarAula()*, *excluirAluno()*. Além disso, o diagrama de classe permite visualizar a multiplicidade dos relacionamentos. Por exemplo, um Professor pode estar associado a várias Aulas, mas cada Aula pertence a um único Professor. Isso é representado com notações como 1..*, 0..1, etc., que indicam quantos objetos de uma classe podem estar relacionados a objetos de outra.

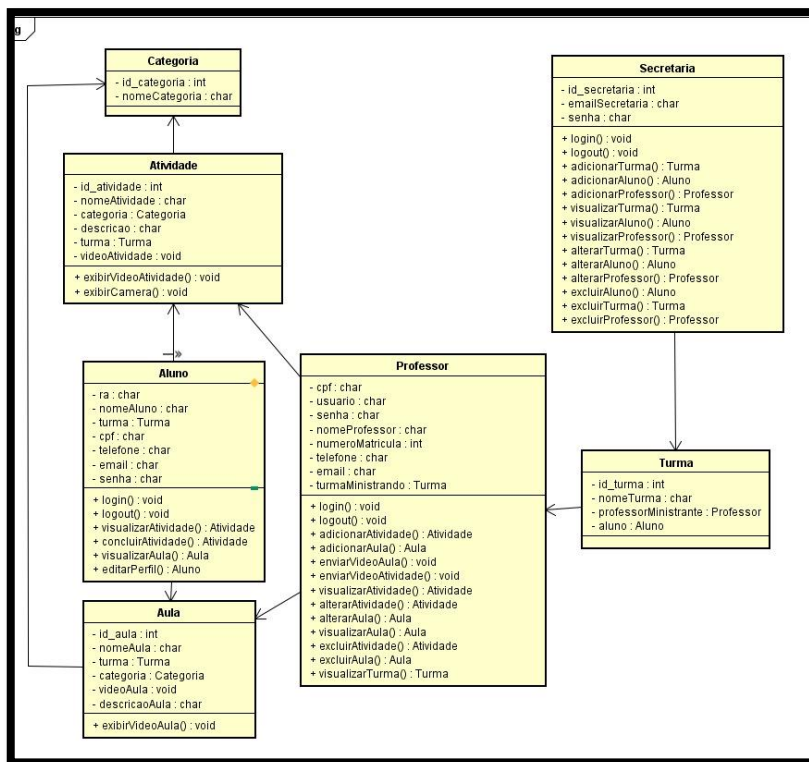
Outro ponto importante é a visibilidade dos atributos e métodos. No diagrama, símbolos como “+”, “-” e “#” indicam se o elemento é público, privado ou protegido, respectivamente. Isso ajuda a definir o encapsulamento, que é um dos princípios fundamentais da programação orientada a objetos.

Além disso, o diagrama de classe pode servir como base para discutir regras de negócio e validar requisitos com os *stakeholders* antes do desenvolvimento, evitando retrabalho. Ele ainda permite mapear herança, interfaces e classes abstratas, deixando explícitas as generalizações e especializações presentes no sistema. Ferramentas modernas de modelagem possibilitam a geração automática de esboços de código a partir do diagrama, acelerando o desenvolvimento e reduzindo erros.

Por fim, o diagrama de classe também é útil para documentação técnica, facilitando a manutenção, evolução e auditoria do sistema ao longo do tempo. A clareza na representação de relacionamentos e responsabilidades torna mais simples

a comunicação entre analistas, desenvolvedores e demais interessados, garantindo que todos tenham uma visão unificada da estrutura do *software*.

Figura 55 - Tela de diagrama de classe



FONTE: AUTORIA PRÓPRIA, 2025

No diagrama de classes representado na Figura 55, é apresentada a estrutura fundamental do sistema de gestão educacional, que envolve os principais atores da instituição: secretaria, professores e alunos. O objetivo do diagrama é demonstrar como as entidades se organizam, se relacionam e quais funcionalidades cada uma pode desempenhar no ambiente digital.

A classe Secretaria representa o perfil administrativo com acesso a funcionalidades mais amplas. Esse ator é responsável por cadastrar e gerenciar turmas, alunos e professores. A classe possui atributos como *id_secretaria*, *emailSecretaria* e *senha*, além de métodos que permitem realizar ações como adicionar e excluir usuários, visualizar informações e alterar dados cadastrados no sistema. Já a classe Professor contempla atributos pessoais, como CPF, nomeProfessor e telefone, e possui um conjunto de métodos voltados à rotina pedagógica. Dentre as funcionalidades acessíveis estão: criação de atividades e aulas, envio de vídeos, visualização de turmas e correções, além de atribuição de

atividades às aulas. A associação com a classe Turma indica que um professor pode ministrar mais de uma turma.

A classe Aluno, por sua vez, possui dados como ra, nomeAluno, *email* e senha, além de métodos para *login*, visualização de conteúdo, entrega de atividades e edição de perfil. Cada aluno está vinculado a uma turma, e pode interagir com as aulas e atividades conforme a organização definida pelo professor. As entidades Atividade e Aula representam os conteúdos pedagógicos elaborados pelo professor. Ambas possuem atributos como título, vídeo e descrição, além de estarem associadas a categorias específicas.

A classe Categoria, por sua vez, permite organizar os conteúdos por temas, como "Verbos", "Esportes", entre outros, facilitando a navegação e filtragem dentro do sistema. Por fim, a classe Turma centraliza a relação entre os professores e alunos, permitindo a administração de grupos distintos de aprendizagem. Cada turma possui um nome, um professor responsável e uma lista de alunos associados, reforçando a estrutura colaborativa entre os agentes envolvidos.

Assim, o diagrama de classes documenta a arquitetura estática do sistema, organizando os principais elementos, suas propriedades e responsabilidades, e fornecendo uma base sólida para o desenvolvimento e manutenção da aplicação. Ele oferece uma visão geral das funcionalidades e da lógica interna do sistema, tornando mais clara a dinâmica entre os usuários e os objetos manipulados.

3.4.2 – Diagrama de Caso de Uso

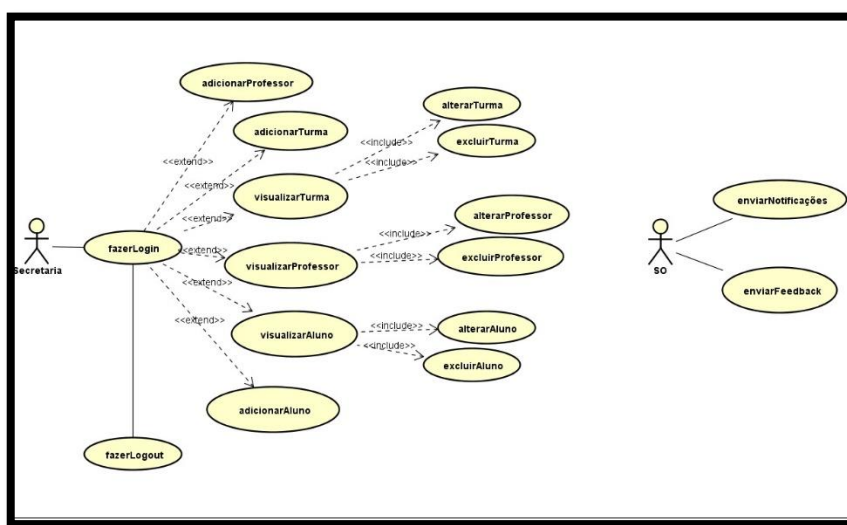
O diagrama de caso de uso é uma ferramenta da UML (*Unified Modeling Language* – Linguagem de Modelagem Unificada) utilizada para representar as funcionalidades de um sistema sob a perspectiva dos usuários. Ele descreve quem interage com o sistema (atores) e quais ações podem ser realizadas (casos de uso), sem entrar em detalhes técnicos de implementação. Um ator pode ser uma pessoa, outro sistema ou qualquer entidade externa que interaja com o sistema. Os casos de uso são representados como elipses e indicam as funcionalidades oferecidas, como "Cadastrar Aula", "Excluir Aula", "Visualizar Vídeo", "Gerenciar Categorias".

Esse tipo de diagrama é útil para entender os requisitos do sistema, facilitar a comunicação entre desenvolvedores e clientes, e servir como base para testes e

documentação. Ele também ajuda a identificar os limites do sistema, mostrando o que está dentro e fora do escopo.

Por exemplo, em um sistema de gerenciamento de aulas, os atores podem ser "Professor" e "Administrador", e os casos de uso podem incluir "Criar Aula", "Editar Aula", "Excluir Aula", "Associar Categoria", "Visualizar Relatórios". As relações entre atores e casos de uso são representadas por linhas, e é possível indicar dependências entre os casos, como "Excluir Aula" dependendo de "Selecionar Aula".

Figura 56 - Tela de Diagrama de caso de uso



FONTE: AUTORIA PRÓPRIA, 2025

O diagrama de casos de uso representado na Figura 56 descreve as principais interações entre o sistema educacional e seus usuários, destacando os perfis da Secretaria e do Sistema Operacional (SO). Ele apresenta os recursos e funcionalidades que esses atores podem acessar, organizados em casos de uso que evidenciam a lógica de funcionamento do sistema sob a perspectiva dos usuários.

A Secretaria, como perfil administrativo, é responsável por realizar operações fundamentais relacionadas ao gerenciamento de usuários e turmas. Entre as funcionalidades disponíveis estão os casos de uso *fazerLogin* e *fazerLogout*, que controlam o acesso ao sistema. A partir do *login*, o sistema permite que a secretaria adicione professores, alunos e turmas através dos casos de uso *adicionarProfessor*, *adicionarAluno* e *adicionarTurma*.

Além disso, a secretaria pode executar ações de manutenção e visualização de dados por meio dos casos de uso visualizarProfessor, visualizarAluno e visualizarTurma. Essas ações incluem funcionalidades mais específicas como alterar ou excluir registros, representadas nos casos de uso alterarProfessor, excluirProfessor, alterarAluno, excluirAluno, alterarTurma e excluirTurma. Esses casos são conectados por meio da relação «*include*», o que indica que são complementares às ações principais de visualização.

As interações estão organizadas com o auxílio das relações «*include*» e «*extend*», que definem, respectivamente, ações obrigatórias e opcionais associadas a uma funcionalidade principal. Por exemplo, o caso visualizarProfessor inclui as opções de alterarProfessor ou excluirProfessor, indicando que essas funcionalidades só são acessadas a partir da visualização prévia do professor.

À direita do diagrama, é tido o ator SO (Sistema Operacional), que representa uma funcionalidade automatizada ou externa ao usuário comum, responsável por ações como enviarNotificações e enviarFeedback. Essas ações geralmente ocorrem em segundo plano, reforçando a comunicação do sistema com os usuários por meio de alertas, lembretes ou respostas automáticas.

Este diagrama documenta claramente os requisitos funcionais esperados do sistema para o perfil administrativo e destaca a organização modular das funcionalidades, promovendo um fluxo lógico e seguro para a manipulação de dados sensíveis dentro do ambiente educacional.

3.5 – APLICAÇÃO WEB

O *site* a ser desenvolvido será uma aplicação *web* voltada para a tradução de Libras em tempo real, utilizando inteligência artificial e visão computacional. O sistema permitirá que usuários acessem a plataforma por meio de navegadores, ativem a câmera de seus dispositivos e realizem a captura de gestos manuais, que serão interpretados e convertidos em texto ou áudio. A aplicação será especialmente útil em ambientes acadêmicos, educacionais e sociais, promovendo inclusão digital e acessibilidade comunicacional.

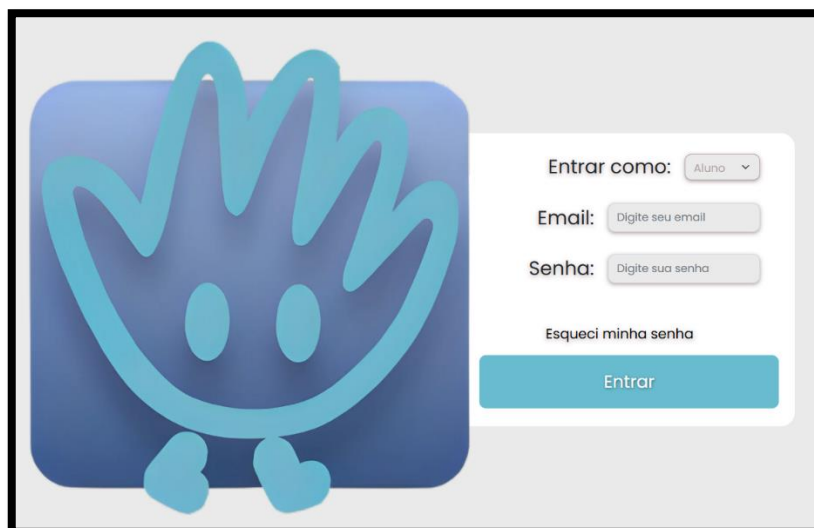
Os usuários poderão realizar *login* para acessar funcionalidades personalizadas, como histórico de traduções, configurações de acessibilidade e preferências de idioma. A plataforma contará com uma interface intuitiva, exibindo os

sinais capturados, sua tradução correspondente e opções de interação com o conteúdo traduzido. O sistema também permitirá o acompanhamento da precisão das traduções, oferecendo *feedback* visual e estatísticas de desempenho.

Para os administradores, o sistema oferecerá ferramentas para gerenciar usuários, acompanhar métricas de uso, atualizar o modelo de reconhecimento e realizar testes de validação. Além disso, será possível gerar relatórios sobre o volume de traduções realizadas, os sinais mais utilizados e o desempenho do sistema em diferentes contextos. A aplicação garantirá uma experiência fluida, segura e acessível, contribuindo para a democratização da comunicação entre surdos e ouvintes.

3.5.1 - Tela de *Login*

Figura 57 - Tela de *login*



FONTE: AUTORIA PRÓPRIA, 2025

A Figura 57 apresenta a tela de *login* do sistema, composta por uma interface moderna, limpa e intuitiva. No lado esquerdo da tela, há um elemento visual em destaque: um ícone gráfico estilizado com um rosto sorridente e mãos, possivelmente representando a identidade visual da aplicação ou instituição.

À direita, encontra-se o formulário de autenticação, que permite ao usuário acessar o sistema. Esse formulário é composto por três campos principais: um menu suspenso com a opção "Entrar como", onde o usuário seleciona seu tipo de perfil (por exemplo, Aluno, Secretaria ou Professor); o campo de *email*, onde o usuário digita seu endereço eletrônico; e o campo de senha, que por padrão oculta os *caracteres* digitados para proteger as informações sensíveis.

Abaixo dos campos de entrada, há a opção "Esqueci minha senha", funcionalidade essencial para permitir a recuperação de acesso por parte do usuário. Em seguida, um botão com o rótulo "Entrar" executa a ação de envio dos dados para validação junto ao sistema.

Figura 58 - Tela de códigos do *login*

```

1  <?php
2  session_start();
3
4  $host = "localhost";
5  $usuario = "u357936358_librinha";
6  $senha = "Librinh4tcc#";
7  $banco = "u357936358_scanhands";
8
9  $conn = new mysqli($host, $usuario, $senha, $banco);
10 if ($conn->connect_error) {
11     die("Conexão falhou: " . $conn->connect_error);
12 }
13
14 $email = $conn->real_escape_string($_POST['email'] ?? '');
15 $senhaDigitada = $conn->real_escape_string($_POST['senha'] ?? '');
16 $tipoSelecionado = $conn->real_escape_string($_POST['tipo'] ?? '');
17
18 $sql = "SELECT * FROM usuario WHERE email = '$email' LIMIT 1";
19 $result = $conn->query($sql);
20
21 if ($result && $result->num_rows > 0) {
22     $usuario = $result->fetch_assoc();
23
24     if ($usuario['tipo'] === $tipoSelecionado && $usuario['senha'] === $senhaDigitada) {
25         $_SESSION['id_usuario'] = $usuario['id_usuario'];
26         $_SESSION['tipo'] = $usuario['tipo'];
27         $_SESSION['email'] = $usuario['email'];
28
29         // Redirecionamento com base no tipo
30         if ($usuario['tipo'] === 'aluno') {
31             header("Location: ../TCC/assets/paginas/inicioAluno.php");
32         } elseif ($usuario['tipo'] === 'secretaria') {
33             header("Location: ../TCC/assets/paginas/inicioSecretaria.php");
34         } elseif ($usuario['tipo'] === 'professor') {
35             header("Location: ../TCC/assets/paginas/inicioProfessor.php");
36         } else {
37             $_SESSION['erro_login'] = "Tipo de usuário inválido.";
38             header("Location: index.php");
39         }
40         exit();
41     } else {
42         $_SESSION['erro_login'] = "Senha ou tipo de usuário incorretos.";
43         header("Location: index.php");
44         exit();
45     }
46 } else {
47     $_SESSION['erro_login'] = "Usuário não encontrado.";
48     header("Location: index.php");
49     exit();
50 }
51
52 $conn->close();
53
54

```

FONTE: AUTORIA PRÓPRIA, 2025

Na figura 58, a porção de código apresentada realiza o processo de autenticação de usuários no sistema, com base no *e-mail*, senha e tipo de perfil selecionado. O *script* inicia com a função “*session_start()*”, que permite o uso de variáveis de sessão para manter o estado do usuário durante a navegação. Em

seguida, é estabelecida a conexão com o banco de dados MySQL utilizando os parâmetros de acesso: “*host*”, nome de usuário, senha e nome do banco.

Logo após os dados enviados pelo formulário — *e-mail*, senha e tipo de usuário — são capturados e tratados com “*real_escape_string()*”, garantindo que caracteres especiais sejam escapados e prevenindo ataques de injeção de SQL. Esses dados são armazenados nas variáveis “*\$email*”, “*\$senhaDigitada*” e “*\$tipoSelecionado*”.

O sistema então executa uma consulta SQL para buscar o usuário com o *e-mail* informado, utilizando a instrução “*SELECT * FROM usuário WHERE email = \$email LIMIT 1*”. Se o usuário for encontrado, os dados são armazenados na variável “*\$usuario*” por meio da função “*fetch_assoc()*”.

Em seguida, o sistema verifica se o tipo de usuário informado corresponde ao tipo armazenado no banco e se a senha digitada é igual à senha registrada. Essa verificação é feita com a condição “*if (\$usuario[“tipo”] == “\$tipoSelecionado” && \$usuario[“senha”] === “\$senhaDigitada”)*”. Caso ambas as condições sejam verdadeiras, o sistema armazena os dados do usuário na sessão, incluindo o ID, o tipo e o *e-mail*, e realiza o redirecionamento para a página inicial correspondente ao tipo de perfil: Se o tipo for aluno, o usuário é redirecionado para “*inicioAluno.php*.” já secretaria, é redirecionado para “*inicioSecretaria.php*.” e professor, é redirecionado para “*inicioProfessor.php*.”

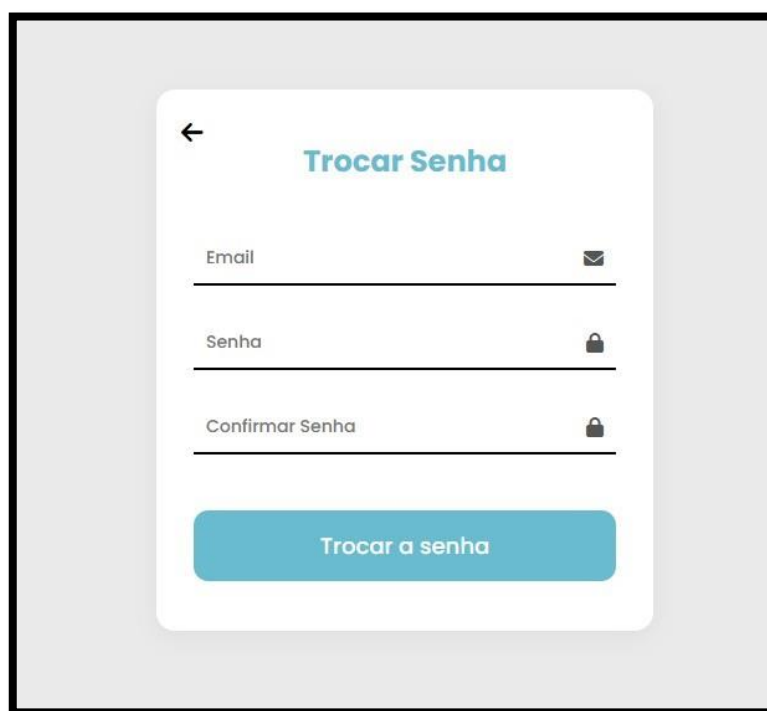
Caso o tipo de usuário seja inválido, a variável de sessão “*erro_login*” recebe a mensagem “Tipo de usuário inválido”, e o usuário é redirecionado para a página de *login*. Se a senha ou o tipo estiverem incorretos, ou se o usuário não for encontrado, mensagens de erro específicas são definidas na sessão e o redirecionamento para “*index.php*” é realizado.

Por fim, a conexão com o banco de dados é encerrada com *\\$conn->close()*, liberando os recursos utilizados pela aplicação. Embora o código funcione corretamente para autenticação, é importante destacar que a senha está sendo comparada diretamente com o valor armazenado no banco de dados, o que representa uma falha de segurança significativa.

Recomenda-se fortemente a utilização das funções `password_hash()` para criptografar a senha no momento do cadastro, e `password_verify()` para verificar a senha durante o *login*. Essas funções implementam algoritmos robustos como *bcrypt*, que protegem os dados sensíveis dos usuários contra acessos indevidos, ataques de força bruta ou vazamentos.

3.5.1.1 - Tela de redefinição de Senha

Figura 59 - Tela de redefinição de Senha

A imagem mostra uma interface de usuário para a troca de senha. No topo, há uma seta para voltar e o título "Trocar Senha". Abaixo, há três campos de entrada: "Email" com um ícone de envelope, "Senha" com um ícone de cadeado, e "Confirmar Senha" com um ícone de cadeado. Um botão azul "Trocar a senha" está na base do formulário.

FONTE: AUTORIA PRÓPRIA, 2025

A figura 59 mostra um formulário de troca de senha, onde o usuário precisa inserir seu *e-mail*, uma nova senha e confirmar essa nova senha para concluir o processo. Cada campo é acompanhado por um ícone ilustrativo — um envelope para o *e-mail* e cadeados para os campos de senha.

No topo da tela, há uma seta para voltar à página anterior e o título “Trocar Senha” centralizado, indicando claramente a função da tela. Ao final do formulário, há um botão azul com o texto “Trocar a senha”, que representa a ação necessária para efetivar a alteração de senha.

A interface é simples e objetiva, facilitando o uso e minimizando erros na digitação das informações. Os ícones contribuem para a clareza dos campos, ajudando o usuário a identificar rapidamente o que deve ser preenchido em cada

espaço. Além disso, o uso de campos separados para a senha e sua confirmação reforça a segurança, prevenindo erros na atualização. A seta no canto superior esquerdo permite uma navegação intuitiva, possibilitando o retorno fácil à tela anterior sem perder as informações já preenchidas.

Figura 60 - Tela de códigos da redefinição de Senha

```

1  |?php
2  session_start();
3
4  $host = "localhost";
5  $usuario = "u357936358_librinha";
6  $senha = "Librinh4tcc#";
7  $banco = "u357936358_scanhands";
8
9  $conn = new mysqli($host, $usuario, $senha, $banco);
10 if ($conn->connect_error) {
11     die(json_encode(["status" => "erro", "mensagem" => "Erro ao conectar com o banco de dados."]));
12 }
13
14 $email = $_SESSION['recupera_email'] ?? '';
15 $novaSenha = $_POST['senha'] ?? '';
16
17 if (!$email || !$novaSenha) {
18     echo json_encode(["status" => "erro", "mensagem" => "Dados inválidos."]);
19     exit;
20 }
21
22 // Atualiza a senha
23 $stmt = $conn->prepare("UPDATE usuario SET senha = ? WHERE email = ?");
24 $stmt->bind_param("ss", $novaSenha, $email);
25
26 if ($stmt->execute()) {
27     echo json_encode(["status" => "ok", "mensagem" => "Senha atualizada com sucesso."]);
28 } else {
29     echo json_encode(["status" => "erro", "mensagem" => "Erro ao atualizar a senha."]);
30 }
31
32 $conn->close();
33 ?>
34

```

FONTE: AUTORIA PRÓPRIA, 2025

Na figura 60, a porção de código apresentada realiza o processo de redefinição de senha de um usuário previamente cadastrado no sistema, utilizando sessões, validação de dados e atualização no banco de dados MySQL.

Primeiramente, o sistema inicia uma sessão com a função “*session_start()*”, permitindo o acesso à variável de sessão “*\$_SESSION[“email”]*”, que contém o *e-mail* do usuário em processo de recuperação. Em seguida, é estabelecida a conexão com o banco de dados por meio da classe “*mysqli*”, utilizando os parâmetros “*localhost*”, “*root*”, “” (senha em branco) e “*tcc*” como nome do banco.

Logo após o sistema captura a nova senha enviada pelo formulário através da variável “*\$_POST[“newPassword”]*”, e recupera o *e-mail* da sessão. Ambos os dados são armazenados nas variáveis “*\$newPassword*” e “*\$email*”, respectivamente. O sistema então verifica se esses valores estão vazios utilizando a função “*empty()*”.

Caso algum dos campos esteja ausente, o *script* retorna uma resposta em formato JSON com `“echo json_encode(array[“success” => false, “message” => “Dados inválidos”]);”`, e a execução é encerrada com `“exit”`.

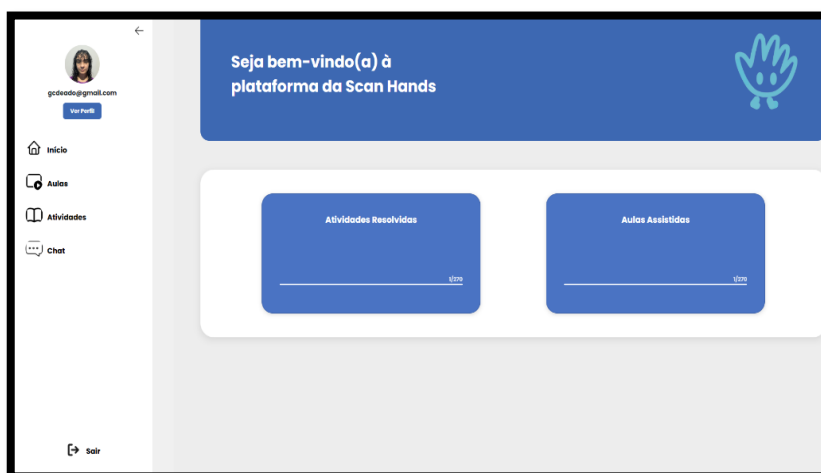
Se os dados forem válidos, o sistema prepara uma instrução SQL para atualizar a senha do usuário no banco de dados. Isso é feito com a função `“$conn->prepare ()”`, que recebe a *query* `“UPDATE usuarios SET senha =? WHERE email =?”`. Em seguida, os parâmetros são vinculados à instrução com `“$stmt->bind_param”` (`“ss”, “$newPassword”, “$email”`), indicando que ambos os valores são do tipo `“string”`. A execução da atualização é feita com `“$stmt->execute ()”`.

Caso a execução seja bem-sucedida, o sistema retorna uma resposta JSON com a mensagem `“Senha atualizada com sucesso”`. Se houver falha, a resposta informará `“Erro ao atualizar a senha”`. Por fim, a conexão com o banco de dados é encerrada com `“$conn->close()”`.

Apesar de o código funcionar corretamente, é importante destacar que a senha está sendo armazenada diretamente no banco de dados sem criptografia. Isso representa uma vulnerabilidade crítica. Recomenda-se o uso da função `“password_hash(“$newPassword”, “PASSWORD_DEFAULT”)”` para criptografar a senha antes de armazená-la, substituindo `“$newPassword”` por `“$hashedPassword”` na instrução SQL. Essa prática garante maior segurança e conformidade com os princípios de proteção de dados.

3.5.2 – Tela inicial do Aluno

Figura 61 - Tela inicial do aluno



FONTE: AUTORIA PRÓPRIA, 2025

A imagem 61 apresenta a tela inicial do aluno, que funciona como um painel de acompanhamento dentro da plataforma. Nela, o usuário pode visualizar informações resumidas do seu desempenho, como atividades resolvidas, aulas assistidas, capítulos concluídos e o tempo de tela utilizado. A interface é organizada de forma clara e intuitiva, com uma barra lateral à esquerda contendo o menu de navegação, incluindo opções como Início, Aulas, Atividades e *Chat*, facilitando o acesso rápido a diferentes funcionalidades.

No topo da tela, uma mensagem de boas-vindas personalizada reforça a identidade do usuário na plataforma, enquanto o logotipo da SCAN HANDS destaca a marca do sistema. Os painéis centrais apresentam dados visualmente simplificados, que permitem ao aluno acompanhar seu progresso de forma rápida e eficaz. A utilização de cores suaves e elementos gráficos contribui para uma experiência agradável e motivadora, incentivando o engajamento contínuo do usuário com as atividades educacionais.

Figura 62 - Tela de código *home* do aluno

```

} // Contar atividades feitas
} $qAtv = $conn->prepare("SELECT COUNT(*) as total FROM usuario_atividade_feita WHERE id_usuario = ?");
} $qAtv->bind_param("i", $idUser);
} $qAtv->execute();
} $resAtv = $qAtv->get_result()->fetch_assoc();
} $atividadesFeitas = $resAtv['total'] ?? 0;
} $qAtv->close();
}
} // Contar aulas assistidas
} $qAulas = $conn->prepare("SELECT COUNT(*) as total FROM usuario_aula_assistida WHERE id_usuario = ?");
} $qAulas->bind_param("i", $idUser);
} $qAulas->execute();
} $resAula = $qAulas->get_result()->fetch_assoc();
} $aulasAssistidas = $resAula['total'] ?? 0;
} $qAulas->close();
}
} // Totais fixos (ou você pode pegar de uma tabela aula/atividade)
} $totalAtividades = 270;
} $totalAulas = 270;
}
} $conn->close();
} ?>

```

FONTE: AUTORIA PRÓPRIA, 2025

O funcionamento dessa tela é garantido pelo código exibido na figura 62. Nele, são realizadas consultas ao banco de dados para recuperar as informações relacionadas ao progresso do aluno. Primeiramente, o código conta o total de atividades finalizadas pelo usuário por meio da query SQL *SELECT COUNT(*) as total FROM usuario_atividade_feita WHERE id_usuario = ?*.

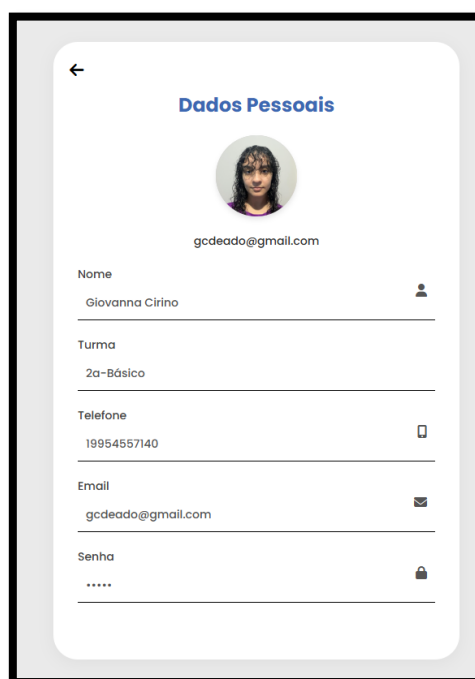
Essa instrução retorna à quantidade de registros associados ao aluno logado, armazenando o valor na variável \$resAtv. Em seguida, outra consulta semelhante (*SELECT COUNT(*) as total FROM usuario_aula_assistida WHERE id_usuario = ?*) obtém o número de aulas assistidas pelo mesmo usuário, cujo resultado é guardado em \$resAulas.

Esses valores são, então, tratados no PHP e vinculados aos elementos gráficos da interface, exibindo-os nos blocos azuis da tela (“Atividades Resolvidas” e “Aulas Assistidas”). O código também define variáveis de totais gerais — por exemplo, \$totalAtv = 270 e \$totalAulas = 270 — que representam a quantidade máxima de atividades e aulas disponíveis no sistema.

Com isso, o sistema pode comparar o progresso do aluno com o total existente, permitindo a geração de indicadores percentuais e barras de progresso exibidas na interface. Portanto, a lógica do código é responsável por conectar os dados armazenados no banco de dados ao painel visual do aluno, transformando registros em informações acessíveis e organizadas para o acompanhamento de seu desempenho.

3.5.2.1 – Tela de visualizar perfil do aluno

Figura 63 – Tela visualizar perfil



FONTE: AUTORIA PRÓPRIA, 2025

A imagem 63 apresenta a tela de visualização do perfil do aluno, onde são exibidos os dados pessoais do usuário. Entre as informações disponíveis estão: nome, turma, telefone, *e-mail* e senha cadastrada. Essa tela possibilita ao estudante consultar rapidamente seus dados dentro da plataforma.

Além disso, a interface mostra uma foto do usuário no topo, que ajuda na identificação visual do perfil. Os campos são apresentados em um formato organizado e de fácil leitura, cada um acompanhado de um ícone ilustrativo que indica o tipo de informação exibida, como o ícone de pessoa para o nome, telefone e *e-mail*, e um cadeado para a senha, reforçando a segurança dos dados. A seta no canto superior esquerdo permite que o usuário retorne facilmente para a tela anterior, tornando a navegação mais fluida e intuitiva.

Figura 64 – Tela de Código visualizar perfil aluno

```

// Conecta ao banco
$host = "localhost";
$usuario = "u357936358_librinha";
$senha = "Librinh4tccc";
$banco = "u357936358_scanhands";

$conn = new mysqli($host, $usuario, $senha, $banco);
if ($conn->connect_error) {
    die("Erro de conexão com o banco: " . $conn->connect_error);
}

$email = $_SESSION['email'];
$nome = $telefone = $turma = $senhaUsuario = '';
$fotoBase64 = '';

// Busca dados do usuário + turma vinculada
$stmt = $conn->prepare("
    SELECT u.nome, u.telefone, u.senha, u.foto_perfil, t.nome AS turma_nome
    FROM usuario u
    JOIN aluno a ON u.id_usuario = a.id_usuario
    LEFT JOIN aluno_turma at ON a.id_aluno = at.id_aluno
    LEFT JOIN turma t ON at.id_turma = t.id_turma
    WHERE u.email = ?
");
$stmt->bind_param("s", $email);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    $dados = $result->fetch_assoc();
    $nome = $dados['nome'];
    $telefone = $dados['telefone'];
    $senhaUsuario = $dados['senha'];
    $turma = $dados['turma_nome'] ?? 'Sem turma';
    $fotoBase64 = !empty($dados['foto_perfil']) ? 'data:image/jpeg;base64,' . base64_encode($dados['foto_perfil']) : '';
} else {
    $nome = "Desconhecido";
    $turma = "Não encontrado";
}

$conn->close();
?>

```

FONTE: AUTORIA PRÓPRIA, 2025

O código apresentado na figura 64 tem como função recuperar e exibir as informações pessoais do aluno cadastradas no banco de dados. O processo inicia com a conexão ao banco de dados MySQL, utilizando os parâmetros de *host*, usuário, senha e nome do banco. Caso a conexão falhe, uma mensagem de erro é exibida interrompendo a execução. Em seguida, o sistema realiza uma consulta SQL para buscar os dados do aluno. Nessa consulta, são recuperados o nome, telefone, *e-mail*,

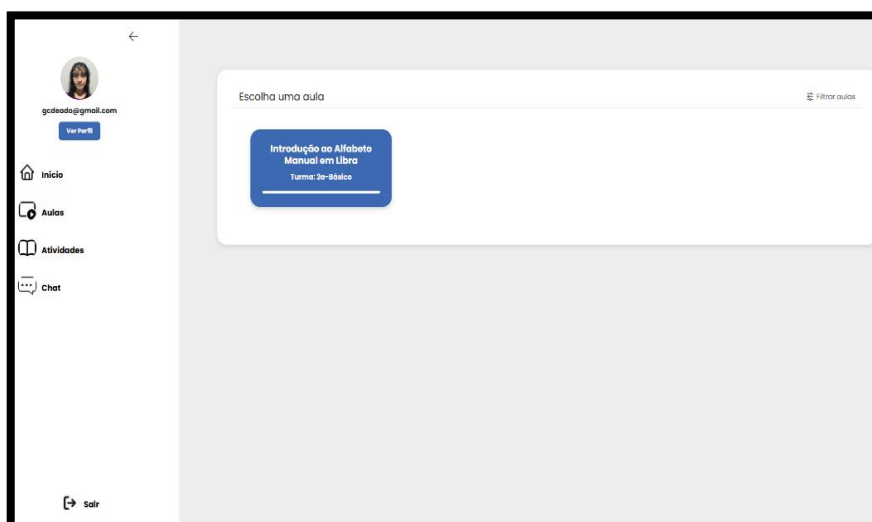
senha, turma e a foto de perfil. Para isso, utiliza-se um *JOIN* entre as tabelas de aluno e turma, garantindo que seja retornado também o nome da turma à qual o aluno está vinculado.

Após a execução da consulta, o sistema verifica se os resultados foram encontrados. Caso positivo, os dados retornados são atribuídos a variáveis específicas, como *\$nomeUsuario*, *\$telefone*, *\$turmaNome*, *\$emailUsuario*, entre outras. Além disso, o campo da foto de perfil é convertido para o formato *base64*, permitindo que a imagem seja exibida diretamente na interface da aplicação sem a necessidade de salvar em um arquivo separado.

Se não houver dados correspondentes, são exibidas mensagens padrão, como “Desconectado” para o *e-mail* ou “Não encontrado” para a turma. Por fim, a conexão com o banco de dados é encerrada. Portanto, esse código garante que, ao acessar a tela de Visualizar Perfil do Aluno, o sistema traga automaticamente as informações salvas no banco, exibindo ao usuário seus dados pessoais de forma organizada e acessível.

3.5.2.2 – Tela de aulas do aluno

Figura 65 - Tela de aulas do aluno



FONTE: AUTORIA PRÓPRIA, 2025

Essa é a tela de aulas do aluno, representada na figura 65, onde o estudante pode visualizar os conteúdos organizados em blocos temáticos. Cada aula apresenta o título e o *status* de progresso, permitindo acompanhar o andamento das aulas. A

interface conta também com a opção de filtro, acessível por meio do botão cinza “Filtrar aulas”, facilitando a busca por conteúdos específicos dentro do curso.

No lado esquerdo da tela, está presente o menu de navegação, com ícones que dão acesso rápido às principais áreas da plataforma, como Início, Aulas, Atividades e *Chat*. A foto e o *e-mail* do usuário são exibidos no topo desse menu, reforçando a personalização e a identidade do estudante dentro do sistema. A organização clara e a simplicidade do *design* proporcionam uma experiência amigável e eficiente para o acompanhamento dos estudos.

Figura 66 – Tela de Código aulas aluno

```
// Buscar atvs relacionadas às turmas do aluno logado
$sql = "SELECT
    atv.id_atv,
    atv.nomeAtv AS titulo,
    t.nome AS turma
FROM atividade atv
INNER JOIN atividade_turma att
    ON atv.id_atv = att.id_atv
INNER JOIN turma t
    ON att.id_turma = t.id_turma
INNER JOIN aluno_turma alt
    ON t.id_turma = alt.id_turma
INNER JOIN aluno al
    ON alt.id_aluno = al.id_aluno
INNER JOIN usuario u
    ON al.id_usuario = u.id_usuario
WHERE u.email = ?";

$stmt = $conn->prepare($sql);
$types = str_repeat("s", count($params));
$stmt->bind_param($types, ...$params);
$stmt->execute(); $result = $stmt->get_result();

$atividades = []; // ✅ nome certo
while ($row = $result->fetch_assoc()) {
    $atividades[] = $row;
}
$stmt->close();
$conn->close();
?>
<!DOCTYPE html>
```

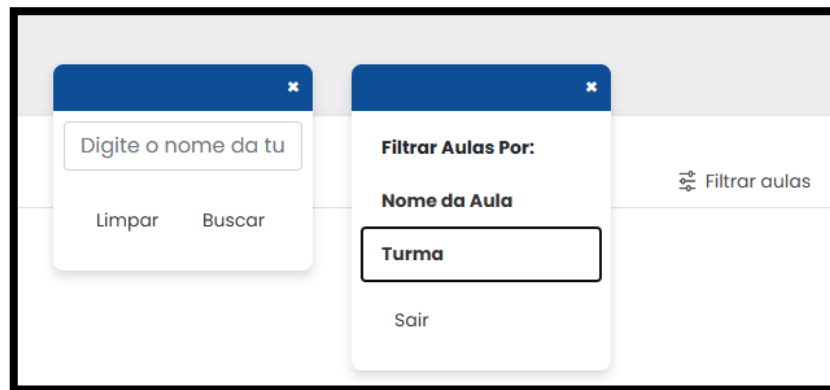
FONTE: AUTORIA PRÓPRIA, 2025

O código apresentado na figura 66 é responsável por buscar e exibir as aulas relacionadas ao aluno logado na plataforma. Para isso, é utilizada uma consulta SQL que realiza múltiplos *JOINS* entre diferentes tabelas do banco de dados, relacionando atividades, turmas e usuários. A *query* seleciona o ID da atividade, o título da aula (nomeAtv) e o nome da turma vinculada, garantindo que sejam recuperadas apenas as aulas associadas ao aluno autenticado, filtradas pelo *e-mail* do usuário logado. O comando SQL é preparado com segurança através de *prepare ()* e *bind_param ()*, evitando falhas de execução e protegendo contra injeção de SQL.

Em seguida, os dados são executados e retornados com `get_result()`. Os resultados são percorridos em um laço `while`, onde cada linha retornada pela consulta (`fetch_assoc()`) é armazenada no `array $atividades[]`. Esse `array` reúne todas as aulas vinculadas ao estudante, possibilitando que sejam exibidas na interface, conforme mostrado na tela. Por fim, o código encerra a execução fechando tanto o `statement` quanto a conexão com o banco de dados.

Dessa forma, esse código garante que, ao acessar a Tela de Aulas do Aluno, sejam exibidos de maneira organizada os títulos das aulas vinculadas à turma do estudante, permitindo acompanhar o progresso e utilizar o recurso de filtragem oferecido pela interface.

Figura 67 – Tela filtrar aulas aluno



FONTE: AUTORIA PRÓPRIA, 2025

Nessa tela o usuário pode encontrar determinada atividade filtrando-a por nome ou categoria, como mostra a figura 67. O sistema oferece um campo para digitar o nome da aula, permitindo uma busca direta e rápida. Além disso, há opções para limpar a busca ou iniciar a pesquisa, facilitando o controle do usuário sobre os filtros aplicados.

Também é possível filtrar as aulas por categorias específicas, como "Turma", conforme ilustrado, o que ajuda a refinar os resultados e localizar conteúdos relacionados ao perfil ou à série do aluno. A interface do filtro é simples e acessível, com botões de ação claros e menus suspensos que melhoram a navegação e a usabilidade da plataforma.

Figura 68 – Tela de Código filtrar aulas aluno

```

/* =====
LIMPAR FILTROS
===== */
if (!empty($_GET['limpar'])) {
    header("Location: atividadesProfessor.php");
    exit();
}

/* =====
FILTROS (nome da atividade, nome da turma)
===== */
$conditions = ["pt.id_professor = ?"];
$params     = [$idProfessor];
$types     = "i";

if (!empty($_GET['nome'])) {
    $conditions[] = "atv.nomeAtv LIKE ?";
    $params[]    = "%" . $_GET['nome'] . "%";
    $types      .= "s";
}

if (!empty($_GET['turma'])) {
    $conditions[] = "t.nome LIKE ?";
    $params[]    = "%" . $_GET['turma'] . "%";
    $types      .= "s";
}

$whereClause = implode(" AND ", $conditions);

```

FONTE: AUTORIA PRÓPRIA, 2025

O código apresentado na figura 68 tem como objetivo permitir que o aluno filtre as aulas disponíveis na plataforma, de acordo com critérios como nome da atividade ou nome da turma. Primeiramente, há uma verificação para identificar se o botão de “limpar filtros” foi acionado. Caso isso ocorra, o sistema redireciona o usuário para a página de atividades (atividadesProfessor.php), removendo todos os parâmetros de busca previamente aplicados. Na sequência, são definidas as condições dinâmicas de filtragem. Por padrão, o código já traz a condição obrigatória que relaciona o professor (idProfessor).

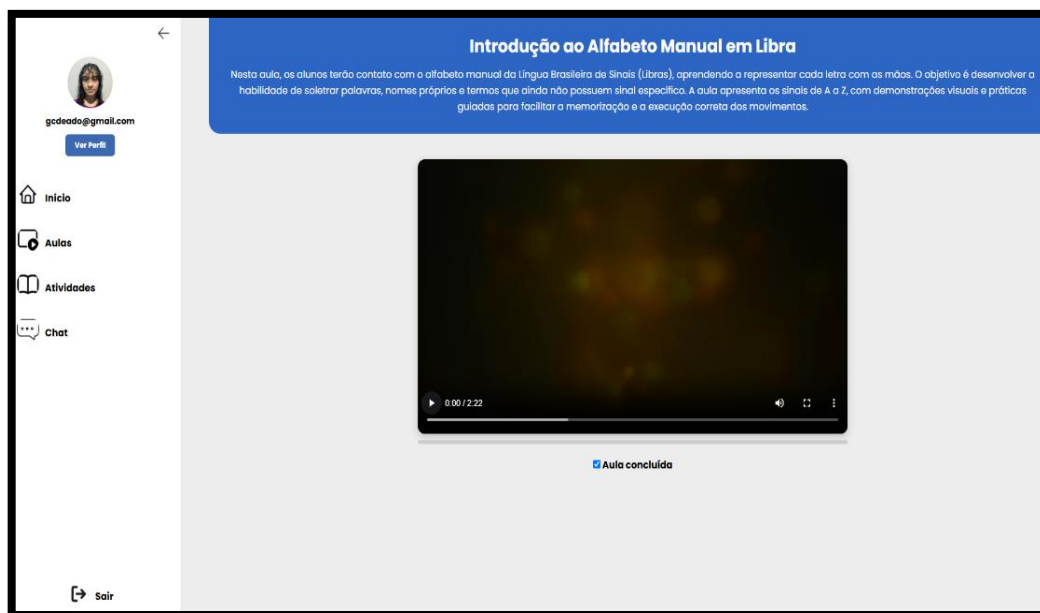
Além disso, se o usuário preencher o campo de nome da atividade, é adicionada uma cláusula *LIKE* que busca correspondências parciais no banco de dados. O mesmo processo é aplicado para o nome da turma, permitindo localizar aulas específicas vinculadas a determinada categoria. Essas condições são acumuladas no *array* *\$conditions* e, ao final, concatenadas em uma única cláusula com o comando *implode(" AND ", \$conditions)*.

Assim, o sistema gera dinamicamente a cláusula *WHERE* da consulta SQL de acordo com os filtros escolhidos pelo usuário. Dessa forma, o código garante que o recurso de filtro de aulas funcione de forma flexível e eficiente, permitindo ao

estudante encontrar rapidamente os conteúdos desejados, seja pelo título da atividade ou pela turma em que está inserido.

3.5.2.3 – Tela aula selecionada do aluno

Figura 69 – Tela de Aula do aluno



FONTE: AUTORIA PRÓPRIA, 2025

Quando o usuário selecionar uma aula, ele será direcionado a um vídeo aula sobre o tema escolhido. Na figura 69, o tema é “Introdução ao Alfabeto Manual em Libras”, acompanhado por uma breve descrição que explica o conteúdo que será abordado, incluindo o aprendizado e o reconhecimento dos sinais básicos do alfabeto.

A descrição também destaca os objetivos da aula, como o desenvolvimento da habilidade de soletrar palavras e demonstrar os sinais com precisão, além de apresentar exemplos visuais e práticos para facilitar o entendimento. A interface da plataforma exibe o vídeo em destaque, permitindo que o usuário acompanhe o conteúdo de forma clara e acessível.

Abaixo do vídeo, há uma indicação visual para marcar a aula como concluída, incentivando o acompanhamento do progresso do aluno dentro do curso. A navegação mantém a barra lateral visível, facilitando o acesso a outras áreas da plataforma sem perder o foco no conteúdo principal.

Figura 70 – Tela de Código aula do aluno

```

// Buscar dados da aula
$stmtAula = $conn->prepare("SELECT nomeAula, descricaoAula FROM aula WHERE id_aula = ?");
$stmtAula->bind_param("i", $idAula);
$stmtAula->execute();
$resultAula = $stmtAula->get_result();

$tituloAula = "Aula";
$descricaoAula = "";

if ($resultAula && $resultAula->num_rows > 0) {
    $aula = $resultAula->fetch_assoc();
    $tituloAula = $aula['nomeAula'];
    $descricaoAula = $aula['descricaoAula'];
}
$stmtAula->close();
$conn->close();
?>

```

FONTE: AUTORIA PRÓPRIA, 2025

Essa tela é responsável por exibir a aula escolhida pelo estudante. No exemplo apresentado na figura 70, o aluno selecionou a aula “Introdução à Libras”, que é carregada juntamente com uma breve descrição do conteúdo a ser aprendido. No código, é feita uma consulta ao banco de dados para buscar as informações da aula com base no seu identificador (*id_aula*).

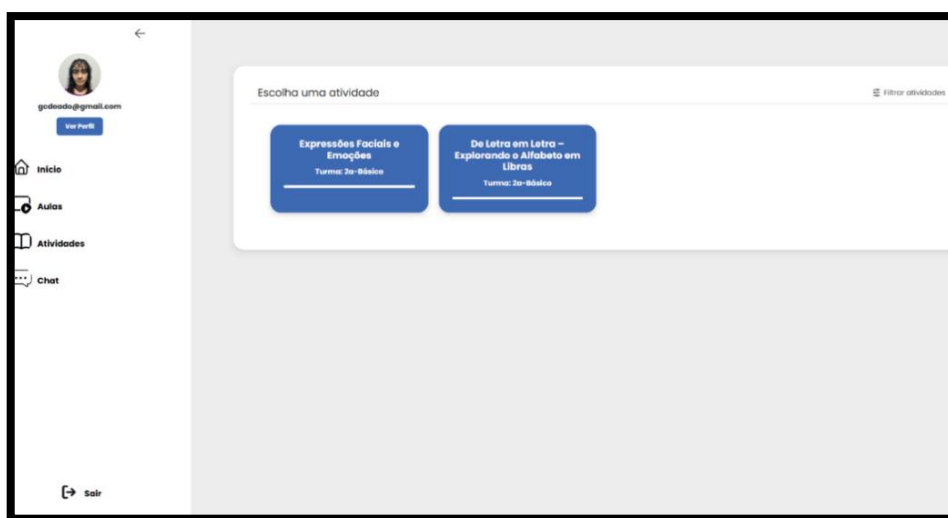
O comando SQL utilizado (*SELECT nomeAula, descricaoAula FROM aula WHERE id_aula = ?*) retorna o nome da aula e a descrição associada. O *bind_param* garante a segurança contra injeções de SQL ao associar o valor do *id_aula* ao parâmetro da consulta. Após a execução (*execute*), o resultado é recuperado com *get_result()*. Se a consulta retornar registros (*num_rows > 0*), os dados são extraídos (*fetch_assoc*) e armazenados em variáveis, que depois são enviadas para a interface. Assim, a tela mostra ao usuário o título da aula e a explicação do tema, junto ao *player* de vídeo exibido na interface. Com esse funcionamento, o sistema assegura que, ao clicar em uma aula, o aluno seja direcionado para o conteúdo correto, visualizando tanto o vídeo quanto a descrição correspondente ao tema selecionado.

Além disso, o uso do método *prepare* juntamente com *bind_param* não só melhora a segurança do sistema contra ataques de injeção de SQL, mas também otimiza a execução das consultas ao banco de dados, permitindo que o mesmo comando seja reutilizado com diferentes parâmetros de forma eficiente. O tratamento condicional da resposta com a verificação de *num_rows* garante que o sistema lide adequadamente com situações onde a aula solicitada não exista, evitando erros na interface do usuário. Por fim, ao fechar tanto a declaração preparada (*\$stmtAula->close()*) quanto a conexão com o banco (*\$conn->close()*), o código assegura uma

boa prática de gerenciamento de recursos, liberando memória e conexões que não serão mais usadas, o que contribui para a estabilidade e desempenho da aplicação como um todo.

3.5.2.4 - Tela atividades do aluno

Figura 71 – Tela atividades do aluno



FONTE: AUTORIA PRÓPRIA, 2025

Nessa tela, representada na figura 71, o aluno visualiza todas as atividades disponíveis relacionadas ao conteúdo do Capítulo 1. Cada atividade é exibida em um formato de cartão com destaque visual, contendo o título da atividade, como “Expressões Faciais e Emoções” e “De Letra em Letra – Explorando o Alfabeto em Libras”, e informações adicionais como a turma associada, neste caso a turma 2º Básico. Essa organização facilita a identificação rápida das atividades, tornando o processo de seleção mais intuitivo e eficiente para o usuário. Além disso, a interface inclui uma funcionalidade de filtro localizada no canto superior direito, permitindo que o aluno refine a lista de atividades exibidas com base em critérios como nome da atividade ou categoria, tornando a navegação ainda mais prática e personalizada conforme as necessidades do estudante.

A barra lateral esquerda mantém o perfil do usuário com o *e-mail* e uma opção para ver o perfil completo, além de oferecer acesso rápido às principais áreas do sistema: Início, Aulas, Atividades e *Chat*. Isso proporciona uma experiência de navegação fluida, onde o aluno pode alternar facilmente entre diferentes módulos sem

perder o foco no aprendizado. Também há um botão para sair do sistema, garantindo uma forma rápida e segura de encerrar a sessão.

A tela prioriza a usabilidade e a clareza, apresentando as informações essenciais de forma limpa e organizada, o que contribui para que o aluno se sinta confortável ao interagir com a plataforma. Essa estrutura não só facilita o acesso aos conteúdos, mas também incentiva o engajamento com as atividades, promovendo um ambiente de aprendizado mais eficiente e agradável.

Figura 72 – Tela Código atividades do aluno

```
// Buscar atvs relacionadas às turmas do aluno logado
$sql = "SELECT
    atv.id_atv,
    atv.nomeAtv AS titulo,
    t.nome AS turma
FROM atividade atv
INNER JOIN atividade_turma att
    ON atv.id_atv = att.id_atv
INNER JOIN turma t
    ON att.id_turma = t.id_turma
INNER JOIN aluno_turma alt
    ON t.id_turma = alt.id_turma
INNER JOIN aluno al
    ON alt.id_aluno = al.id_aluno
INNER JOIN usuario u
    ON al.id_usuario = u.id_usuario
WHERE u.email = ?";

$stmt = $conn->prepare($sql);
$types = str_repeat("s", count($params));
$stmt->bind_param($types, ...$params);
$stmt->execute(); $result = $stmt->get_result();

$atividades = []; //  nome certo
while ($row = $result->fetch_assoc()) {
    $atividades[] = $row;
}
$stmt->close();
$conn->close();
?>
```

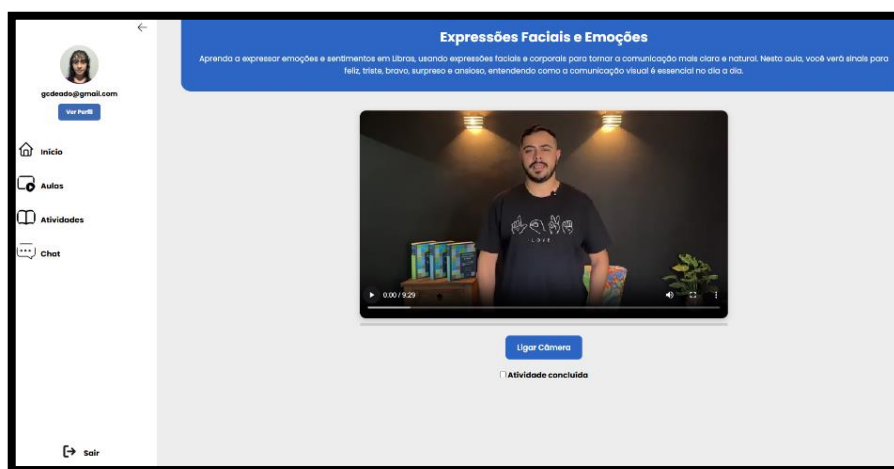
FONTE: AUTORIA PRÓPRIA, 2025

O código apresentado na figura 72 realiza a busca das atividades associadas ao aluno logado. Para isso, a consulta SQL utiliza diversos *INNER JOINS* entre as tabelas atividade, atividade_turma, turma, aluno_turma, aluno e usuario, garantindo que apenas as atividades da turma do estudante conectado sejam exibidas. O filtro é feito através do *WHERE u.email = ?*, que limita os resultados ao usuário autenticado. A *query* é preparada com *prepare()* e os parâmetros são vinculados com *bind_param()*, o que aumenta a segurança contra injeções de SQL.

Em seguida, `execute()` executa a consulta e `get_result()` recupera os dados. O `loop while` percorre cada resultado retornado pelo banco (`fetch_assoc()`) e adiciona as informações ao `array $atividades[]`, permitindo que elas sejam exibidas dinamicamente na tela. Assim, o sistema garante que cada aluno visualize apenas as atividades relacionadas à sua turma e acompanhe seu progresso dentro da plataforma.

3.5.2.5 - Tela da atividade sendo executada do aluno

Figura 73 – Tela da atividade sendo executada do aluno



FONTE: AUTORIA PRÓPRIA, 2025

A figura 73 apresenta a tela após o usuário selecionar uma atividade, ele é direcionado para a tela do vídeo aula correspondente ao tema escolhido, que neste exemplo é “Expressões Faciais e Emoções”. No topo da tela, encontra-se o título da atividade juntamente com uma breve descrição que explica o objetivo da aula, destacando o aprendizado de expressar emoções e sentimentos em Libras por meio de expressões faciais e corporais. A interface exibe o vídeo com o instrutor, que guiará o aluno pelo conteúdo, facilitando a compreensão visual e prática dos conceitos apresentados.

Além do vídeo, a tela disponibiliza um botão para “Ligar Câmera”, incentivando o aluno a ativar seu dispositivo para praticar os movimentos e sinais ensinados, promovendo um aprendizado mais interativo e dinâmico. A opção de marcar a atividade como concluída está presente logo abaixo do botão, permitindo que o aluno acompanhe seu progresso dentro do sistema. A navegação continua intuitiva, com a barra lateral mantendo o acesso rápido a outras seções como Início,

Aulas, Atividades e *Chat*, além do perfil do usuário, oferecendo uma experiência integrada e fluida ao longo do uso da plataforma.

Figura 74 – Tela de Código atividade sendo executada do aluno

```
// Buscar dados da atividade
$stmtAtv = $conn->prepare("SELECT nomeAtv, descricaoAtv FROM atividade WHERE id_atv = ?");
$stmtAtv->bind_param("i", $idAtividade);
$stmtAtv->execute();
$resultAtv = $stmtAtv->get_result();

$tituloAtividade = "Atividade";
$descricaoAtividade = "";

if ($resultAtv && $resultAtv->num_rows > 0) {
    $atividade = $resultAtv->fetch_assoc();
    $tituloAtividade = $atividade['nomeAtv'];
    $descricaoAtividade = $atividade['descricaoAtv'];
}
$stmtAtv->close();
$conn->close();
?>
```

FONTE: AUTORIA PRÓPRIA, 2025

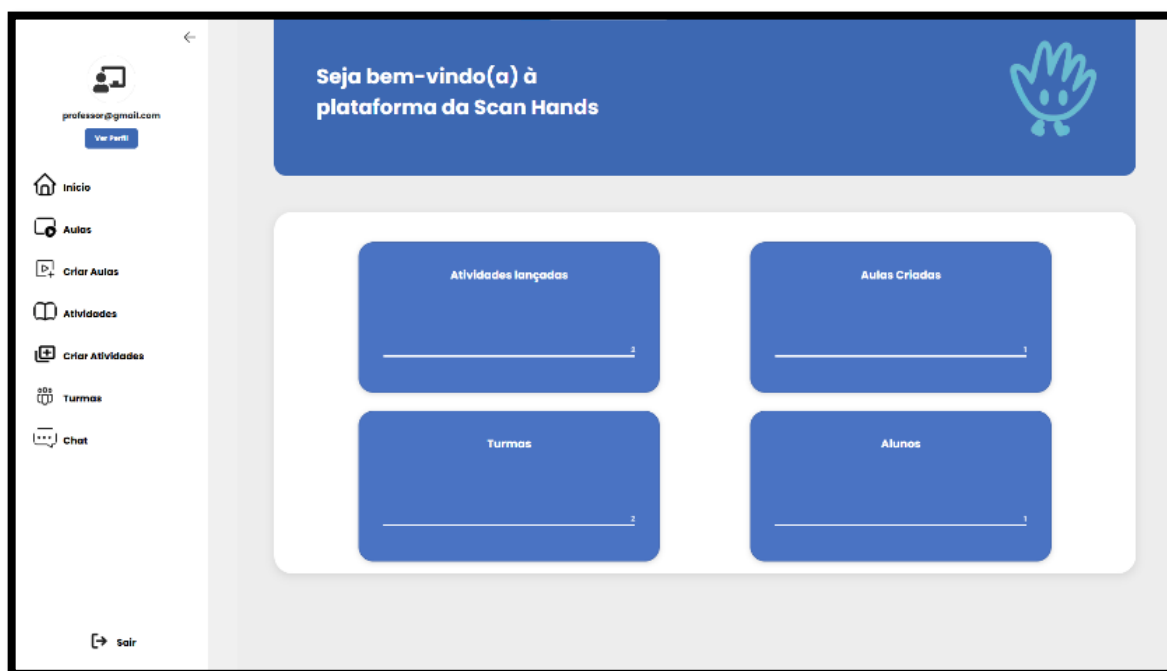
Este código representado na figura 74 em PHP é responsável por buscar os dados de uma atividade específica no banco de dados, utilizando uma consulta preparada (*prepared statement*) para garantir segurança e evitar injeções de SQL. Primeiramente, é feita a preparação da consulta SQL com *prepare()*, onde se seleciona o nome (*nomeAtv*) e a descrição (*descricaoAtv*) da atividade a partir da tabela *atividade*, com base no identificador *id_atv*, passado como parâmetro. Em seguida, o código vincula esse parâmetro à variável *\$idAtividade* utilizando *bind_param()*, executa a consulta com *execute()* e obtém os resultados através de *get_result()*.

Antes de tratar os dados retornados, são inicializadas duas variáveis padrão: *\$tituloAtividade*, com o valor "Atividade", e *\$descricaoAtividade*, como uma *string* vazia. Isso serve como *fallback*, caso não sejam encontrados dados no banco. Se o resultado da consulta for válido e houver ao menos uma linha retornada (*\$resultAtv->num_rows > 0*), os dados da atividade são extraídos com *fetch_assoc()*, e os valores de *nomeAtv* e *descricaoAtv* são armazenados nas variáveis correspondentes, substituindo os valores padrão definidos anteriormente. Por fim, tanto o *statement* quanto a conexão com o banco de dados são encerrados com *close()*, garantindo a liberação dos recursos utilizados.

Em resumo, o código realiza a busca segura de informações de uma atividade específica, trata o retorno de forma apropriada e garante o fechamento adequado dos recursos após a execução.

3.5.3 – Tela inicial do Professor

Figura 75 - Tela inicial do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Essa é a tela inicial do professor, representada na figura 75, que apresenta um resumo das principais informações da plataforma de forma clara e organizada. Nela, o professor pode visualizar rapidamente a quantidade de atividades lançadas, aulas criadas, turmas cadastradas e alunos vinculados ao seu perfil. Esses dados são exibidos em cartões destacados e distribuídos de maneira equilibrada, facilitando o acompanhamento do progresso e o gerenciamento das tarefas. Além dos indicadores numéricos, a interface mantém uma barra lateral com acesso fácil às principais funcionalidades, como criação e gerenciamento de aulas, atividades, turmas e *chat*, proporcionando uma navegação ágil e intuitiva.

O cabeçalho superior ainda dá as boas-vindas ao usuário, reforçando a identidade da plataforma SCAN HANDS e promovendo uma experiência mais acolhedora. Essa estrutura permite que o professor tenha uma visão geral do seu ambiente de trabalho na plataforma, tornando o processo de organização e

monitoramento das turmas mais eficiente e produtivo. A disposição visual contribui para a redução de tempo em tarefas operacionais, permitindo foco no planejamento pedagógico. Além disso, a centralização das informações melhora a tomada de decisão e o acompanhamento do desempenho estudantil.

Figura 76 – Tela Código tela inicial do Professor

```

/** 3) Turmas do professor */
$qtDTurmas = scalarInt(
    $conn,
    "SELECT COUNT(DISTINCT id_turma) AS total FROM professor_turma WHERE id_professor = ?",
    "i",
    [$idProfessor]
);

/** 4) Atividades ligadas às turmas do professor */
// Tabelas conforme seus prints: atividade_turma (id_atv, id_turma) + professor_turma
if (tableExists($conn, $banco, "atividade_turma")) {
    $qtDAtividades = scalarInt(
        $conn,
        "SELECT COUNT(DISTINCT atv.id_atv) AS total
        FROM atividade_turma atv
        INNER JOIN professor_turma pt ON atv.id_turma = pt.id_turma
        WHERE pt.id_professor = ?",
        "i",
        [$idProfessor]
    );
} else {
    // fallback simples: conta todas as atividades (pior cenário)
    if (tableExists($conn, $banco, "atividade")) {
        $qtDAtividades = scalarInt($conn, "SELECT COUNT(*) AS total FROM atividade", "", []);
    }
}

/** 5) Aulas ligadas ao professor (tenta diferentes esquemas) */
// Preferência: tabela aula/aulas com id_turma -> join professor_turma
if (tableExists($conn, $banco, "aula")) {
    if (columnExists($conn, $banco, "aula", "id_turma")) {
        $qtDAulas = scalarInt(
            $conn,
            "SELECT COUNT(DISTINCT a.id_aula) AS total
            FROM aula a
            INNER JOIN professor_turma pt ON a.id_turma = pt.id_turma
            WHERE pt.id_professor = ?",
            "i",
            [$idProfessor]
        );
    } elseif (columnExists($conn, $banco, "aula", "id_professor")) {
        $qtDAulas = scalarInt(
            $conn,
            "SELECT COUNT(*) AS total FROM aula WHERE id_professor = ?",
            "i",
            [$idProfessor]
        );
    } elseif (tableExists($conn, $banco, "aula_turma")) {
        $qtDAulas = scalarInt(
            $conn,
            "SELECT COUNT(DISTINCT a.id_aula) AS total
            FROM aula a
            INNER JOIN aula_turma at ON a.id_aula = at.id_aula
            INNER JOIN professor_turma pt ON at.id_turma = pt.id_turma
            WHERE pt.id_professor = ?",
            "i",
            [$idProfessor]
        );
    }
}

```

FONTE: AUTORIA PRÓPRIA, 2025

O código responsável pela tela exibida na figura 76 realiza a contagem das informações principais diretamente no banco de dados, garantindo que os dados apresentados estejam sempre atualizados e precisos. Para isso, utiliza consultas SQL com a função *COUNT*, que soma os registros existentes em diferentes tabelas relacionadas ao professor logado.

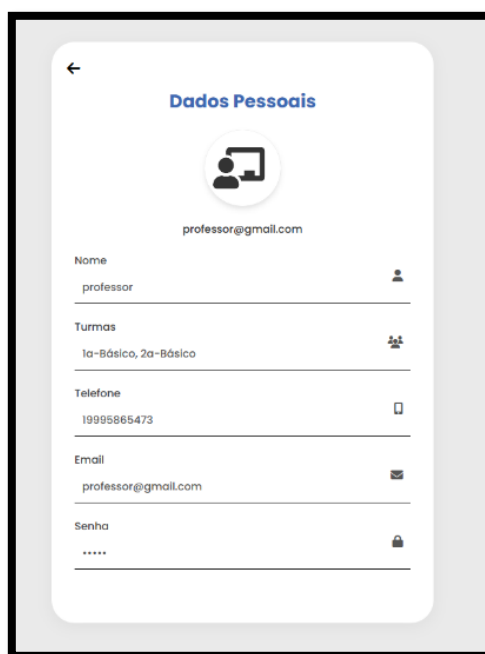
Primeiramente, o código conta o número de turmas vinculadas ao professor, consultando a tabela `professor_turma` e filtrando pelo identificador do professor. Em seguida, verifica a existência da tabela `atividade_turma` para contar as atividades associadas às turmas do professor. Caso essa tabela não esteja disponível, o código recorre a uma consulta direta na tabela `atividade`, assegurando uma contagem mesmo no pior cenário.

Para as aulas, a lógica é semelhante: o sistema verifica a estrutura do banco, identificando se a tabela `aula` contém a coluna `id_turma` ou `id_professor`, e realiza a contagem das aulas relacionadas ao professor com base nessas relações. Também é considerada a tabela intermediária `aula_turma` para garantir a associação correta entre aulas, turmas e professores.

Por fim, a quantidade de alunos vinculados às turmas do professor é obtida consultando a tabela `aluno_turma`, cruzando as informações com as turmas cadastradas. Dessa forma, a tela inicial do professor apresenta uma visão geral atualizada e confiável do seu ambiente acadêmico, refletindo em tempo real os dados extraídos do banco de dados, o que assegura eficiência na gestão das turmas, atividades e alunos.

3.5.3.1 - Tela de visualizar perfil do Professor

Figura 77 - Tela de visualizar perfil do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Essa é a tela de dados pessoais do professor, representada na figura 77, onde são exibidas e organizadas as principais informações do usuário. Nela, constam campos como nome, CPF, turmas em que ministra aulas, telefone, *e-mail* e senha, em que os caracteres digitados são mascarados (substituídos por asteriscos) para garantir a privacidade do usuário.

Figura 78 – Tela Código visualizar perfil do professor

```
<?php
session_start();

// Redireciona se o usuário não estiver Logado
if (!isset($_SESSION['email'])) {
    header("Location: ../../index.php");
    exit();
}

// Conecta ao banco
$host = "localhost";
$usuario = "u357936358_librinha";
$senha = "Librinhdttcc#";
$banco = "u357936358_scanhands";

$conn = new mysqli($host, $usuario, $senha, $banco);
if ($conn->connect_error) {
    die("Erro de conexão com o banco: " . $conn->connect_error);
}

$email = $_SESSION['email'];
$nome = $telefone = $turmas = $senhaUsuario = '';
$fotoBase64 = '';

// Busca dados do professor + turmas vinculadas
$stmt = $conn->prepare("
SELECT u.nome, u.telefone, u.senha, u.foto_perfil,
       COALESCE(GROUP_CONCAT(DISTINCT t.nome SEPARATOR ', '), 'Sem turmas') AS turmas
FROM usuario u
JOIN professor p ON u.id_usuario = p.id_usuario
LEFT JOIN professor_turma pt ON p.id_professor = pt.id_professor
LEFT JOIN turma t ON pt.id_turma = t.id_turma
WHERE u.email = ?
GROUP BY u.id_usuario
");
$stmt->bind_param("s", $email);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    $dados = $result->fetch_assoc();
    $nome = $dados['nome'];
    $telefone = $dados['telefone'];
    $senhaUsuario = $dados['senha'];
    $turmas = $dados['turmas'];
    $fotoBase64 = !empty($dados['foto_perfil']) ? 'data:image/jpeg;base64,' . base64_encode($dados['foto_perfil']) : '';
} else {
    $nome = "Desconhecido";
    $turmas = "Não encontrado";
}

$conn->close();
?>
```

FONTE: AUTORIA PRÓPRIA, 2025

O código exibido na figura 78, que dá suporte a essa tela começa iniciando a sessão com `session_start()`, garantindo que apenas usuários logados possam acessar a página. Caso o *e-mail* não esteja registrado na sessão, o sistema redireciona automaticamente para a tela de *login*. Em seguida, o *script* estabelece a conexão com o banco de dados utilizando as credenciais configuradas e armazena o *e-mail* do usuário ativo na variável `$email`. Com essa informação, é feita uma consulta ao banco para buscar os dados do professor.

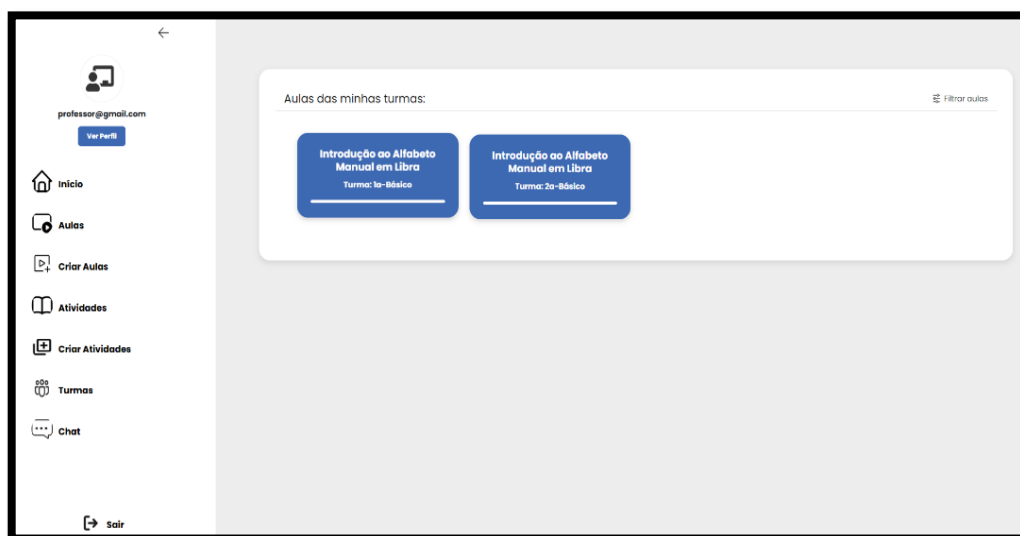
O *SELECT* traz informações como nome, telefone, senha, foto de perfil e, principalmente, as turmas vinculadas ao docente. A função *GROUP_CONCAT*

organiza todas as turmas em uma única linha, facilitando a exibição. Caso o professor não tenha turmas, o sistema retorna a mensagem "Sem turma". Após a execução da *query* com *prepare()* e *bind_param()*, os resultados são tratados: se existir pelo menos um registro, os dados são extraídos com *fetch_assoc()* e armazenados em variáveis como *\$nome*, *\$telefone*, *\$turmas* e *\$senhaUsuario*. Além disso, se houver foto de perfil cadastrada, ela é exibida utilizando *base64_encode*, permitindo renderização direta da imagem na tela.

Se nenhum dado for encontrado, a tela exibe a mensagem "Não encontrado". Por fim, a conexão com o banco é encerrada com *\$conn->close()*. Assim, a lógica do código garante que apenas o professor logado visualize suas próprias informações, protegendo dados sensíveis como senha e *e-mail*, além de permitir a visualização das turmas em que atua.

3.5.3.2 – Tela de aulas do Professor

Figura 79 - Tela de aulas do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Na tela de aulas, representada na figura 79, o professor tem acesso a todas as aulas cadastradas na plataforma de forma clara e organizada. As aulas são exibidas em cartões visualmente destacados, facilitando a navegação e o reconhecimento rápido dos conteúdos. Cada cartão apresenta o título da aula, como "Introdução a Libras", "Cumprimentos e Apresentações", "Alfabeto em Libras", entre outros temas essenciais para o aprendizado da disciplina. Além disso, cada cartão informa a qual turma aquela aula está vinculada, como "Turma: 1º-Básico" ou "Turma:

2º-Básico", possibilitando ao professor identificar facilmente para qual grupo de alunos aquela aula foi designada.

A interface também inclui a opção de filtrar as aulas por meio do botão cinza “Filtrar aulas”, localizado no canto superior direito da seção. Essa funcionalidade permite ao professor buscar por aulas específicas, facilitando a localização e o gerenciamento de conteúdos em turmas diferentes ou por critérios variados. Essa organização ajuda o professor a manter o controle sobre as aulas disponíveis e a otimizar o planejamento didático, tornando a experiência de uso da plataforma mais eficiente e intuitiva.

Dessa forma, a tela de aulas não apenas apresenta os conteúdos de maneira visualmente acessível, mas também oferece ferramentas que agilizam o acesso e a administração das aulas dentro do ambiente educacional digital.

Figura 80 – Tela Código aulas do Professor

```
// Montar SQL das aulas do professor
$sql = "SELECT
    a.id_aula,
    a.nomeAula AS titulo,
    t.nome AS turma,
    t.id_turma AS turma_id
FROM aula a
INNER JOIN aula_turma at ON a.id_aula = at.id_aula
INNER JOIN turma t ON at.id_turma = t.id_turma
INNER JOIN professor_turma pt ON pt.id_turma = t.id_turma
WHERE pt.id_professor = ?";

$params = [$idProfessor];
$types = "i";

// Filtros opcionais
if (!empty($_GET['nome'])) {
    $sql .= " AND a.nomeAula LIKE ?";
    $types .= "s";
    $params[] = "%" . $_GET['nome'] . "%";
}

if (!empty($_GET['turma'])) {
    $sql .= " AND t.nome LIKE ?";
    $types .= "s";
    $params[] = "%" . $_GET['turma'] . "%";
}

$stmt = $conn->prepare($sql);
$stmt->bind_param($types, ...$params);
$stmt->execute();
$result = $stmt->get_result();

$aulas = [];
while ($row = $result->fetch_assoc()) {
    $aulas[] = $row;
}
$stmt->close();
$conn->close();
?>
```

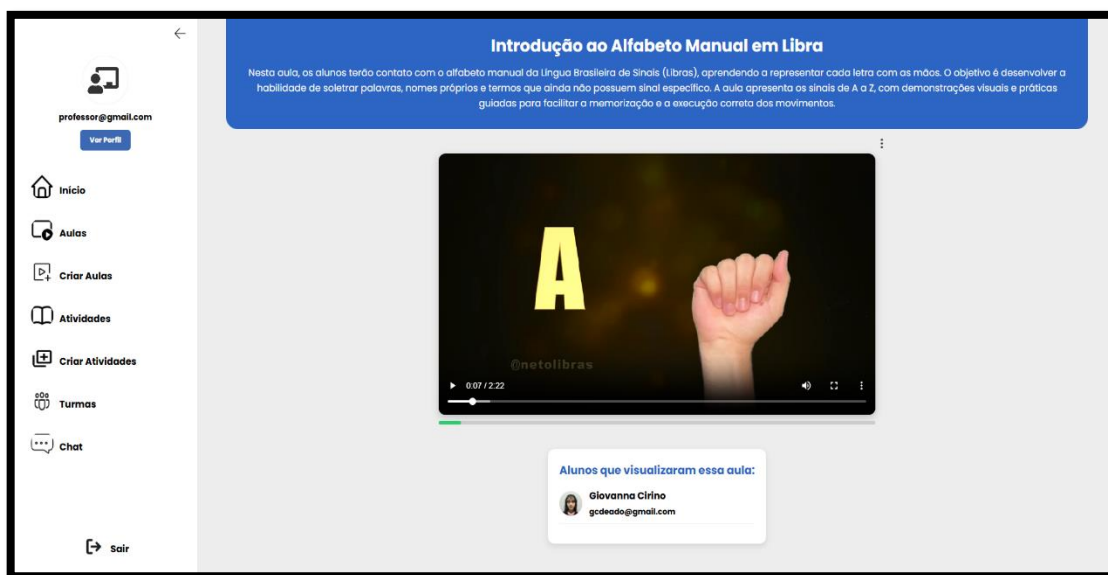
FONTE: AUTORIA PRÓPRIA, 2025

O funcionamento dessa tela é sustentado pelo código exibido na figura 80 em PHP, que realiza a busca das aulas associadas ao professor logado. O comando SQL seleciona o ID e o nome da aula, além da turma vinculada, através de junções entre as tabelas aula, aula_turma, turma e professor_turma. O critério *WHERE* `pt.id_professor = ?` garante que apenas as aulas do professor logado sejam carregadas. O código também contempla filtros opcionais: caso o usuário insira dados de pesquisa, como parte do nome da aula (*nome*) ou da turma (*turma*), a *query* é adaptada utilizando *LIKE*, possibilitando buscas mais flexíveis.

Os parâmetros são armazenados dinamicamente em *arrays* e aplicados na consulta com segurança, utilizando *prepare()* e *bind_param()*. Após a execução da *query*, os resultados são percorridos em um laço *while*, no qual cada linha retornada é armazenada no *array* `$aulas`. Dessa forma, a aplicação consegue exibir na tela todos os cartões de aulas vinculados ao professor. Por fim, a conexão é encerrada com `$stmt->close()` e `$conn->close()`, assegurando o bom uso dos recursos do sistema.

3.5.3.3 - Tela aula selecionada do professor

Figura 81 - Tela aula selecionada do professor



FONTE: AUTORIA PRÓPRIA, 2025

Ao selecionar uma aula, o professor é levado para a tela de vídeo aula, semelhante à do aluno, porém com funcionalidades adicionais. Conforme mostrado na figura 81, um menu no canto superior direito permite o gerenciamento da aula. Abaixo do vídeo, são exibidos os alunos que já assistiram à aula, facilitando o

acompanhamento do progresso dos estudantes. Essa interface oferece ao professor tanto o conteúdo didático quanto ferramentas para monitorar o engajamento dos alunos e organizar o ensino de forma mais eficiente.

Figura 82 – Tela de Código aula selecionada do professor

```

</div>
<a class="logout" href="../../index.php">

<span> Sair</span>
</a>
</div>
<form class="main" method="POST">
<div class="header-aula">
<?php echo htmlspecialchars($tituloAula); ?>
<p style="font-size: 16px; font-weight: 400; margin-top: 10px;">
<?php echo nl2br(htmlspecialchars($descricaoAula)); ?>
</p>
</div>
<div class="video-container">
<!-- Bordo de 3 pontinhos -->
<div class="menu-opcoes">
<button type="button" class="menu-btn" onclick="toggleMenu(event)"></button>
<div class="menu-card">
<a href="alterarAula.php?id=<?php echo $idAula; ?>">Alterar aula</a>
<a href="excluirAula.php?id=<?php echo $idAula; ?>" onclick="return confirm('Tem certeza que deseja excluir esta aula?')">Excluir aula</a>
</div>
</div>
<!-- Video -->
<video id="aulaVideo" controls poster="../../img/thumbVideo.png">
<source src="video.php?id=<?php echo $idAula; ?>" type="video/mp4">
Seu navegador não suporta vídeo.
</video>
<div class="progress-wrapper">
<div class="progress-bar"></div>
</div>
<!-- Lista de alunos -->
<div style="max-width:800px; margin:auto; background:white; padding:20px; border-radius:12px; box-shadow:0 4px 10px rgba(0,0,0,0.1);">
<div style="margin-bottom:15px; color:#28a745;">Alunos que visualizaram essa aula:</div>
<?php if ($resultAlunos->num_rows > 0):>
<ul style="list-style:none; padding:0; margin:0;">
<?php while($aluno = $resultAlunos->fetch_assoc()):>
<li style="display:flex; align-items:center; margin-bottom:12px; border-bottom:1px solid #eee; padding-bottom:8px;">
"
alt="Foto"
style="width:40px; height:40px; border-radius:50%; object-fit:cover; margin-right:12px;">
<div>
<strong><?php echo htmlspecialchars($aluno['nome']); ?></strong><br>
<small><?php echo htmlspecialchars($aluno['email']); ?></small><br>
</div>
</li>
</ul>
<?php endwhile; ?>
<?php else: ?>
<p style="color:gray;">Nenhum aluno desta turma marcou como assistido ainda.</p>
<?php endif; ?>
</div>

```

FONTE: AUTORIA PRÓPRIA, 2025

O código da figura 82 é construído em HTML e PHP. Primeiramente, a interface organiza os elementos em seções, exibindo o título da aula, a descrição e o vídeo dentro da *div* “*video-container*”. Abaixo do vídeo, são disponibilizados botões que permitem ao professor alterar ou excluir a aula, sendo o último protegido por um alerta de confirmação em JavaScript para evitar exclusões acidentais.

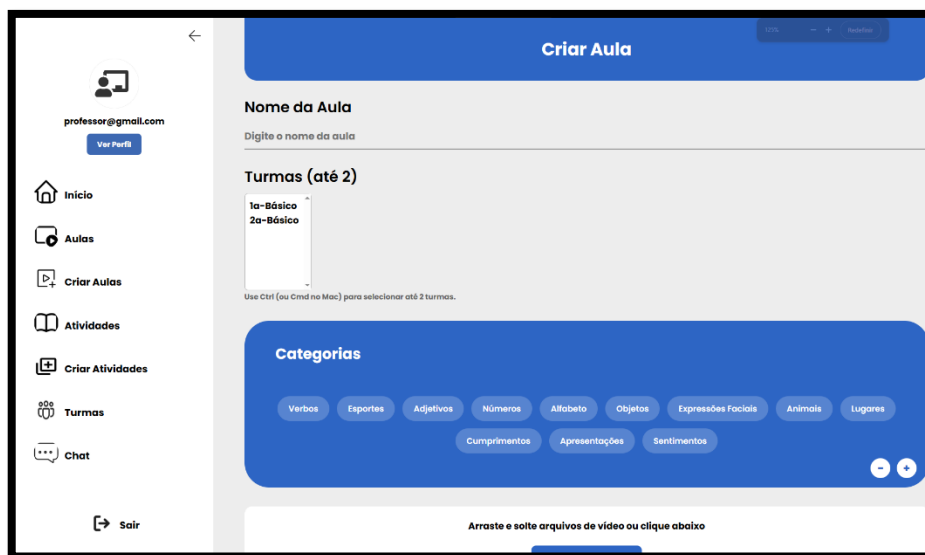
Outra parte importante é a seção “Alunos que visualizaram essa aula”, onde os dados são carregados dinamicamente a partir do banco. O código PHP percorre os registros retornados e exibe o nome e a foto de cada aluno que marcou a aula como assistida. Caso não haja registros, é exibida a mensagem “Nenhum aluno desta turma marcou como assistido ainda.”, garantindo clareza no *feedback* ao professor.

Por fim, o *design* mantém a consistência com as demais telas da plataforma, utilizando divisões estruturadas em HTML, estilos CSS para organização visual e PHP para integração com os dados armazenados no banco. Assim, essa tela oferece ao

professor tanto os recursos de visualização quanto de gestão das aulas, de forma prática e organizada.

3.5.3.4 – Tela criar aulas do Professor

Figura 83 - Tela criar aulas do Professor



FONTE: AUTORIA PRÓPRIA, 2025

A figura 83 representa a criação das aulas, onde o professor pode criar e personalizar suas aulas de forma simples e eficiente. No campo principal, ele pode definir o nome da aula, como "Esportes e Atividades Físicas", facilitando a identificação do conteúdo por alunos e professores. Logo abaixo, há a opção para selecionar até duas turmas que receberão a aula, permitindo o envio direcionado do material. Na seção de categorias, o professor pode escolher os temas que melhor descrevem o conteúdo da aula, como Esportes, Verbos, Números e Cumprimentos, o que ajuda na organização e busca por parte dos alunos.

A área destinada ao *upload* do vídeo oferece duas formas práticas de envio: arrastando o arquivo diretamente para o espaço indicado ou clicando para selecionar o vídeo a partir do computador. Isso torna o processo rápido e flexível, facilitando o trabalho do professor.

Além disso, há um campo para inserção da descrição da aula, onde o professor pode contextualizar o conteúdo, destacando os pontos principais ou objetivos, enriquecendo o entendimento dos alunos. No menu lateral, a navegação é simples e intuitiva, com acesso rápido às principais funcionalidades da plataforma,

como Início, Aulas, Criar Atividades, Turmas e *Chat*, proporcionando uma experiência fluida e organizada.

Essa interface foi projetada para tornar o processo de criação de aulas ágil e acessível, permitindo que o professor dedique mais tempo ao planejamento pedagógico e menos à parte técnica.

Figura 84 – Tela de código de criar aulas do Professor

```

1 // Criar aula
2
3 if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['criar_aula'])) {
4     $nomeAula = $_POST['nome_aula'];
5     $turmasSelecionadas = $_POST['turmas'];
6     $categoriaNome = $_POST['categoria'];
7     $descricao = $_POST['descricao'];
8
9     if (count($turmasSelecionadas) > 2) {
10        echo "<script>alert('Você só pode selecionar no máximo 2 turmas.');

```

FONTE: AUTORIA PRÓPRIA, 2025

A figura 84 apresenta o funcionamento da tela de criação de aula é sustentado pelo código PHP que processa o envio do formulário. Primeiramente, o *script* verifica se o método da requisição é *POST* e se o formulário de criação de aula foi submetido, usando a condição `if ($_SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['criar_aula']))`. Em seguida, ele recupera os dados enviados pelo professor, como o nome da aula, as turmas selecionadas, a categoria e a descrição, armazenando-os em variáveis.

Logo após o código verifica se o número de turmas selecionadas não ultrapassa o limite máximo de duas, exibindo uma mensagem de alerta e encerrando a execução caso esse limite seja excedido. A seguir, o *script* verifica se o arquivo de

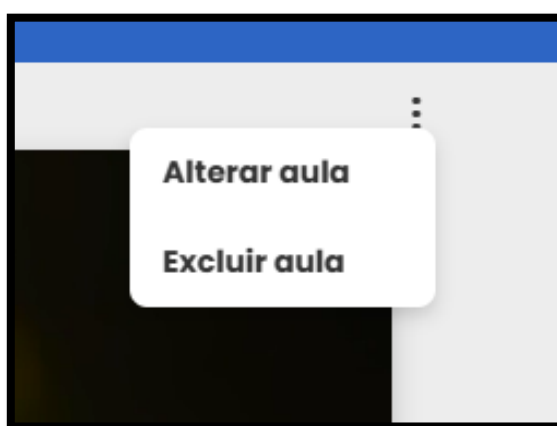
vídeo foi enviado corretamente, conferindo se não houve erro no *upload*. Caso positivo, ele lê o conteúdo do vídeo em formato binário para posterior armazenamento.

Para garantir a integridade dos dados, o código realiza uma consulta preparada para obter o ID da categoria correspondente ao nome informado, verificando se essa categoria existe no banco. Se a categoria não for encontrada, uma mensagem de erro é exibida e o processo é interrompido. Depois, as turmas selecionadas são convertidas para um *array* de inteiros, garantindo maior segurança e consistência.

O próximo passo é inserir os dados da aula na tabela do banco, incluindo nome, categoria, vídeo e descrição. O envio do vídeo é realizado por meio do método *send_long_data()*, que permite enviar arquivos binários de forma segura. Caso a inserção da aula seja bem-sucedida, o código recupera o ID gerado automaticamente para essa nova aula.

Com esse ID em mãos, o *script* faz a associação entre a aula e cada uma das turmas selecionadas, inserindo registros na tabela intermediária que relaciona aulas e turmas. Por fim, o sistema exibe uma mensagem de sucesso para o usuário. Se ocorrer algum erro durante a criação da aula ou no *upload* do vídeo, alertas específicos são apresentados para informar o problema.

Figura 85 – Tela alterar e excluir aulas do professor



FONTE: AUTORIA PRÓPRIA, 2025

A opção de menu apresenta as funcionalidades de alterar ou excluir a aula selecionada, como ilustrado na figura 85 acima. Ao clicar nos três pontos localizados no canto superior direito da tela da aula, o professor pode rapidamente editar o

conteúdo, corrigir informações ou realizar ajustes necessários. Além disso, caso a aula não seja mais relevante, há também a opção de excluí-la permanentemente da plataforma. Essas funcionalidades oferecem mais autonomia e controle sobre o material didático disponibilizado.

Figura 86 – Tela do Código alterar e excluir aulas do professor

```
<form class="main" method="POST">
  <div class="header-aula">
    <?php echo htmlspecialchars($tituloAula); ?>
    <p style="font-size: 16px; font-weight: 400; margin-top: 10px;">
    <?php echo nl2br(htmlspecialchars($descricaoAula)); ?>
  </p>
</div>

<div class="video-container">
  <!-- Botão de 3 pontinhos -->
  <div class="menu-opcoes">
    <button type="button" class="menu-btn" onclick="toggleMenu(event)"></button>
    <div class="menu-card" id="menuCard">
      <a href="alterarAula.php?id=<?php echo $idAula; ?>">Alterar aula</a>
      <a href="excluirAula.php?id=<?php echo $idAula; ?>" onclick="return confirm('Tem certeza que deseja excluir esta aula?')">Excluir aula</a>
    </div>
  </div>

  <!-- Vídeo -->
  <video id="aulaVideo" controls poster="..img/thumbVideo.png">
    <source src="video.php?id=<?php echo $idAula; ?>" type="video/mp4">
    Seu navegador não suporta vídeo.
  </video>

  <div class="progress-wrapper">
    <div class="progress-bar"></div>
  </div>

  <!-- Lista de alunos -->
  <div style="max-width:800px; margin:40px auto; background:white; padding:20px; border-radius:12px; box-shadow:0 4px 10px rgba(0,0,0,0.1);">
    <h3 style="margin-bottom:15px; color:#2d65c4;">Alunos que visualizaram essa aula:</h3>
  </div>
</form>
```

FONTE: AUTORIA PRÓPRIA, 2025

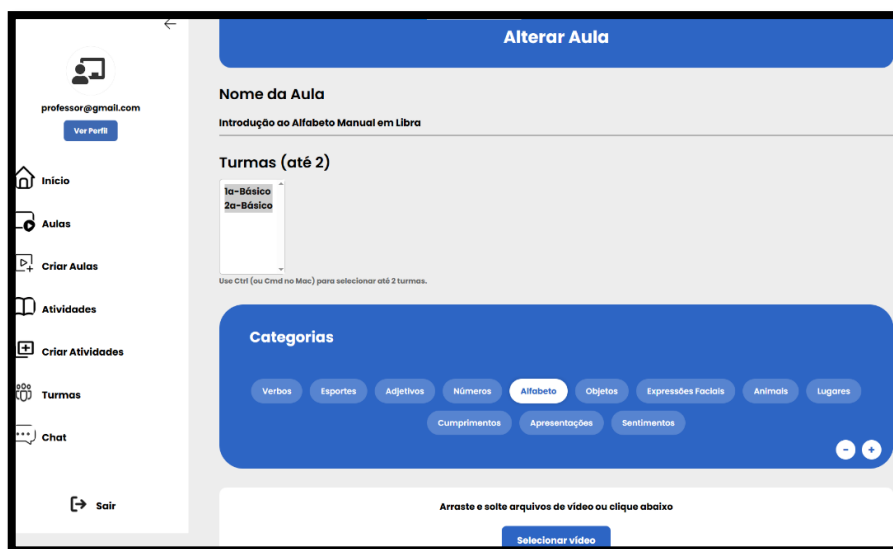
O código da figura 86 é construído em HTML e PHP, utilizando um menu em forma de lista de opções. O botão com o ícone de três pontos abre o menu (classe “*menu-props*”), no qual são exibidos os *links*. O *link* “Alterar aula” direciona o professor para a página de edição correspondente, levando junto o identificador da aula selecionada. Já a opção “Excluir aula” também utiliza o ID da aula, mas adiciona uma proteção com JavaScript por meio da função *confirm()*, que exibe uma mensagem de confirmação antes de concluir a ação de exclusão, evitando que a operação ocorra de forma acidental.

Além disso, a tela mantém o contexto da aula em execução, incluindo o vídeo e demais elementos já estruturados anteriormente. Isso garante que as opções de gerenciamento estejam integradas diretamente na mesma tela de exibição do conteúdo, otimizando a experiência de navegação do professor.

Assim, esse recurso funciona como um menu de ações rápidas, trazendo praticidade para a gestão das aulas, sem a necessidade de acessar outras páginas separadas para alterar ou excluir conteúdos.

3.5.3.5 - Tela de alterar aulas do Professor

Figura 87 - Tela do código de alterar aulas do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Na tela de alteração de aula, representada na figura 87, o professor tem à sua disposição diversas funcionalidades para modificar os detalhes de uma aula já cadastrada de forma intuitiva e organizada. O campo para edição do nome da aula está localizado na parte superior, permitindo a rápida atualização do título conforme a necessidade pedagógica.

Logo abaixo, o professor pode associar a aula a uma ou mais turmas específicas, utilizando uma lista interativa que apresenta as turmas disponíveis, facilitando o controle da distribuição do conteúdo. Além disso, há a opção de selecionar o nível de dificuldade da aula, o que ajuda a organizar o material didático de acordo com o perfil dos alunos.

A tela também oferece uma seção dedicada à categorização da aula. O professor pode escolher entre diversas categorias representadas por botões, como "Verbos", "Esportes", "Alfabeto", entre outras, para organizar o conteúdo de forma temática e facilitar a navegação e a busca futura. Outra funcionalidade importante é o campo para edição do material de estudo, onde o professor pode anexar vídeos ou outros recursos, tornando a aula mais rica e dinâmica. A área para descrição da aula permite que sejam adicionadas informações detalhadas sobre os objetivos, conteúdo e metodologia da aula, garantindo clareza para quem for acessar posteriormente.

Por fim, após realizar todas as modificações necessárias, o usuário confirma as alterações clicando no botão azul destacado “Alterar Aula”, que valida e salva as atualizações no sistema. Essa interface promove uma experiência de uso eficiente, intuitiva e alinhada às necessidades pedagógicas do docente, facilitando a gestão das aulas dentro da plataforma.

Figura 88 – Tela de código aula do professor

```

if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_POST['alterar_aula'])) {
    $nomeAula = $_POST['nome_aula'];
    $turmasPost = $_POST['turmas'] ?? [];
    $categoriaNome = $_POST['categoria'];
    $descricao = $_POST['descricao'];

    if (count($turmasPost) > 2) {
        echo "<script>alert('Você só pode selecionar no máximo 2 turmas.');

```

FONTE: AUTORIA PRÓPRIA, 2025

A figura 88 apresenta o código responsável por realizar a alteração de uma aula previamente cadastrada na plataforma. O processo se inicia com a verificação do método de requisição, assegurando que se trata de um envio via *POST* e que o botão de alteração (*alterar_aula*) foi acionado. Em seguida, os dados do formulário são capturados: *nomeAula* (nome da aula), *turmasPost* (lista de turmas associadas), *categoriaNome* (nome da categoria escolhida) e *descricao* (descrição textual da aula). O sistema impõe uma regra que permite associar a aula a no máximo duas turmas. Caso o número de turmas ultrapasse esse limite, uma mensagem de alerta é exibida ao usuário e o processo de alteração é interrompido.

Dando continuidade, o sistema realiza uma consulta SQL utilizando *SELECT id_categoria FROM categoria WHERE nomeCategoria =?* com o objetivo de recuperar o *id_categoria* correspondente ao nome da categoria fornecido. Essa etapa é essencial, pois o banco de dados utiliza o ID da categoria e não o nome textual. Caso a categoria informada não exista, é exibido um alerta indicando que a categoria é inválida. Se o ID for encontrado corretamente, o sistema passa à etapa de atualização dos dados principais da aula.

Nesse ponto, o código trata duas possibilidades: a aula pode ou não ter um novo vídeo associado a ela. Se um novo vídeo foi enviado (verificado com `$_FILES['video']['error'] === UPLOAD_ERR_OK`), o arquivo temporário é armazenado e seu conteúdo é lido como *long_data*. Em seguida, é preparada uma consulta SQL *'UPDATE aula SET nomeAula=?, id_categoria=?, video=?, descricaoAula=? WHERE id_aula=?'*, com parâmetros correspondentes ao novo nome, ID da categoria, conteúdo do vídeo, descrição e o ID da aula a ser atualizada. Caso nenhum vídeo tenha sido enviado, a instrução SQL exclui o campo de vídeo, mantendo os demais dados para alteração.

Independentemente da inclusão de vídeo, após a atualização da aula, o código realiza a limpeza das relações anteriores da aula com as turmas através de um *DELETE FROM aula_turma WHERE id_aula =?*. Após isso, as novas turmas informadas no formulário são associadas à aula utilizando a instrução *INSERT INTO aula_turma (id_aula, id_turma) VALUES (?,?)*, repetida para cada turma selecionada.

Caso todas as operações de atualização e reassociação sejam bem-sucedidas, o sistema exibe uma mensagem confirmando o sucesso da alteração, e redireciona o professor para a tela de gerenciamento de aulas. Em caso de falha, a mensagem de erro é exibida com o conteúdo do erro retornado pelo banco de dados.

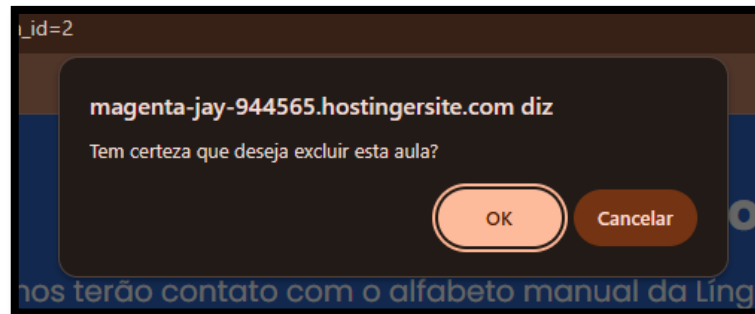
Dessa forma, o código garante a integridade e consistência das informações da aula, tratando adequadamente cada parte do processo de forma segura e estruturada. A implementação também demonstra preocupação com a usabilidade, proporcionando *feedback* imediato ao usuário por meio de alertas.

Além disso, o uso de *prepared statements* contribui para a proteção contra ataques de SQL Injection, reforçando a segurança da aplicação. Todo esse conjunto

de práticas resulta em um sistema mais confiável e eficiente para a gestão do conteúdo educacional.

3.5.3.6 - Tela de excluir aulas do Professor

Figura 89 - Tela de excluir aulas do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Nesta interface representada na figura 89, o professor pode revisar os principais dados da aula antes de confirmar sua exclusão. São exibidas informações como título, categorias vinculadas, vídeo associado e descrição da aula. Após essa verificação, o docente pode optar por excluir a aula clicando no botão "Excluir Aula", sendo solicitado a confirmar a ação por meio de uma caixa de diálogo de segurança.

Figura 90 - Tela do Código excluir aulas do Professor

```

<?php
session_start();
if (!isset($_SESSION['email'])) {
    header("Location: ../../index.php");
    exit();
}

$host = "localhost";
$usuario = "u357936358_librinha";
$senha = "Librinh4tcc#";
$banco = "u357936358_scanhands";

$conn = new mysqli($host, $usuario, $senha, $banco);
if ($conn->connect_error) {
    die("Erro de conexão com o banco: " . $conn->connect_error);
}

$idAula = isset($_GET['id']) ? intval($_GET['id']) : 0;
if ($idAula <= 0) {
    die("ID da aula inválido.");
}

// Apagar vínculos com turmas
$stmtDelTurmas = $conn->prepare("DELETE FROM aula_turma WHERE id_aula = ?");
$stmtDelTurmas->bind_param("i", $idAula);
$stmtDelTurmas->execute();
$stmtDelTurmas->close();

// Apagar aula
$stmtDelAula = $conn->prepare("DELETE FROM aula WHERE id_aula = ?");
$stmtDelAula->bind_param("i", $idAula);
if ($stmtDelAula->execute()) {
    $stmtDelAula->close();
    $conn->close();
    echo "<script>alert('Aula excluída com sucesso!'); window.location.href='aulasProfessor.php';</script>";
} else {
    $stmtDelAula->close();
    $conn->close();
    echo "<script>alert('Erro ao excluir aula.');

```

FONTE: AUTORIA PRÓPRIA, 2025

O código representado na figura 90 possui linguagem PHP e SQL que implementa essa tela inicia com a verificação da sessão do professor, garantindo que apenas usuários autenticados possam realizar a exclusão. Em seguida, recupera o ID da aula a partir da URL (`GET['id']`) e valida se o valor é realmente um número inteiro. Caso o ID seja inválido, o sistema interrompe a execução e exibe a mensagem de erro. A exclusão ocorre em duas etapas:

Remover vínculos da aula com turmas: o código executa um comando *DELETE* na tabela intermediária *aula_turma*, eliminando todas as associações daquela aula com as turmas cadastradas. Remover a aula em si: em seguida, um novo comando *DELETE* é executado diretamente na tabela *aula*, apagando definitivamente o registro correspondente. O código utiliza *statements* preparados (*prepare* e *bind_param*) para evitar falhas de segurança, como SQL Injection, e garante maior confiabilidade na execução.

Ao final, caso a exclusão seja bem-sucedida, o sistema retorna uma mensagem em JavaScript informando que a aula foi excluída com sucesso e redireciona o professor para a página de gerenciamento de aulas. Se ocorrer algum erro no processo, uma mensagem de falha é exibida, orientando o professor de que a exclusão não foi concluída. Assim, essa tela garante que o professor tenha controle total sobre o gerenciamento das aulas, oferecendo segurança e clareza no processo de exclusão, além de evitar exclusões acidentais por meio da confirmação obrigatória.

3.5.3.7 – Tela de atividades do Professor

Figura 91 - Tela de atividades do Professor



FONTE: AUTORIA PRÓPRIA, 2025

A figura 91 representa a tela de atividades, onde o professor pode visualizar todas as atividades criadas para suas turmas de forma clara e organizada. Cada atividade aparece em um cartão destacando o título e a turma a que está vinculada, facilitando a identificação rápida do conteúdo. Por exemplo, atividades como "De Letra em Letra – Explorando o Alfabeto em Libras" estão associadas às turmas 1º- Básico e 2º-Básico, enquanto a atividade "Expressões Faciais e Emoções" está vinculada à turma 2º- Básico.

Além da visualização, a tela conta com a opção de filtrar as atividades, permitindo que o professor encontre rapidamente conteúdos específicos entre todas as atividades cadastradas, otimizando a gestão do material didático. Essa funcionalidade torna o processo de acompanhamento e organização mais eficiente, especialmente para professores que trabalham com múltiplas turmas e diversos temas.

O menu lateral mantém a navegação simples e direta, oferecendo acesso rápido a outras seções importantes da plataforma, como Início, Aulas, Criar Aulas, Criar Atividades e Turmas. A interface foi projetada para proporcionar uma experiência intuitiva, auxiliando o professor na administração das atividades educacionais.

Figura 92 – Tela de código da Atividades do Professor

```

}
$whereClause = implode(" AND ", $conditions);

/* =====
   BUSCAR ATIVIDADES DAS TURMAS DO PROFESSOR
   (somente turmas relacionadas ao professor logado)
   =====*/
$sql = "SELECT
        atv.id_atv,
        atv.nomeAtv AS titulo,
        t.nome      AS turma,
        t.id_turma AS turma_id
    FROM atividade atv
    INNER JOIN atividade_turma att ON att.id_atv = atv.id_atv
    INNER JOIN turma      t      ON t.id_turma = att.id_turma
    INNER JOIN professor_turma pt ON pt.id_turma = t.id_turma
    WHERE $whereClause
    ORDER BY t.nome, atv.nomeAtv";

$stmt = $conn->prepare($sql);
$stmt->bind_param($types, ...$params);
$stmt->execute();
$result = $stmt->get_result();

$atividades = [];
while ($row = $result->fetch_assoc()) {
    $atividades[] = $row; // cada card representa (atividade + turma)
}
$stmt->close();
$conn->close();
?>

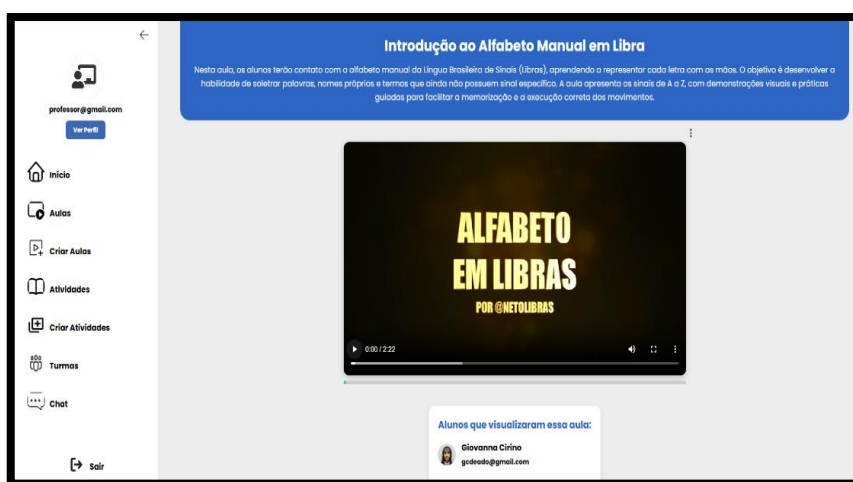
```

O código da figura 92 realiza a busca utilizando uma consulta SQL. Primeiro, são criadas as condições de filtro que ficam armazenadas na variável *\$whereClause*, construída a partir do *array \$conditions* com *implode(" AND ", \$conditions)*. Em seguida, a *query* faz a seleção dos dados principais: o identificador da atividade (*id_atv*), o nome da atividade (renomeado como título), o nome da turma e o identificador da turma. Para isso, utiliza *INNER JOIN* em três tabelas: *atividade_turma* (que relaciona atividades e turmas), *turma* (dados da turma) e *professor_turma* (que garante que apenas as turmas daquele professor sejam retornadas).

Com a *query* montada, o comando *\$stmt = \$conn->prepare(\$sql)* prepara a execução no banco. Os parâmetros necessários são vinculados com *\$stmt->bind_param(\$types, ...\$params)* e, depois, a consulta é executada. O resultado é capturado em *\$result = \$stmt->get_result()*, e o *loop while* percorre cada linha retornada, armazenando as informações em *\$atividades[]*. Assim, cada posição do *array* representa um "card" de atividade vinculado a uma turma. Por fim, a conexão e o *statement* são fechados com *\$stmt->close()* e *\$conn->close()*.

3.5.3.8 - Tela de atividade selecionada do professor

Figura 93 - Tela atividade selecionada do professor



FONTE: AUTORIA PRÓPRIA, 2025

Ao selecionar uma atividade, o professor é direcionado para uma tela que exibe um vídeo explicativo relacionado à atividade escolhida, conforme ilustrado na figura 93. Essa interface foi projetada para proporcionar um melhor entendimento do conteúdo por meio de uma demonstração visual clara e objetiva. Além do vídeo, a tela apresenta uma opção para ativar a câmera, permitindo que o professor realize a parte

prática da atividade em tempo real, favorecendo uma experiência interativa e mais próxima da realidade da aula presencial.

Na lateral esquerda, encontra-se um menu de navegação que facilita o acesso às principais funcionalidades da plataforma, como Início, Aulas, Criar Aulas, Atividades, Criar Atividades, Turmas e *Chat*, garantindo uma navegação intuitiva e eficiente. Além disso, há opções específicas para que o professor possa alterar ou excluir a atividade selecionada, conferindo maior controle e flexibilidade na gestão do conteúdo ministrado. Essa combinação de recursos assegura que o professor tenha todas as ferramentas necessárias para adaptar, gerenciar e executar suas atividades de forma dinâmica e personalizada, aprimorando a interação com os alunos e a qualidade do ensino oferecido.

Figura 94 –Tela de código atividade selecionada do professor

```
// Dados da atividade
$stmtAtv = $conn->prepare("SELECT nomeAtv, descricaoAtv FROM atividade WHERE id_atv = ?");
$stmtAtv->bind_param("i", $idAtividade);
$stmtAtv->execute();
$resultAtv = $stmtAtv->get_result();

$tituloAtividade = "Atividade";
$descricaoAtividade = "";
if ($resultAtv && $resultAtv->num_rows > 0) {
    $atividade = $resultAtv->fetch_assoc();
    $tituloAtividade = $atividade['nomeAtv'];
    $descricaoAtividade = $atividade['descricaoAtv'];
}
$stmtAtv->close();

// Buscar alunos da turma que assistiram a atividade
$sqlAlunos = "
    SELECT u.id_usuario, u.nome, u.email, u.foto_perfil, uf.data_feita
    FROM usuario_atividade_feita uf
    INNER JOIN usuario u ON u.id_usuario = uf.id_usuario
    INNER JOIN aluno a ON a.id_usuario = u.id_usuario
    INNER JOIN aluno_turma atur ON atur.id_aluno = a.id_aluno
    WHERE uf.id_atividade = ? AND atur.id_turma = ?
    ORDER BY uf.data_feita DESC
";
$stmtAlunos = $conn->prepare($sqlAlunos);
$stmtAlunos->bind_param("ii", $idAtividade, $idTurma);
$stmtAlunos->execute();
$resultAlunos = $stmtAlunos->get_result();

$conn->close();
?>
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title><?php echo htmlspecialchars($tituloAtividade); ?></title>
<link rel="icon" href="..img/librinhaSemFundo.png" type="image/png">
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@400;600;700&display=swap" rel="stylesheet">
<link rel="stylesheet" href="..css/inicioAluno.css">
```

FONTE: AUTORIA PRÓPRIA, 2025

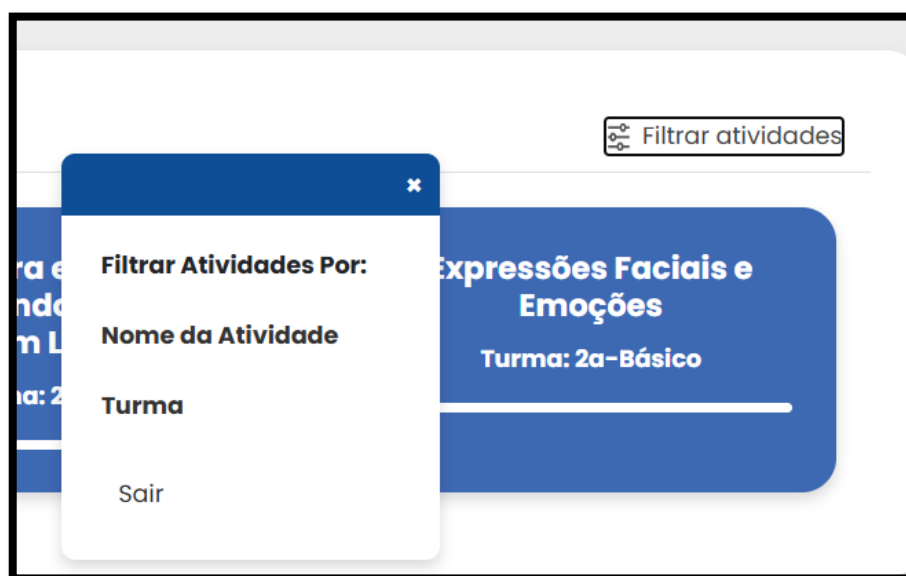
O código da figura 94 inicia com uma consulta SQL preparada para buscar no banco de dados as informações da atividade selecionada. A *query* seleciona o nome (nomeAtv) e a descrição (descricaoAtv) a partir do identificador único da atividade (id_atv). Após a execução, os resultados são verificados: caso exista uma correspondência, os valores são armazenados nas variáveis \$tituloAtividade e \$descricaoAtividade, que servirão para compor o cabeçalho da página.

Em seguida, é realizada uma segunda consulta, destinada a identificar os alunos da turma vinculados à atividade. Essa *query* utiliza a tabela *usuario_atividade_feita* para verificar quais estudantes concluíram a atividade, relacionando os registros às tabelas *usuario*, *aluno* e *aluno_turma*. São coletados dados como identificador do usuário, nome, *e-mail*, foto de perfil e a data em que a atividade foi concluída (*data_feita*). Os resultados são organizados em ordem decrescente de data, possibilitando ao professor visualizar os alunos mais recentes no topo da lista.

Por fim, os dados coletados são preparados para exibição em uma página HTML, cujo título é definido dinamicamente com base no nome da atividade. O código também integra folhas de estilo e fontes externas para compor a interface visual. Assim, o professor consegue consultar de forma prática tanto as informações da atividade quanto o desempenho dos alunos da turma associada.

3.5.3.9 – Tela de filtrar atividades

Figura 95 - Tela de filtrar atividades



FONTE: AUTORIA PRÓPRIA, 2025

A figura 95 apresenta a opção de filtro na tela de atividades permite que o professor localize facilmente uma atividade específica, tornando a busca mais rápida e eficiente. Conforme ilustrado na figura acima, ao clicar em "Filtrar atividades", é exibido um menu com as opções para filtrar por "Nome da Atividade" ou por "Turma". Essa funcionalidade possibilita que o professor refine sua pesquisa, facilitando o

acesso a conteúdos já criados, principalmente quando há muitas atividades cadastradas.

Além disso, o filtro contribui para a organização do material didático, ajudando o professor a gerenciar suas atividades de maneira mais prática e personalizada conforme suas necessidades pedagógicas. A interface do filtro é simples e intuitiva, garantindo que mesmo usuários com pouca experiência tecnológica possam utilizá-la sem dificuldades.

Figura 96 – Tela de código de filtrar atividades

```
// Limpar filtros
if (!empty($_GET['limpar'])) {
    header("Location: atividadesAluno.php");
    exit();
}

// Montar condições dinamicamente
$conditions = ["u.email = ?"];
$params = [$emailLogado];

if (!empty($_GET['nome'])) {
    $conditions[] = "atv.nomeAtv LIKE ?";
    $params[] = "%" . $_GET['nome'] . "%";
}

if (!empty($_GET['turma'])) {
    $conditions[] = "t.nome LIKE ?";
    $params[] = "%" . $_GET['turma'] . "%";
}

$whereClause = implode(" AND ", $conditions);
```

FONTE: AUTORIA PRÓPRIA, 2025

O código da figura 96 inicia com a verificação do botão de limpar filtros. Caso o parâmetro `$_GET['limpar']` esteja preenchido, o sistema redireciona o usuário para a página principal de atividades (*atividadesAluno.php*), removendo todos os filtros aplicados. Em seguida, são construídas dinamicamente as condições da consulta. Inicialmente, a condição básica garante que os registros exibidos estejam vinculados ao usuário logado, armazenando o *e-mail* desse usuário na variável `$params`.

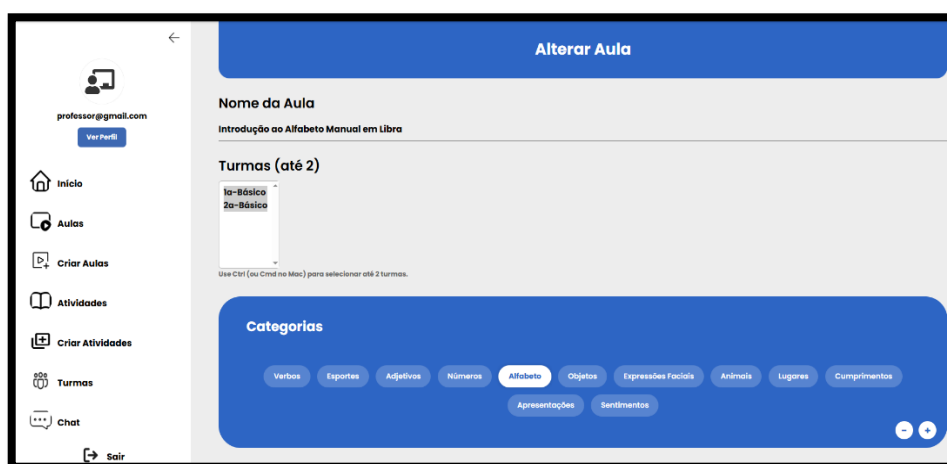
Se o campo de nome for informado (`$_GET['nome']`), o sistema adiciona à lista de condições a busca por atividades cujo título contenha o valor digitado, utilizando a cláusula *LIKE*. De maneira semelhante, caso o campo de turma seja

preenchido (`$_GET['turma']`), a condição adicionada verifica as turmas que contenham o nome correspondente.

Por fim, todas as condições são unidas em uma única *string* através de *implode(" AND ", \$conditions)*, formando o trecho da cláusula *WHERE* que será integrado à consulta SQL. Dessa forma, o professor pode visualizar apenas as atividades que atendam aos filtros definidos, otimizando a navegação e gestão de conteúdos.

3.5.3.10 - Tela de Alterar atividades do Professor

Figura 97 - Tela de Alterar atividades do Professor



FONTE: AUTORIA PRÓPRIA, 2025

A tela de alteração de atividade, representada na figura 97, oferece ao professor uma interface funcional e intuitiva para editar atividades previamente cadastradas na plataforma. Nela, é possível alterar o nome da atividade, garantindo que o título reflita com clareza o conteúdo abordado. O sistema também permite ao professor vincular a atividade a até duas turmas específicas, por meio de uma caixa de seleção que limita a escolha a no máximo duas opções.

Outro recurso importante apresentado na tela é a seção de categorias temáticas, que organiza os conteúdos de forma pedagógica. As opções visíveis incluem temas como Verbos, Esportes, Adjetivos, Números, Alfabeto, Objetos, Expressões Faciais, Animais, Lugares, Cumprimentos, Apresentações e Sentimentos, facilitando a classificação da atividade e seu alinhamento com o planejamento pedagógico.

Além disso, há um campo dedicado à descrição da atividade, onde o professor pode detalhar os objetivos, instruções, materiais necessários e outras informações relevantes para a execução da atividade por parte dos alunos.

Para concluir o processo de edição, o professor deve clicar no botão “Alterar Atividade”, localizado ao final da tela. Esse comando confirma e aplica todas as modificações feitas, garantindo que as informações estejam atualizadas e coerentes com os objetivos da aula. A disposição clara dos elementos e o uso de cores contrastantes contribuem para uma experiência de usuário eficiente, tornando a gestão de atividades simples e acessível.

Figura 98 – Tela de código de alterar atividades do Professor

```

/* =====
SALVAR ALTERAÇÕES
===== */
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $nome = $_POST['nomeAtv'] ?? '';
    $descricao = $_POST['descricaoAtv'] ?? '';
    $categoriaNome = $_POST['categoria'] ?? '';
    $novasTurmas = $_POST['turmas'] ?? [];

    // Buscar ID da categoria
    $stmtCat = $conn->prepare("SELECT id_categoria FROM categoria WHERE nomeCategoria = ?");
    $stmtCat->bind_param("s", $categoriaNome);
    $stmtCat->execute();
    $resCat = $stmtCat->get_result();
    $idCategoria = ($resCat->num_rows > 0) ? $resCat->fetch_assoc()['id_categoria'] : null;
    $stmtCat->close();

    // Atualiza dados da atividade
    $stmtUp = $conn->prepare("UPDATE atividade SET nomeAtv = ?, descricaoAtv = ?, id_categoria = ? WHERE id_atv = ?");
    $stmtUp->bind_param("ssii", $nome, $descricao, $idCategoria, $idAtividade);
    $stmtUp->execute();
    $stmtUp->close();

    // Se foi enviado vídeo novo
    if (!empty($_FILES['videoAtv']['tmp_name'])) {
        $videoData = file_get_contents($_FILES['videoAtv']['tmp_name']);
        $stmtVid = $conn->prepare("UPDATE atividade SET video = ? WHERE id_atv = ?");
        $null = NULL;
        $stmtVid->bind_param("bi", $null, $idAtividade);
        $stmtVid->send_long_data(0, $videoData);
        $stmtVid->execute();
        $stmtVid->close();
    }

    // Atualizar turmas vinculadas
    $conn->query("DELETE FROM atividade_turma WHERE id_atv = $idAtividade");
    $stmtRel = $conn->prepare("INSERT INTO atividade_turma (id_atv, id_turma) VALUES (?, ?)");
    foreach ($novasTurmas as $idTurma) {
        $idTurma = intval($idTurma);
        $stmtRel->bind_param("ii", $idAtividade, $idTurma);
        $stmtRel->execute();
    }
    $stmtRel->close();

    echo "<script>alert('Atividade atualizada com sucesso!'); window.location.href='verAtividadeProf.php?id={$idAtividade}';</script>";
}

```

FONTE: AUTORIA PRÓPRIA, 2025

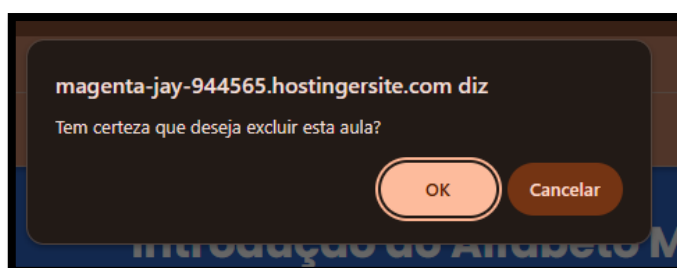
Este de código exibido na figura 98 é escrito em PHP atualiza os dados de uma atividade em um sistema, com base nos dados enviados por um formulário via método *POST*. O código realiza diferentes operações no banco de dados, utilizando MySQL com *prepared statements* para evitar SQL Injection. Primeiramente, ele verifica se o método da requisição é *POST*, o que garante que os dados foram enviados corretamente através do formulário.

Em seguida, ele coleta as informações enviadas, como o nome da atividade (\$nome), descrição (\$descricao), nome da categoria (\$nomeCategoria), ID da turma (\$idTurma), ID da atividade (\$idAtividade) e as novas turmas que devem ser associadas à atividade (\$novasTurmas). Depois disso, o código faz uma busca no banco de dados para obter o ID da categoria com base no nome fornecido. Se a categoria for encontrada, ela é usada para atualizar os dados da atividade na tabela atividade. Em seguida, o código remove todas as associações anteriores da atividade com as turmas na tabela atividade_turma para evitar duplicidade ou inconsistência.

Ele então reatribui a atividade às novas turmas informadas, inserindo um novo registro para cada turma associada. Ao final, uma mensagem de sucesso é exibida com um *alert* em JavaScript, redirecionando o usuário para a página de verificação do professor, utilizando o ID da atividade como parâmetro. Em resumo, este código permite editar uma atividade existente, alterando seu nome, descrição, categoria e as turmas associadas, garantindo que os dados estejam atualizados corretamente no banco de dados.

3.5.3.11 - Tela de Excluir atividades do Professor

Figura 99 - Tela de Excluir atividades do Professor



FONTE: AUTORIA PRÓPRIA, 2025

A figura 99 representa a tela de confirmação para exclusão de uma aula. Quando o professor opta por remover uma aula previamente cadastrada, o sistema exibe uma janela de alerta com a mensagem: "Tem certeza que deseja excluir esta aula?", solicitando que o usuário confirme ou cancele a ação. Essa interface tem como objetivo evitar exclusões acidentais, reforçando a importância da decisão.

Antes de chegar a esse ponto, o sistema exibe informações relevantes sobre a aula, como o nome ("Números"), as turmas associadas (por exemplo, "Básico") e as categorias vinculadas, como Números, Cumprimentos e Expressões Faciais. A exclusão só é realizada de forma definitiva quando o professor pressiona o botão

"OK", enquanto a opção "Cancelar" permite interromper a ação e retornar à tela anterior. Essa abordagem garante maior controle ao usuário, além de seguir boas práticas de usabilidade e segurança da informação, oferecendo uma experiência mais confiável e clara durante o processo de gerenciamento de conteúdos.

Figura 100 – Tela de código de excluir atividades do Professor

```

$host = "localhost";
$usuario = "u357936358_librinha";
$senha = "Librinh4tccc";
$banco = "u357936358_scanhands";

$conn = new mysqli($host, $usuario, $senha, $banco);
if ($conn->connect_error) {
    die("Erro de conexão com o banco: " . $conn->connect_error);
}

$idAtividade = isset($_GET['id']) ? intval($_GET['id']) : 0;
if ($idAtividade <= 0) {
    die("ID da atividade inválido.");
}

// Verificar se a atividade pertence ao professor logado
$emailLogado = $_SESSION['email'];
$stmt = $conn->prepare("
    SELECT atv.id_atv
    FROM atividade atv
    INNER JOIN atividade_turma att ON att.id_atv = atv.id_atv
    INNER JOIN turma t ON t.id_turma = att.id_turma
    INNER JOIN professor_turma pt ON pt.id_turma = t.id_turma
    INNER JOIN professor p ON p.id_professor = pt.id_professor
    INNER JOIN usuario u ON u.id_usuario = p.id_usuario
    WHERE atv.id_atv = ? AND u.email = ?
");
$stmt->bind_param("is", $idAtividade, $emailLogado);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows === 0) {
    die("Você não tem permissão para excluir esta atividade.");
}
$stmt->close();

// Excluir vínculos com turmas
$stmtDelTurmas = $conn->prepare("DELETE FROM atividade_turma WHERE id_atv = ?");
$stmtDelTurmas->bind_param("i", $idAtividade);
$stmtDelTurmas->execute();
$stmtDelTurmas->close();

// Excluir atividade
$stmtDelAtv = $conn->prepare("DELETE FROM atividade WHERE id_atv = ?");
$stmtDelAtv->bind_param("i", $idAtividade);
if ($stmtDelAtv->execute()) {
    $stmtDelAtv->close();
    $conn->close();
    echo "<script>alert('Atividade excluída com sucesso!'); window.location.href='atividadesProfessor.php';</script>";
} else {
    $stmtDelAtv->close();
    $conn->close();
    echo "<script>alert('Erro ao excluir atividade.');

```

FONTE: AUTORIA PRÓPRIA, 2025

Este código exibido na figura 100 em PHP é responsável por excluir uma atividade do sistema, garantindo que somente o professor dono da atividade possa realizar essa exclusão. O *script* realiza várias etapas importantes, como conexão ao banco de dados, verificação de permissões e remoção segura dos dados relacionados. Inicialmente, o código define os parâmetros de conexão ao banco de dados, como *host*, usuário, senha e nome do banco.

Em seguida, tenta se conectar ao MySQL. Caso a conexão falhe, uma mensagem de erro é exibida e o *script* é encerrado. Depois disso, o *script* obtém o ID da atividade que se deseja excluir por meio do parâmetro *id* enviado via *GET*, garantindo que o valor seja um número inteiro com a função *intval*. A etapa seguinte

é crucial para a segurança do sistema: o código verifica se a atividade realmente pertence ao professor atualmente logado.

Para isso, é feita uma consulta SQL que une várias tabelas (atividade, atividade_turma, turma, professor) e compara o ID do professor com o ID da sessão atual. Se essa verificação não retornar resultados, significa que o professor não tem permissão para excluir a atividade, e uma mensagem de alerta é exibida ao usuário.

Se a verificação for bem-sucedida, o código prossegue para remover todas as associações da atividade com as turmas na tabela atividade_turma, garantindo integridade referencial. Em seguida, a atividade em si é excluída da tabela atividade. Ao final, uma mensagem de sucesso ou erro é exibida através de um *alert* em JavaScript, redirecionando o usuário para a página de atividades do professor.

3.5.3.12 – Tela criar Atividades do Professor

Figura 101 - Tela criar Atividades do Professor



FONTE: AUTORIA PRÓPRIA, 2025

A tela de criação de atividade, ilustrada na figura 101, oferece uma interface prática e intuitiva para que o professor possa facilmente inserir todos os dados necessários para o cadastro de uma nova atividade na plataforma. O campo para o nome da atividade é simples, permitindo que o professor defina um título claro e objetivo, que será facilmente identificado por ele e pelos alunos. A possibilidade de associar a atividade a uma ou até duas turmas torna o processo mais flexível, permitindo que o mesmo conteúdo seja compartilhado com diferentes grupos de alunos, otimizando o tempo e o esforço do professor.

Além disso, o espaço destinado à descrição permite detalhar o propósito da atividade, as instruções para execução e quaisquer informações adicionais relevantes, garantindo que os alunos compreendam claramente o que se espera deles. O seletor de categorias, que apresenta opções como Verbos, Esportes, Adjetivos, Números, Alfabeto, entre outras, facilita a organização e a classificação do conteúdo, o que pode ser útil para buscas futuras e para manter a plataforma bem estruturada.

Outro ponto importante é o *design* responsivo e visualmente agradável da interface, que utiliza cores e espaçamentos que tornam a navegação fluida e agradável. O botão “Criar Atividade” é destacado para que o professor conclua o processo de forma clara, evitando confusões. Essa funcionalidade completa assegura que a criação de atividades seja feita de forma rápida, eficiente e organizada, promovendo um ambiente de ensino mais dinâmico e acessível.

Figura 102 – Tela de código de criar Atividades do Professor

```

=====
CADASTRAR NOVA ATIVIDADE
=====
if ($SERVER['REQUEST_METHOD'] === 'POST' && isset($_POST['criar_atv'])) {
    $nomeAula = $_POST['nome_atv'];
    $turmasSelecionadas = $_POST['turmas']; // array
    $categoriaNome = $_POST['categoria'];
    $descricao = $_POST['descricao'];

    if (count($turmasSelecionadas) > 2) {
        echo "<script>alert('Você só pode selecionar no máximo 2 turmas.');

```

FONTE: AUTORIA PRÓPRIA, 2025

Este código exibido na figura 102 em PHP é responsável por cadastrar uma nova atividade em um sistema educacional, incluindo o *upload* de vídeo, inserção das informações no banco de dados e associação da atividade com uma ou mais turmas.

Todo o processo ocorre quando o formulário é enviado via método *POST*. A execução começa verificando se o formulário foi submetido corretamente (*isset(\$_POST['criar_atv'])*) e se foram selecionadas ao menos uma turma para associar à nova atividade. Caso contrário, um alerta é exibido e a execução é encerrada. Em seguida, o código valida o *upload* do vídeo. Se um vídeo for enviado sem erros, ele é carregado e convertido para uma *string* binária usando *file_get_contents()*. Depois disso, o *script* realiza uma consulta ao banco de dados para obter o ID da categoria informada, com base no nome da categoria (*\$nomeCategoria*). Essa informação é usada como chave estrangeira ao cadastrar a atividade.

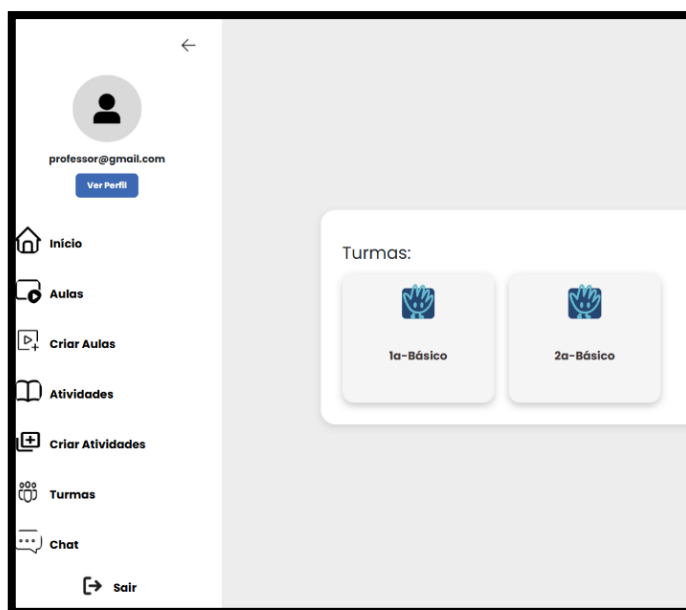
Depois de obter o ID da categoria, o *script* insere os dados da nova atividade na tabela atividade, incluindo nome, categoria, vídeo, descrição e o ID do professor responsável (provavelmente vindo da sessão). Após a inserção, o ID da atividade recém-criada é recuperado com *\$stmt->insert_id*. O passo seguinte é associar a nova atividade às turmas selecionadas. Para isso, é feito um *loop* que percorre todas as turmas e insere registros na tabela atividade_turma, criando o vínculo entre a atividade e cada turma individualmente.

Por fim, uma mensagem de sucesso ou erro é exibida por meio de um *alert* em JavaScript, informando o usuário sobre o resultado da operação. O *script* também garante o fechamento das conexões com o banco de dados e o uso de *prepared statements* para evitar vulnerabilidades como SQL Injection. Em resumo, este código realiza o cadastro completo de uma nova atividade, incluindo o *upload* de vídeo, categorização, descrição, vinculação com turmas e associação ao professor responsável, tudo de forma segura e validada.

Além disso, o controle de limite de turmas (máximo de duas) assegura que a estrutura da aplicação seja mantida conforme as regras de negócio definidas previamente. Essa verificação ajuda a evitar inconsistências nos dados e melhora a experiência do usuário ao alertar imediatamente quando um limite é ultrapassado. O uso de estruturas condicionais bem definidas, como *if* e *foreach*, contribui para a legibilidade e organização do código. Também é importante destacar que, ao isolar as operações de inserção e relacionamento em etapas distintas, o sistema facilita futuras manutenções e melhorias no processo. Isso demonstra uma boa prática de modularização dentro da lógica do *script*.

3.5.3.13 – Tela de turmas do Professor

Figura 103 - Tela de turmas do Professor



FONTE: AUTORIA PRÓPRIA, 2025

Esta é a tela de gerenciamento de turmas, representada na figura 103, onde o professor visualiza todas as turmas organizadas por níveis (Básico, Intermediário e Avançado) e por séries (1ª a 3ª). A interface foi desenvolvida para proporcionar uma experiência simples e intuitiva, permitindo ao usuário localizar rapidamente as turmas às quais está vinculado. Cada turma é exibida em um cartão individual, com identificação clara do nome e um ícone visual que facilita o reconhecimento imediato. Ao clicar sobre uma turma, o professor pode acessar funcionalidades específicas como visualização de alunos matriculados, conteúdos associados (como aulas e atividades), bem como realizar edições ou atribuir novos materiais.

Além da visualização, a tela conta com um sistema de filtragem que possibilita ao professor refinar os resultados exibidos conforme o nível ou série, tornando mais eficiente o gerenciamento em contextos com grande número de turmas. Essa função de filtro é particularmente útil em instituições maiores, onde o docente pode atuar em múltiplos segmentos. A proposta visual também mantém a consistência com o restante da plataforma, utilizando um menu lateral fixo que garante fácil navegação entre diferentes módulos, como Aulas, Atividades e *Chat*. Essa organização contribui para a produtividade do professor, centralizando as informações acadêmicas de forma acessível e funcional, promovendo um ambiente de trabalho mais ágil e controlado.

Figura 104 – Tela de código de turmas do Professor

```

<div class="sidebar" style="margin-right: 50px;">
  <div class="sidebar-top">
    <button id="toggleSidebar">
      
    </button>
    
    <p class="email"><?php echo htmlspecialchars($emailLogado); ?></p>
    <button style="font-size: 12px;" onclick="window.location.href='perfilSecretaria.php'">Ver Perfil</button>
    <nav>
      <a href="inicioProfessor.php"><span> Início</span></a>
      <a href="aulasProfessor.php"><span> Aulas</span></a>
      <a href="criarAula.php"><span> Criar Aulas </span></a>
      <a href="atividadesProfessor.php"><span> Atividades</span></a>
      <a href="criarAtividades.php"><span> Criar Atividades </span></a>
      <a href="turmasProfessor.php"><span> Turmas </span></a>
      <a href="#"><span> Chat</span></a>
    </nav>
  </div>
  <a class="logout" href="../..../index.php">
    
    <span> Sair</span>
  </a>
</div>

<div class="main">
  <div class="card-container">
    <div class="header-buttons">
    </div>
    <h4 style="margin-bottom: 15px;">Turmas:</h4>
    <div class="turma-grid">
      <?php if ($result && $result->num_rows > 0): ?>
      <?php while($row = $result->fetch_assoc()): ?>
        <div class="turma-card" onclick="window.location.href='publicacoes.php?id_turma=?= $row['id_turma'] ?>'>
          
          <div class="turma-nome"><?= htmlspecialchars($row["nome"]) ?></div>
          <div class="turma-actions">
          </div>
        </div>
      <?php endwhile; ?>
      <?php else: ?>
      <p style="text-align: center;">Nenhuma turma encontrada.</p>
      <?php endif; ?>
    </div>
  </div>
</div>

<script>
const toggleBtn = document.getElementById('toggleSidebar');
const sidebar = document.querySelector('.sidebar');
toggleBtn.addEventListener('click', () => {
  sidebar.classList.toggle('collapsed');
});
</script>
</body>
</html>
<?php $conn->close(); ?>

```

FONTE: AUTORIA PRÓPRIA, 2025

Este código ilustrado na figura 104 é uma combinação de HTML, PHP e JavaScript usada para estruturar e dinamizar uma interface *web* de um sistema educacional ou administrativo. Na parte superior do código, há um menu lateral (*sidebar*) identificado por "*toggleSidebar*", que apresenta informações do usuário logado, incluindo uma imagem (possivelmente o avatar do usuário) e seu nome, obtido dinamicamente com `{{ Auth::user()->name }}`.

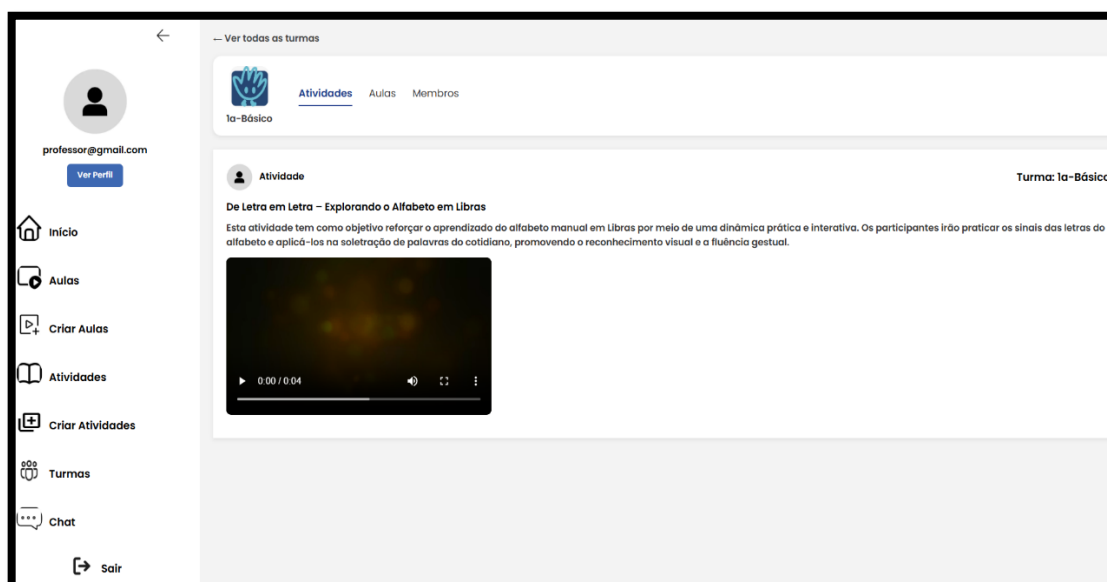
Logo abaixo, existe um submenu com opções de navegação, como "Meus Dados", "Meus Professores", "Publicações", "Fórum", "Chat", entre outros, que redirecionam o usuário para diferentes seções do sistema ao clicar em cada item. Em seguida, há um conteúdo principal da página, que exibe uma seção de turmas vinculadas ao usuário. Essa parte do código verifica, por meio de um *loop foreach*, se existem turmas associadas à variável `$turmas`. Caso existam, elas são listadas como

links, com base no nome e ID de cada turma. Caso contrário, é exibida a mensagem: "Nenhuma turma encontrada."

Na parte inferior do código, encontra-se um *script* JavaScript que permite alternar (expandir ou recolher) o menu lateral. Quando o botão identificado por *toggle* é clicado, ele adiciona ou remove a classe CSS "*collapse*" ao elemento "*sidebar*", fazendo com que o menu seja mostrado ou ocultado visualmente. No geral, este código cria uma interface funcional e dinâmica para o usuário interagir com suas turmas e navegar pelo sistema, combinando lógica do servidor (PHP), marcação HTML e interatividade via JavaScript.

3.5.3.14 – Tela turma selecionada do professor

Figura 105 - Tela turma selecionada do professor



FONTE: AUTORIA PRÓPRIA, 2025

Ao selecionar uma turma, a plataforma direciona automaticamente o professor para a aba Atividades, onde são exibidas as propostas já criadas e publicadas por ele, como ilustrado na figura 105. Essa interface oferece uma visão clara e organizada das atividades vinculadas à turma, facilitando o acompanhamento do conteúdo aplicado. O professor pode visualizar rapidamente o título da atividade, uma descrição explicativa, a data e hora da publicação, e os recursos associados, como vídeos incorporados que enriquecem o processo de ensino. Isso permite uma avaliação imediata da proposta didática desenvolvida e da sua aplicabilidade com os alunos.

Na parte superior da tela, há um menu de navegação em abas com opções como Atividades, Aulas, Minhas Publicações e Membros. Esse cabeçalho facilita o gerenciamento dos diversos aspectos da turma, permitindo ao professor alternar entre as seções com apenas um clique. A aba "Aulas" apresenta o conteúdo teórico ou instrucional associado à turma, enquanto "Minhas Publicações" pode conter avisos, comunicados ou reflexões postadas. Já a aba "Membros" exibe a lista de alunos pertencentes à turma, possibilitando o acompanhamento individual de desempenho ou participação.

A estrutura da interface foi pensada para centralizar a gestão pedagógica em um único ambiente digital, otimizando o tempo do professor e tornando a navegação mais fluida. Dessa forma, ele consegue manter o controle das atividades e do engajamento dos alunos, tudo de maneira acessível, visual e organizada.

Figura 106 - Tela de código da turma selecionada do professor

```

</div>
<!-- Atividades -->
<div id="atividades" class="tab-content">
  <?php if (count($atividades) === 0): ?>
    <div class="empty">Não há atividades cadastradas para esta turma.</div>
  <?php else: foreach ($atividades as $atv): ?>
    <div class="card">
      <div class="card-header">
        <div class="autor">
          
          <div class="info"><strong>Atividade</strong></div>
        </div>
        <span class="badge">Turma: <?php echo htmlspecialchars($nomeTurma); ?></span>
      </div>
      <h5><?php echo htmlspecialchars($atv['nomeAtv']); ?></h5>
      <p><?php echo nl2br(htmlspecialchars($atv['descricaoAtv'] ?? '')); ?></p>
      <?php if (!empty($atv['video'])): ?>
        <div class="video-box">
          <video controls>
            <source src="publicacoes.php?video_atv=<?php echo $atv['id_atv']; ?>" type="video/mp4">
          </video>
        </div>
      <?php endif; ?>
    </div>
  <?php endforeach; endif; ?>
</div>

<!-- Aulas -->
<div id="aulas" class="tab-content" style="display:none">
  <?php if (count($aulas) === 0): ?>
    <div class="empty">Não há aulas cadastradas para esta turma.</div>
  <?php else: foreach ($aulas as $au): ?>
    <div class="card">
      <div class="card-header">
        <div class="autor">
          
          <div class="info"><strong>Aula</strong></div>
        </div>
        <span class="badge">Turma: <?php echo htmlspecialchars($nomeTurma); ?></span>
      </div>
      <h5><?php echo htmlspecialchars($au['nomeAula']); ?></h5>
      <p><?php echo nl2br(htmlspecialchars($au['descricaoAula'] ?? '')); ?></p>
      <?php if (!empty($au['video'])): ?>
        <div class="video-box">
          <video controls><source src="publicacoes.php?video_aula=<?php echo $au['id_aula']; ?>" type="video/mp4"></video>
        </div>
      <?php endif; ?>
    </div>
  <?php endforeach; endif; ?>
</div>

<!-- Membros -->
<div id="membros" class="tab-content" style="display:none">

```

O código exibido na figura 106 é responsável por gerar dinamicamente a exibição de conteúdos vinculados a uma turma específica dentro de um ambiente educacional, mais especificamente atividades e aulas. Utilizando PHP como linguagem de servidor, o *script* realiza verificações, monta estruturas HTML e incorpora recursos multimídia conforme a disponibilidade de dados no banco.

A estrutura se divide em duas seções principais. A primeira seção, com o identificador `id="atividades"`, lida com a exibição de atividades. Ela começa verificando se há atividades disponíveis por meio da função `count($atividades)`. Caso a contagem retorne zero, ou seja, nenhuma atividade cadastrada, é exibida uma mensagem informando essa ausência ao usuário, mantendo a interface clara e objetiva. No entanto, se houver atividades, o código entra em um *loop foreach*, percorrendo cada uma delas e montando blocos de conteúdo com as informações relevantes.

Cada bloco de atividade apresenta dados como o nome da turma associada, o nome da atividade, uma breve descrição, a data de envio formatada com `nl2br()` e, se houver vídeo vinculado, um *player* de vídeo HTML5. Esse *player* é carregado dinamicamente com base no `id_atv`, através de um arquivo PHP (`publicacoes.php`) que serve o conteúdo correto. A presença condicional do vídeo é verificada antes de exibir o *player*, evitando falhas na renderização do conteúdo.

A segunda parte do código é voltada para as aulas e está identificada por `id="aulas"`. Ela segue uma lógica bastante similar à das atividades, porém é inicialmente ocultada com `style="display: none;"`. Isso indica que a exibição dessa aba depende de uma interação do usuário, normalmente feita via JavaScript, como ao clicar em uma aba de navegação. Assim como nas atividades, o *script* verifica se existem aulas disponíveis com `count($aulas)` e, caso não haja nenhuma, uma mensagem informativa é apresentada. Se houver dados, um *foreach* percorre as aulas disponíveis, exibindo nome da turma, título da aula, descrição, e um *player* de vídeo com o conteúdo respectivo, que também é carregado dinamicamente com base no `id_aula`.

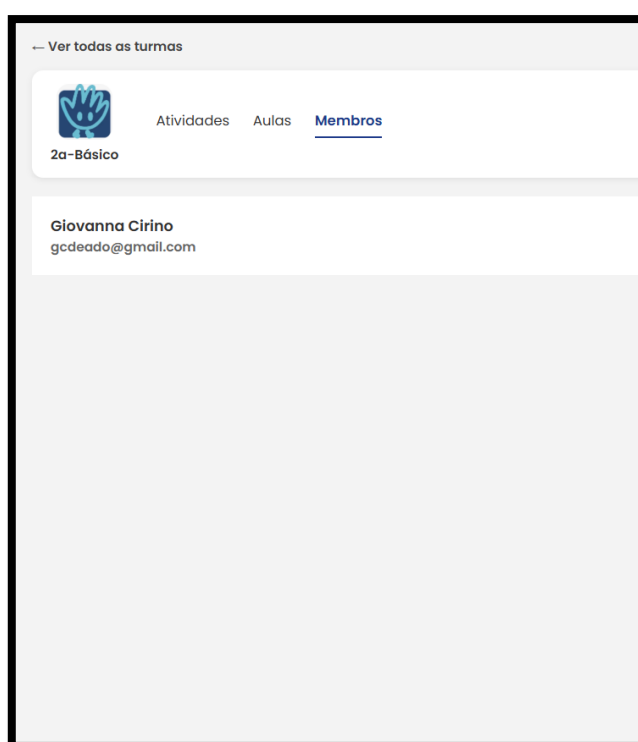
Além das funcionalidades de exibição, o código adota boas práticas como o uso da função `htmlspecialchars()` para escapar caracteres especiais e evitar vulnerabilidades do *Cross-Site Scripting*, garantindo que os dados sejam apresentados de forma segura. O uso do *player* `<video>` embutido proporciona uma

experiência multimídia acessível diretamente na interface do sistema, permitindo que professores e alunos visualizem os conteúdos sem sair da plataforma.

Em essência, o código permite que o ambiente educacional ofereça uma interface rica e funcional, apresentando conteúdos relacionados às atividades e aulas de forma condicional e organizada, conforme a disponibilidade dos dados e a interação do usuário com o sistema.

3.5.3.15 – Tela de membros da turma do professor

Figura 107 - Tela turma membros do professor



FONTE: AUTORIA PRÓPRIA, 2025

A tela ilustrada na figura 107 mostrada apresenta a interface da turma “2a-Básico”, destacando um menu de navegação com quatro opções principais: Atividades, Aulas, Membros e uma opção para voltar à lista geral de turmas. No exemplo, a aba selecionada é “Membros”, onde aparece a listagem dos participantes que fazem parte dessa turma específica. A lista exhibe informações importantes como o nome do aluno, no caso “Giovanna Cirino”, e seu *e-mail* de contato “gcdeado@gmail.com”.

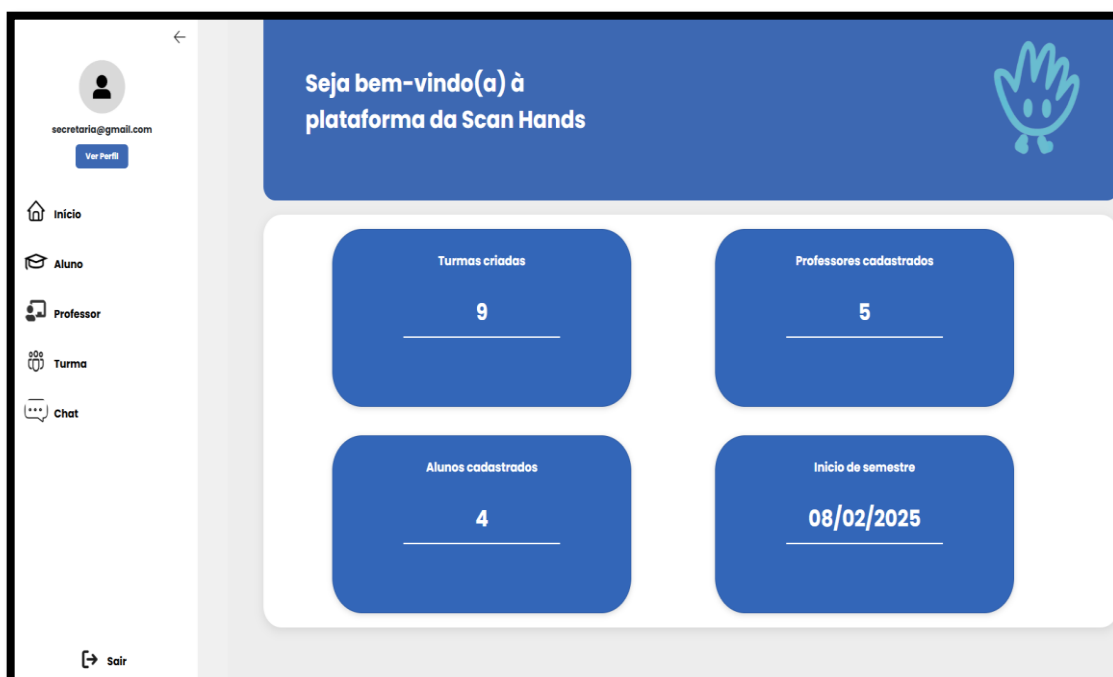
O *design* da interface é simples e organizado, facilitando a navegação do professor ou administrador entre as diferentes seções da turma. A disposição clara

Após a renderização dos membros, um bloco de código JavaScript adiciona dinamismo à interface, selecionando todos os elementos com a classe `.tab-link`. Um evento de clique é atribuído a cada um desses elementos, que ao ser acionado, remove a classe `active` dos botões das abas anteriores e oculta o conteúdo de todas as seções com a classe `.tab-content`. Depois disso, o `script` identifica qual aba foi clicada através do atributo `data-tab` e exibe apenas o conteúdo correspondente, aplicando `style.display = "block"`.

Por fim, o `script` PHP encerra a conexão com o banco de dados utilizando `$conn->close()`, liberando recursos do servidor. Dessa forma, o código não só garante a exibição segura e organizada dos membros da turma, como também provê uma navegação fluida e responsiva entre as diferentes abas da interface.

3.5.4 – Tela inicial da Secretaria

Figura 109 – tela inicial da secretaria



FONTE: AUTORIA PRÓPRIA, 2025

Na figura 109 acima, é apresentada a tela inicial da secretaria no sistema, que tem como objetivo fornecer uma visão geral dos principais indicadores acadêmicos. O painel exibe, em formato de cartões, informações como a quantidade de turmas criadas, professores cadastrados, alunos registrados e a data de início do semestre. Esses elementos permitem que a secretaria acompanhe, de maneira rápida e

organizada, o panorama geral da instituição, facilitando a gestão e o acesso às informações mais relevantes logo na entrada do sistema.

Figura 110 – Tela de código da tela inicial da secretaria

```
// Buscar imagem de perfil
$stmt = $conn->prepare("SELECT foto_perfil FROM usuario WHERE email = ?");
$stmt->bind_param("s", $emailLogado);
$stmt->execute();
$result = $stmt->get_result();

$fotoBase64 = "../img/iconPerfil.png"; // imagem padrão
if ($result && $result->num_rows > 0) {
    $usuario = $result->fetch_assoc();
    if (!empty($usuario['foto_perfil'])) {
        $fotoBase64 = 'data:image/jpeg;base64,' . base64_encode($usuario['foto_perfil']);
    }
}
$stmt->close();

// Consulta: total de turmas
$sqlTurmas = "SELECT COUNT(*) as total FROM turma";
$resultTurmas = $conn->query($sqlTurmas);
$turmas = ($resultTurmas && $resultTurmas->num_rows > 0) ? $resultTurmas->fetch_assoc()['total'] : 0;

// Consulta: total de professores
$sqlProfessores = "SELECT COUNT(*) as total FROM usuario WHERE tipo = 'professor'";
$resultProfessores = $conn->query($sqlProfessores);
$professores = ($resultProfessores && $resultProfessores->num_rows > 0) ? $resultProfessores->fetch_assoc()['total'] : 0;

// Consulta: total de alunos
$sqlAlunos = "SELECT COUNT(*) as total FROM usuario WHERE tipo = 'aluno'";
$resultAlunos = $conn->query($sqlAlunos);
$alunos = ($resultAlunos && $resultAlunos->num_rows > 0) ? $resultAlunos->fetch_assoc()['total'] : 0;

// Consulta: total de dados (exemplo: somando usuários + turmas)
$sqlTotalUsuarios = "SELECT COUNT(*) as total FROM usuario";
$resultTotalUsuarios = $conn->query($sqlTotalUsuarios);
$totalUsuarios = ($resultTotalUsuarios && $resultTotalUsuarios->num_rows > 0) ? $resultTotalUsuarios->fetch_assoc()['total'] : 0;

$dadosGerais = $totalUsuarios + $turmas;

$conn->close();
?>
```

FONTE: AUTORIA PRÓPRIA, 2025

Este código PHP exibido na figura 110 realiza uma série de operações de consulta a um banco de dados MySQL para coletar informações estatísticas e de perfil de usuários de um sistema. Inicialmente, o *script* prepara uma consulta para buscar a imagem de perfil de um usuário com base no *e-mail* fornecido (armazenado na variável *\$emailLogado*). Utiliza-se uma instrução preparada (*prepare*) para evitar ataques de SQL Injection. Caso a consulta retorne resultados, o código verifica se o campo “foto_perfil” está preenchido. Se estiver, a imagem é convertida para o formato *base64* e armazenada em *\$fotoBase64*. Caso contrário, uma imagem padrão (*iconPerfil.png*) é utilizada.

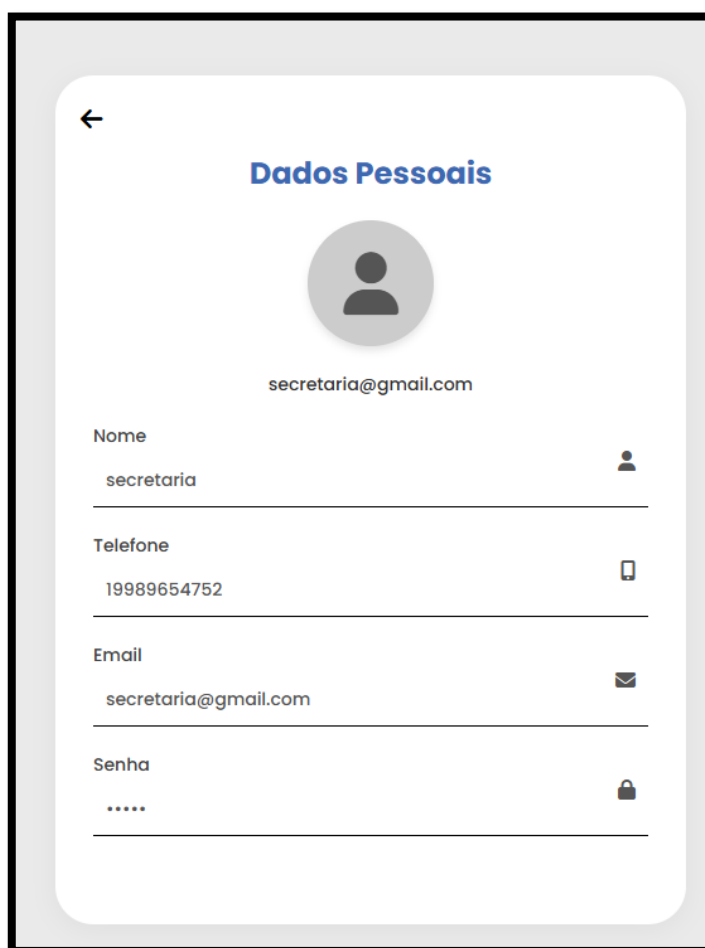
Após essa parte, o código realiza três consultas distintas para contar registros nas tabelas: A primeira consulta retorna o número total de turmas existentes na tabela *turma*, a segunda busca a quantidade de usuários com o tipo “professor” na tabela *usuario* e a terceira retorna a quantidade de usuários com o tipo “aluno”, também na tabela *usuario*. Cada uma dessas contagens é armazenada em variáveis separadas

(\$turmas, \$professores, \$alunos) e só é atribuída se os resultados existirem e possuírem linhas.

Por fim, o código executa mais uma consulta para contar o total de usuários cadastrados na tabela usuário, independentemente do tipo. Essa contagem é armazenada em \$totalUsuarios. Depois, a variável \$dadosGerais é calculada somando o total de usuários com o total de turmas, representando um dado agregado do sistema. O *script* termina fechando a conexão com o banco de dados, garantindo o uso eficiente dos recursos.

3.5.4.1 - Tela de visualizar perfil da secretaria

Figura 111 – Tela visualizar perfil da secretaria



FONTE: AUTORIA PRÓPRIA, 2025

Na figura 111 acima, é exibida a tela de Dados Pessoais da Secretaria, responsável por apresentar e permitir a gestão das informações cadastrais do usuário logado. A interface contém campos como nome, telefone, e-mail e senha, além da foto

de perfil, que centraliza a identidade visual do usuário. Essa tela possibilita a atualização dos dados de forma prática e organizada, garantindo que as informações da secretaria permaneçam corretas e atualizadas dentro do sistema.

Figura 112 - Tela do código visualizar perfil da secretaria

```

<?php
session_start();

// Redireciona se o usuário não estiver logado
if (!isset($_SESSION['email'])) {
    header("Location: ../../index.php");
    exit();
}

// Conecta ao banco
$host = "localhost";
$usuario = "u357936358_librinha";
$senha = "Librinh4tcc#";
$banco = "u357936358_scanhands";

$conn = new mysqli($host, $usuario, $senha, $banco);
if ($conn->connect_error) {
    die("Erro de conexão com o banco: " . $conn->connect_error);
}

$email = $_SESSION['email'];
$nome = $telefone = $senhaUsuario = '';
$fotoBase64 = '';

// Busca dados da secretaria
$stmt = $conn->prepare("
    SELECT nome, telefone, senha, foto_perfil
    FROM usuario
    WHERE email = ? AND tipo = 'secretaria'
");
$stmt->bind_param("s", $email);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows > 0) {
    $dados = $result->fetch_assoc();
    $nome = $dados['nome'];
    $telefone = $dados['telefone'];
    $senhaUsuario = $dados['senha'];
    $fotoBase64 = !empty($dados['foto_perfil']) ? 'data:image/jpeg;base64,' . base64_encode($dados['foto_perfil']) : '';
} else {
    $nome = "Desconhecido";
}

$conn->close();
?>

```

FONTE: AUTORIA PRÓPRIA, 2025

Este código exibido na figura 112 em PHP tem como principal objetivo autenticar e buscar informações da conta de uma secretária em um sistema *web*. Ele é dividido em etapas distintas que garantem segurança, conexão com banco de dados e recuperação de dados do usuário logado. A execução começa com a função `session_start()`, que inicia uma sessão PHP para acessar variáveis de sessão. Em seguida, o *script* verifica se a sessão contém um *e-mail* válido, o que indica que o usuário está logado.

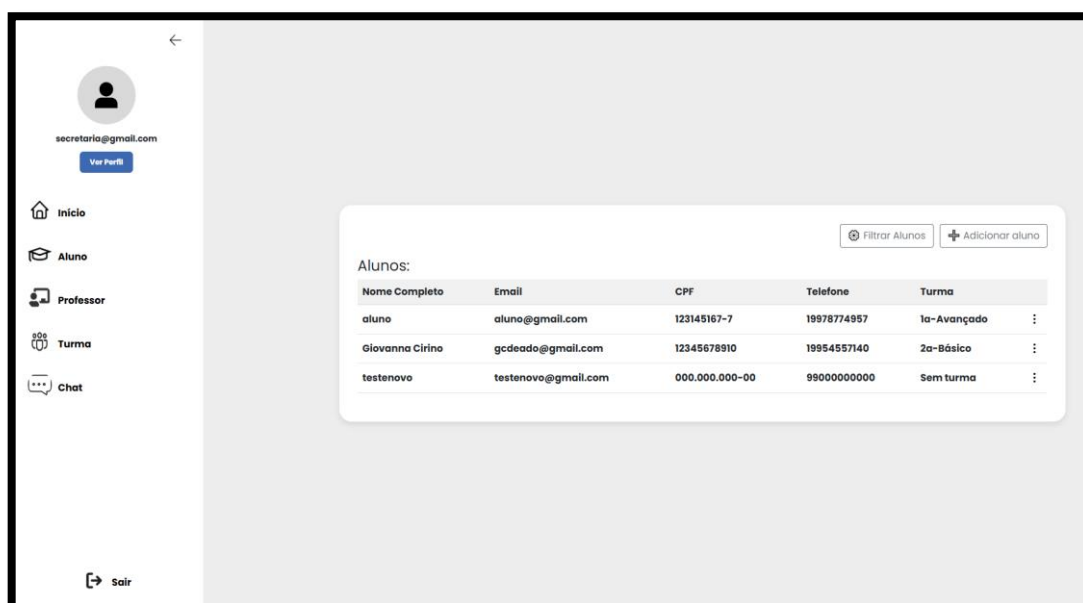
Caso não esteja, ele é redirecionado para a página de *login* (`index.php`) e a execução é encerrada. Logo após, são definidos os parâmetros para conexão com o banco de dados, incluindo *host*, usuário, senha e nome do banco. Utiliza-se a classe

mysqli para estabelecer a conexão e, caso haja falha, é exibida uma mensagem de erro informando o problema na conexão. O *e-mail* do usuário logado é então recuperado da sessão e armazenado na variável `$email`. Outras variáveis (`$nome`, `$telefone`, `$senhaUsuario`, `$fotoBase64`) são inicializadas para receber os dados posteriormente. A parte seguinte prepara uma consulta SQL com segurança utilizando `prepare()` e `bind_param()` para evitar injeção de SQL. A consulta seleciona o nome, telefone, senha e foto de perfil da tabela usuário, onde o *e-mail* corresponde ao informado e o tipo de usuário seja "secretaria". Após executar a consulta, o código verifica se há resultados. Se houver dados, eles são extraídos e atribuídos às variáveis correspondentes.

A foto de perfil, se existir, é convertida para *base64* para ser usada diretamente em um elemento HTML ``. Caso contrário, uma *string* vazia é atribuída. Se não forem encontrados dados no banco, o nome é definido como "Desconhecido". Ao final, a conexão com o banco é encerrada com `$conn->close()`; garantindo liberação dos recursos. Este *script* é um exemplo típico de autenticação e extração de dados seguros em aplicações *web* baseadas em PHP e MySQL.

3.5.4.2 - Tela alunos da secretaria

Figura 113 – Tela alunos da secretaria



FONTE: AUTORIA PRÓPRIA, 2025

Na figura 113 acima, é apresentada a tela de Alunos da Secretaria, que exhibe em formato de tabela os dados essenciais de cada estudante cadastrado na

plataforma. São listadas informações como nome completo, *e-mail*, CPF, telefone e turma associada, permitindo uma visão clara e organizada. No canto superior direito, estão disponíveis os botões “Filtrar Alunos” e “Adicionar aluno”, que possibilitam, respectivamente, aplicar filtros de pesquisa para localizar registros específicos ou incluir novos estudantes no sistema. Além disso, cada linha da tabela contém um menu de opções representado por três pontos, que oferece ações adicionais como editar, excluir ou visualizar detalhes do aluno selecionado

Figura 114 – Tela de código alunos da secretaria

```
// Monta a consulta com filtros
$conditions = ["u.tipo = 'aluno'"];
$params = [];

$campos = ['nome', 'turma', 'cpf', 'telefone', 'email'];
foreach ($campos as $campo) {
    if (!empty($_GET[$campo])) {
        if ($campo === 'turma') {
            $conditions[] = "t.nome LIKE ?";
        } else {
            $conditions[] = "u.$campo LIKE ?";
        }
        $params[] = "%" . $_GET[$campo] . "%";
    }
}

$whereClause = implode(" AND ", $conditions);

// Query agora com JOIN entre usuario, aluno, aluno_turma e turma
$sql = "
    SELECT
        u.nome,
        u.email,
        u.cpf,
        u.telefone,
        GROUP_CONCAT(DISTINCT t.nome SEPARATOR ', ') AS turma
    FROM usuario u
    INNER JOIN aluno a ON u.id_usuario = a.id_usuario
    LEFT JOIN aluno_turma at ON a.id_aluno = at.id_aluno
    LEFT JOIN turma t ON at.id_turma = t.id_turma
    WHERE $whereClause
    GROUP BY u.id_usuario
";

$stmt = $conn->prepare($sql);

if (!empty($params)) {
    $types = str_repeat("s", count($params));
    $stmt->bind_param($types, ...$params);
}

$stmt->execute();
$result = $stmt->get_result();
?>
```

FONTE: AUTORIA PRÓPRIA, 2025

Este código PHP exibido na figura 114, tem como objetivo realizar uma consulta dinâmica ao banco de dados para buscar informações de alunos cadastrados no sistema, aplicando filtros opcionais fornecidos via parâmetros *GET* (normalmente passados pela URL ou por formulários HTML). A lógica inicia definindo uma condição

base onde apenas registros de usuários cujo tipo seja "aluno" serão considerados. Em seguida, é criado um *array* de campos possíveis para filtragem — incluindo nome, turma, CPF, telefone e *e-mail* — e para cada um desses campos, o código verifica se um valor foi enviado. Caso haja um valor, uma cláusula *LIKE* é adicionada ao *array* de condições. Quando o campo for "turma", a condição é aplicada à tabela de turmas, pois ela está relacionada à tabela de usuários por meio de *joins*. Após a montagem de todas as condições, elas são unidas por operadores *AND* e formam a cláusula *WHERE* da consulta.

A consulta SQL propriamente dita é então construída com diversos *JOINS*: um *INNER JOIN* liga a tabela *usuario* à tabela *aluno*, enquanto dois *LEFT JOINS* conectam as tabelas *aluno_turma* e *turma*. Isso permite recuperar os dados do aluno mesmo que ele não esteja associado a nenhuma turma. Além disso, utiliza-se "*GROUP_CONCAT*" para listar os nomes das turmas em uma única linha, separadas por vírgulas. Com a *query* montada, o código prepara a instrução SQL utilizando *prepared statements*, garantindo a segurança contra injeções de SQL. Os parâmetros coletados são vinculados à consulta usando *bind_param*, e em seguida a consulta é executada. Por fim, os resultados são recuperados com "*get_resul*()" para posterior processamento ou exibição.

Esse código é um exemplo eficiente e seguro de como implementar buscas avançadas com múltiplos filtros em sistemas de gerenciamento acadêmico. Sua aplicação possibilita maior flexibilidade no acesso às informações, tornando o processo de consulta mais rápido e intuitivo. Além disso, por adotar boas práticas de programação e segurança, contribui para a confiabilidade e escalabilidade do sistema, permitindo futuras expansões sem comprometer a integridade dos dados ou a performance do banco.

Outro ponto relevante é que a estrutura modular utilizada favorece a manutenção do código, uma vez que novos filtros ou relacionamentos podem ser adicionados sem a necessidade de grandes alterações na lógica principal. Isso garante que a solução acompanhe a evolução das necessidades da instituição de ensino, assegurando longevidade e adaptabilidade ao sistema. Dessa forma, o recurso não apenas otimiza a experiência de uso, como também representa um investimento estratégico em qualidade e sustentabilidade tecnológica.

3.5.4.3 - Tela professores da secretaria

Figura 115 – Tela professores da secretaria

secretaria@gmail.com
Ver Perfil

Início
Aluno
Professor
Turma
Chat

Filtrar Professor Adicionar Professor

Professores:

Nome Completo	Email	CPF	Telefone	Turmas	
professor	professor@gmail.com	45678978945	19995865473	1a-Básico, 2a-Básico	⋮
nilton	nilton@gmail.com	12345678915	19949461052	sem turma	⋮
professor2	professor2@gmail.com	1231458677	199787746837	2a-Avançado	⋮
professor3	professor3@gmail.com	583.623.228-59	19978723957	1a-Básico, 2a-Avançado	⋮
alan	alan@gmail.com	42562139968	19954557144	1a-Básico, 2a-Básico	⋮

FONTE: AUTORIA PRÓPRIA, 2025

Na figura 115 acima, é apresentada a tela de Professores da Secretaria, que exibe em formato de tabela os dados cadastrados de cada docente. São listadas informações como nome completo, *e-mail*, CPF, telefone e as turmas associadas, permitindo uma gestão clara e organizada do corpo docente. No canto superior direito, encontram-se os botões “Filtrar Professor” e “Adicionar Professor”, que possibilitam, respectivamente, aplicar critérios de busca específicos ou cadastrar novos professores na plataforma. Além disso, cada linha da tabela contém um menu de opções representado por três pontos, que oferece ações adicionais como editar, excluir ou visualizar detalhes do professor selecionado.

Essa tela tem como objetivo central facilitar a administração dos professores vinculados à instituição, fornecendo à Secretaria Escolar uma interface intuitiva e funcional para o controle do corpo docente. O recurso de filtragem permite que a busca seja refinada por atributos específicos, otimizando o tempo de localização de registros em contextos de grande volume de dados. Da mesma forma, a possibilidade de adicionar professores diretamente pela interface contribui para a agilidade no processo de cadastro e atualização da base de dados. O menu de ações individuais garante maior flexibilidade, uma vez que possibilita ajustes pontuais sem a

necessidade de navegar por diversas telas do sistema. Dessa maneira, a tela de Professores se apresenta como um módulo essencial para a organização acadêmica, promovendo praticidade, precisão e eficiência no gerenciamento das informações.

Figura 116 – Tela do código professores da secretaria

```

<?php
session_start();
if (!isset($_SESSION['email'])) {
    header("Location: ../../index.php");
    exit();
}
$emailLogado = $_SESSION['email'];

// Conexão com o banco de dados
$servername = "localhost";
$username = "u357936358_librinha";
$password = "Librinh4tcc#";
$dbname = "u357936358_scanhands";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Erro na conexão: " . $conn->connect_error);
}

// Limpar filtros
if (!empty($_GET['limpar'])) {
    header("Location: visuProf.php");
    exit();
}

// Monta a consulta com filtros
$conditions = ["u.tipo = 'professor'"];
$params = [];

$campos = ['nome', 'turma', 'cpf', 'telefone', 'email'];
foreach ($campos as $campo) {
    if (!empty($_GET[$campo])) {
        if ($campo === 'turma') {
            $conditions[] = "t.nome LIKE ?";
        } else {
            $conditions[] = "u.$campo LIKE ?";
        }
        $params[] = "%" . $_GET[$campo] . "%";
    }
}

$whereClause = implode(" AND ", $conditions);

// Query com filtros e GROUP_CONCAT
$sql = "
SELECT
    p.id_professor,
    u.nome AS nome_professor,
    u.email,
    u.cpf,
    u.telefone,
    COALESCE(GROUP_CONCAT(DISTINCT t.nome ORDER BY t.id_turma SEPARATOR ', '), 'sem turmas') AS turmas
FROM professor p
JOIN usuario u ON u.id_usuario = p.id_usuario
LEFT JOIN professor_turma pt ON pt.id_professor = p.id_professor
LEFT JOIN turma t ON t.id_turma = pt.id_turma
WHERE $whereClause
GROUP BY p.id_professor
";

```

FONTE: AUTORIA PRÓPRIA, 2025

Este código exibido na figura 116 em PHP tem como finalidade realizar uma consulta filtrada no banco de dados para listar professores e suas respectivas turmas em um sistema educacional. A execução inicia-se com a chamada à função `session_start()` para garantir o acesso às variáveis de sessão.

Em seguida, o *script* verifica se existe um *e-mail* armazenado na sessão, o que confirma que o usuário está autenticado. Caso contrário, ocorre um redirecionamento para a página de *login* e a execução do *script* é encerrada. Uma vez autenticado, o *script* estabelece conexão com o banco de dados utilizando as credenciais fornecidas. Se houver falha na conexão, uma mensagem de erro é exibida. O *e-mail* do usuário logado é então armazenado em uma variável, embora não seja utilizado diretamente na lógica a seguir.

O código também trata a possibilidade de "limpar" os filtros de busca. Se o parâmetro *limpar* for detectado na URL ($\$_GET$), o usuário é redirecionado para a página padrão de visualização de professores (*visuProf.php*) e o *script* é encerrado. Na sequência, o código monta dinamicamente os filtros de busca. Um *array* de condições é iniciado com uma cláusula fixa que limita os resultados aos usuários do tipo "professor". Em paralelo, um *array* de parâmetros é preparado para armazenar os valores digitados pelo usuário. A lista de campos que podem ser filtrados inclui nome, turma, CPF, telefone e *e-mail*.

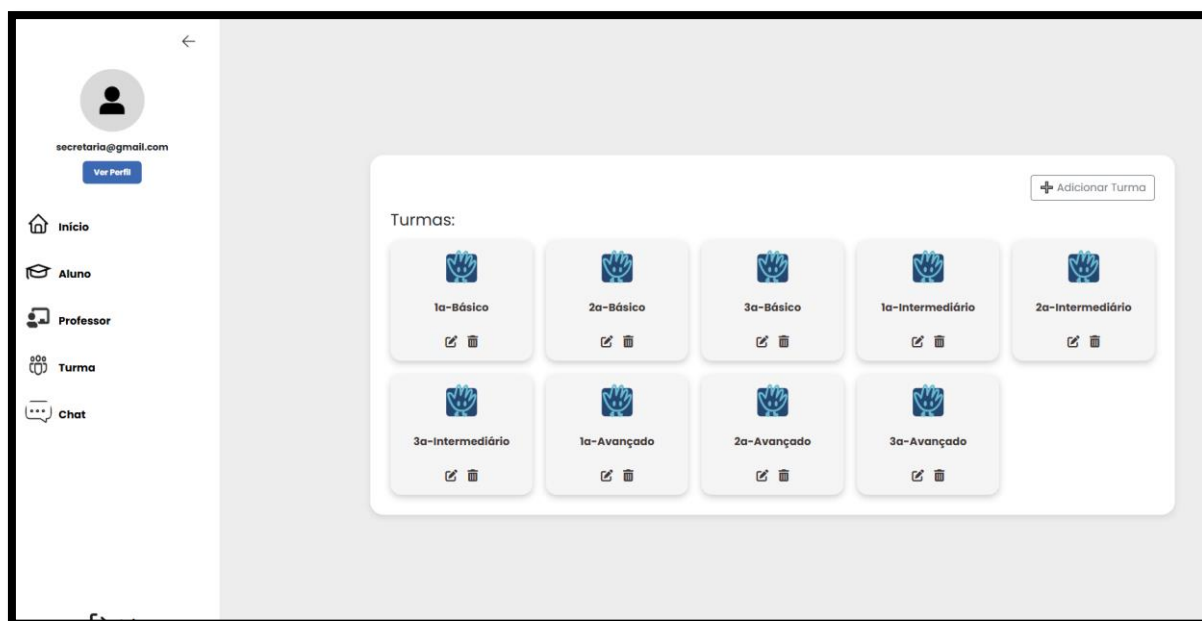
Para cada campo, se houver valor informado via *GET*, uma condição *LIKE* é adicionada ao *array* de condições e o valor correspondente é formatado com curingas (%) para permitir buscas parciais. Essas condições são então unidas por *AND* e formam a cláusula *WHERE* da consulta SQL. A *query* é construída para selecionar informações da tabela de professores (*professor*) junto com a tabela de usuários (*usuario*), turmas (*turma*) e a tabela intermediária *professor_turma*.

A consulta utiliza *JOINS* para combinar as tabelas de forma relacional, e emprega a função *GROUP_CONCAT* para agrupar os nomes das turmas associadas a cada professor. Caso o professor não esteja vinculado a nenhuma turma, a função *COALESCE* garante que o valor exibido seja "sem turma".

Por fim, a consulta é preparada com segurança usando *prepare()* e, se existirem parâmetros, são vinculados com *bind_param()*. A execução da consulta é realizada com *execute()*, e os resultados são obtidos com *get_result()* para posterior exibição. Este *script* representa uma solução robusta para realizar buscas refinadas de professores no sistema, permitindo múltiplos filtros ao mesmo tempo e exibindo claramente os vínculos com turmas, inclusive nos casos em que não há vínculo algum.

3.5.4.4 - Tela de turmas da secretaria

Figura 117 – Tela de turmas da secretaria



FONTE: AUTORIA PRÓPRIA, 2025

Na figura 117 acima, é exibida a tela de Turmas da Secretaria, responsável por organizar e apresentar todas as turmas cadastradas no sistema. Cada turma é representada em formato de cartão, contendo o nome correspondente e ícones de ação que permitem editar, excluir ou visualizar os detalhes da turma selecionada. No canto superior direito, está disponível o botão “Adicionar Turma”, que possibilita o cadastro de novas turmas de forma rápida e prática. Essa interface garante uma gestão centralizada e intuitiva, facilitando a administração acadêmica e a navegação da secretaria.

Além disso, a disposição visual em cartões promove maior clareza na identificação das turmas, facilitando a compreensão e reduzindo erros na manipulação dos dados. Essa abordagem melhora a experiência do usuário ao tornar a navegação mais objetiva e dinâmica, permitindo que a secretaria escolar execute suas tarefas com mais agilidade e precisão. Outro ponto positivo é que o *layout* em cartões favorece a escalabilidade da interface, permitindo que novas funcionalidades, como filtros, cores ou indicadores de *status*, sejam incorporadas de forma simples, acompanhando as futuras demandas da instituição sem comprometer a usabilidade do sistema.

Figura 118 – Tela de código das turmas da secretaria

```

<div class="sidebar" style="margin-right: 50px;">
  <div class="sidebar-top">
    <button id="toggleSidebar">
      
    </button>
    
    <p class="email"><?php echo htmlspecialchars($emailLogado); ?></p>
    <button style="font-size: 12px;" onclick="window.location.href='perfilSecretaria.php'">Ver Perfil</button>
    <nav>
      <a href="inicioSecretaria.php"><span> Início</span></a>
      <a href="visuAlunos.php"><span> Aluno</span></a>
      <a href="visuProf.php"><span> Professor </span></a>
      <a href="visuTurma.php"><span> Turma </span></a>
      <a href="#"><span> Chat</span></a>
    </nav>
  </div>
  <a class="logout" href="../../../../index.php">
    
    <span> Sair</span>
  </a>
</div>

<div class="main">
  <div class="card-container">
    <div class="header-buttons">
      <button class="btn btn-outline-secondary" onclick="window.location.href='adicionaTurma.php'">+ Adicionar Turma</button>
    </div>
    <h4 style="margin-bottom: 15px;">Turmas:</h4>
    <div class="turma-grid">
      <?php if ($result && $result->num_rows > 0): ?>
        <?php while($row = $result->fetch_assoc()): ?>
          <div class="turma-card" onclick="window.location.href='turma.php?id=<?= urlencode($row['id_turma']) ?>'>
            
            <div class="turma-nome"><?= htmlspecialchars($row["nome"]) ?></div>
            <div class="turma-actions">
              <a href="alterarTurma.php?nome=<?= urlencode($row["nome"]) ?>" onclick="event.stopPropagation();"><i class="fas fa-edit"></i></a>
              <a href="excluirTurma.php?nome=<?= urlencode($row["nome"]) ?>" onclick="event.stopPropagation();"><i class="fas fa-trash-alt"></i></a>
            </div>
          </div>
        <?php endwhile; ?>
      <?php else: ?>
        <p style="text-align: center;">Nenhuma turma encontrada.</p>
      <?php endif; ?>
    </div>
  </div>
</div>

<script>
const toggleBtn = document.getElementById('toggleSidebar');
const sidebar = document.querySelector('.sidebar');
toggleBtn.addEventListener('click', () => {
  sidebar.classList.toggle('collapsed');
});
</script>
</body>
</html>
<?php $conn->close(); ?>

```

FONTE: AUTORIA PRÓPRIA, 2025

Este código PHP exibido na figura 118, integrado com HTML, CSS e JavaScript, compõe a interface de um painel administrativo onde a secretária pode visualizar e gerenciar as turmas cadastradas no sistema. A estrutura está dividida em duas seções principais: a barra lateral de navegação e o conteúdo principal da página.

Na parte lateral, é exibido um menu com botões de navegação para outras seções do sistema, como Início, Aluno, Professor, Turma e *Chat*. Também há um botão de “Sair”, que redireciona para a página de *logout*. O nome da secretária logada é exibido dinamicamente por meio da variável *\$emailLogado*, e existe um botão que leva ao seu perfil pessoal, permitindo acessar ou alterar suas informações cadastrais. Um botão com a imagem de “Voltar” também está presente, permitindo retornar à tela anterior de forma prática. Além disso, o menu lateral é responsivo e pode ser

expandido ou recolhido por meio de um botão de alternância que utiliza JavaScript. Isso melhora a usabilidade em dispositivos com diferentes resoluções de tela, tornando a interface mais adaptável.

No conteúdo principal (*main*), o título "Turmas" é exibido, acompanhado de um botão para adicionar uma nova turma, que redireciona para o formulário correspondente (*adicionaTurma.php*). Abaixo disso, o sistema realiza uma consulta ao banco de dados para obter as turmas cadastradas. Caso existam turmas, o *script* entra em um laço *while* que percorre os resultados e cria *cards* para cada uma delas, exibindo o nome da turma e uma imagem ilustrativa que representa visualmente o conteúdo.

Cada *card* contém dois botões de ação: um botão de edição, que redireciona para a tela de alteração da turma com base no nome; e um botão de exclusão, que leva ao script responsável por excluir a turma, também utilizando o nome como identificador. Ambos os botões possuem *onclick* com *event.stopPropagation()* para evitar que o clique neles redirecione para a página da turma (ação do *card* principal). Se nenhuma turma for encontrada, é exibida a mensagem "Nenhuma turma encontrada." com um estilo centralizado, garantindo uma melhor experiência visual mesmo quando os dados estiverem ausentes.

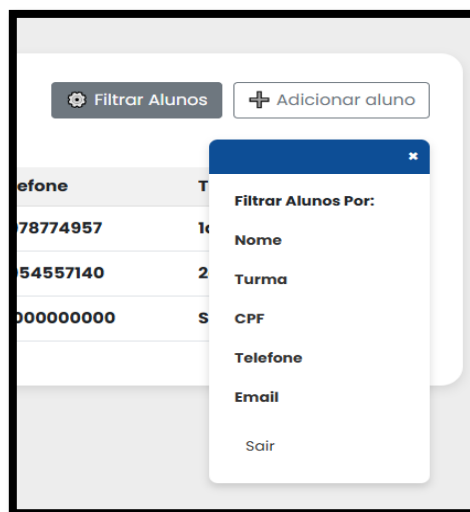
Ao final do *script*, há um trecho de JavaScript responsável por controlar o comportamento do botão que recolhe ou expande a barra lateral. Isso é feito alternando a classe CSS *collapsed* na *sidebar*, criando uma interface mais interativa e moderna. Esse comportamento melhora a experiência do usuário e ajuda a manter o foco no conteúdo principal da página.

Por fim, a conexão com o banco de dados é encerrada com *\$conn->close()*, garantindo que os recursos sejam liberados corretamente. Este código representa uma interface administrativa funcional, responsiva e amigável para que a secretária possa gerenciar turmas com facilidade e eficiência, integrando múltiplas tecnologias de forma harmoniosa. Além disso, a aplicação foi estruturada para permitir manutenção e futuras expansões sem comprometer a estabilidade do sistema, oferecendo maior segurança e confiabilidade. O uso de boas práticas no desenvolvimento favorece o desempenho e reduz falhas durante o uso diário. Essa

abordagem também facilita o treinamento de novos colaboradores e a integração de outros módulos administrativos, fortalecendo ainda mais a gestão escolar.

3.5.4.5 – Filtrar alunos da secretaria

Figura 119 – Tela de filtrar alunos da secretaria



FONTE: AUTORIA PRÓPRIA, 2025

Na figura 119 acima, é apresentada a tela de Filtrar Alunos, que possibilita à secretaria localizar estudantes de forma mais rápida e precisa. O painel de filtro oferece critérios como nome, turma, CPF, telefone e *e-mail*, permitindo buscas específicas dentro do sistema. Além disso, há a opção “Sair”, que encerra a janela de filtragem e retorna à visualização principal. Essa funcionalidade contribui para a organização dos dados e para a otimização da gestão.

O uso de múltiplos critérios de busca proporciona maior flexibilidade no atendimento às demandas diárias, facilitando o acesso às informações mesmo em grandes volumes de dados. A interface é clara e objetiva, permitindo que o processo de filtragem seja feito de forma intuitiva por qualquer usuário. Com isso, reduz-se o tempo gasto em tarefas operacionais, melhorando a eficiência do setor. A filtragem ágil também é fundamental em períodos de matrícula, rematrícula ou atualização cadastral, quando a rapidez na localização de dados é essencial.

Além disso, a presença visual do botão de filtro e sua acessibilidade tornam a navegação mais prática, promovendo uma experiência de uso mais agradável. Esse recurso também contribui para evitar erros, como consultas equivocadas ou duplicidade de registros. A centralização dos critérios em um único menu facilita a

padronização das buscas, permitindo que diferentes usuários adotem os mesmos parâmetros de consulta. Assim, a funcionalidade de filtragem não só melhora o fluxo de trabalho, como também fortalece a confiabilidade do sistema como um todo.

Figura 120 – Tela de código de filtrar alunos da secretaria

```

<?php
session_start();
if (!isset($_SESSION['email'])) {
    header("Location: ../../index.php");
    exit();
}
$emailLogado = $_SESSION['email'];

// Conexão com o banco de dados
$servername = "localhost";
$username = "u357936358_librinha";
$password = "Librinh4tcc#";
$dbname = "u357936358_scanhands";

$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Erro na conexão: " . $conn->connect_error);
}

// Limpar filtros
if (!empty($_GET['limpar'])) {
    header("Location: visuAlunos.php");
    exit();
}

// Monta a consulta com filtros
$conditions = ["u.tipo = 'aluno'"];
$params = [];

$campos = ['nome', 'turma', 'cpf', 'telefone', 'email'];
foreach ($campos as $campo) {
    if (!empty($_GET[$campo])) {
        if ($campo === 'turma') {
            $conditions[] = "t.nome LIKE ?";
        } else {
            $conditions[] = "u.$campo LIKE ?";
        }
        $params[] = "%" . $_GET[$campo] . "%";
    }
}

$whereClause = implode(" AND ", $conditions);

```

FONTE: AUTORIA PRÓPRIA, 2025

Este código PHP exibido na figura 120 tem como função preparar uma consulta dinâmica ao banco de dados com base em filtros fornecidos pelo usuário, visando listar apenas usuários do tipo "aluno". A execução inicia com a chamada de `session_start()` para habilitar o uso de sessões, seguida de uma verificação que garante que o usuário esteja logado. Caso contrário, ocorre um redirecionamento para

a página de *login* e o *script* é encerrado. O *e-mail* do usuário autenticado é armazenado na variável `$emailLogado`, ainda que neste trecho ele não seja utilizado. Logo após, o *script* realiza a conexão com o banco de dados utilizando as credenciais fornecidas.

Se houver erro na conexão, o sistema exibe uma mensagem de erro e interrompe a execução. O código também verifica se o parâmetro *limpar* foi enviado pela URL, o que indicaria que os filtros devem ser limpos. Nesse caso, o usuário é redirecionado para a página de visualização padrão de alunos (`visuAlunos.php`), sem filtros aplicados. Em seguida, a lógica principal de filtragem é executada. Um *array* chamado `$conditions` é iniciado com uma condição fixa para limitar os resultados a registros cujo tipo seja "aluno". Além disso, um *array* vazio chamado `$params` é preparado para armazenar os valores dos filtros informados.

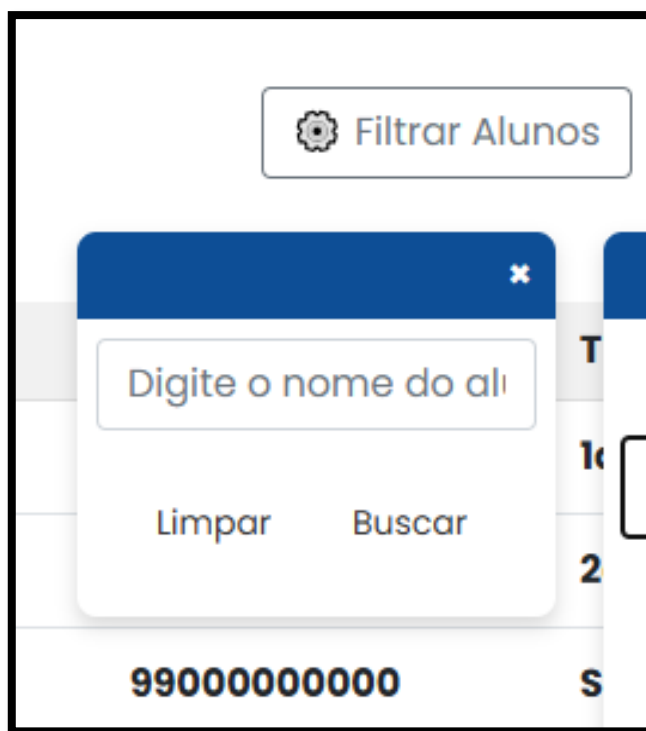
O código define uma lista de campos que podem ser utilizados como filtro: nome, turma, cpf, telefone e *email*. Em um *loop foreach*, cada campo é verificado individualmente: se houver um valor informado para ele via `$_GET`, o código constrói uma condição do tipo *LIKE*, adicionando-a ao *array* de condições. Se o campo for "turma", a condição é aplicada sobre a tabela *turma* (referenciada como `t.nome`). Para os demais campos, a filtragem ocorre na tabela de usuários (referenciada como `u.campo`).

Os valores dos filtros são formatados com `%` para permitir buscas parciais e adicionados ao *array* `$params`. Por fim, todas as condições são unidas em uma única *string* utilizando *implode* (" *AND* ", `$conditions`), gerando a cláusula *WHERE* da consulta SQL. Esta estrutura será utilizada posteriormente para montar a *query* final com *JOINS* entre tabelas (embora esta parte específica da execução não esteja visível na imagem). O resultado é uma consulta flexível, segura e adaptável a múltiplos critérios de busca para alunos.

Esse tipo de abordagem demonstra boas práticas tanto de segurança quanto de escalabilidade, pois a utilização de *prepared statements* junto ao controle dinâmico de condições reduz significativamente os riscos de injeção de SQL e melhora a manutenção do sistema. Além disso, a lógica modular empregada permite que novos filtros sejam incorporados facilmente, garantindo que a aplicação acompanhe o crescimento e as necessidades futuras da instituição de ensino.

3.5.4.6 – Pesquisar alunos da secretaria

Figura 121 – Tela de pesquisar alunos da secretaria



FONTE: AUTORIA PRÓPRIA, 2025

Na figura 121 acima, observa-se a tela de Busca de Alunos, que permite à secretaria localizar estudantes de forma direta por meio de um campo de texto, no qual deve ser digitado o nome do aluno desejado. A interface conta ainda com os botões “Limpar” e “Buscar”: o primeiro redefine os critérios inseridos, enquanto o segundo confirma a pesquisa e exibe os resultados correspondentes. Essa funcionalidade garante rapidez na navegação e praticidade na gestão dos cadastros.

Além disso, a tela de busca oferece um mecanismo intuitivo que minimiza erros e torna o processo de localização de alunos mais eficiente, principalmente em instituições com grande volume de estudantes. A simplicidade do *design*, aliada à clareza das opções de interação, proporciona maior produtividade à equipe da secretaria, permitindo que consultas frequentes sejam realizadas de maneira ágil e organizada.

A presença de *feedback* visual imediato após a busca também contribui para a usabilidade da interface, pois permite que os usuários saibam rapidamente se os dados foram encontrados ou não. Isso reduz a incerteza e evita retrabalhos. Outro

ponto positivo é que a funcionalidade pode ser facilmente integrada a outros filtros ou critérios adicionais no futuro, demonstrando a escalabilidade do sistema. Assim, essa ferramenta reforça a eficiência operacional e melhora significativamente o atendimento prestado pela equipe administrativa.

Figura 122 – Tela de código de pesquisar alunos da secretaria

```
const searchCard = document.getElementById('searchCard');
const searchInput = document.getElementById('searchInput');
const closeSearchBtns = document.querySelectorAll('.close-search');

document.querySelectorAll('.open-search').forEach(btn => {
  btn.addEventListener('click', function(e) {
    e.stopPropagation();
    const field = this.getAttribute('data-field');
    searchInput.name = field;
    searchInput.placeholder = `Digite o ${field === 'cpf' ? 'CPF' : field === 'telefone' ? 'telefone' : field} do aluno`;
    searchInput.value = '';
    searchCard.style.display = 'block';
  });
});

closeSearchBtns.forEach(btn => {
  btn.addEventListener('click', function(e) {
    e.stopPropagation();
    searchCard.style.display = 'none';
  });
});
</script>
</body>
</html>
<?php $conn->close(); ?>
```

FONTE: AUTORIA PRÓPRIA, 2025

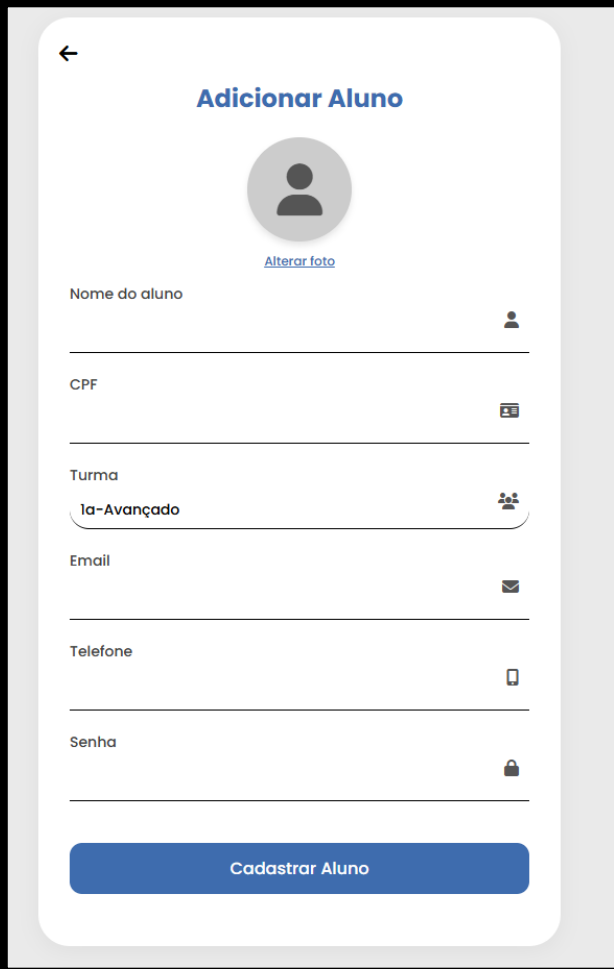
Este código exibido na figura 122 em JavaScript gerencia a exibição e interação de um componente de busca na interface, especificamente um *card* de pesquisa que permite ao usuário buscar informações específicas sobre alunos. Ele começa selecionando elementos do DOM: o *container* do *card* de busca (*searchCard*), o campo de entrada de texto (*searchInput*) e todos os botões que fecham o *card* (*closeSearchBtns*).

Para todos os botões que possuem a classe “.*open-search*”, o código adiciona um evento de clique. Quando clicado, ele previne a propagação do evento, captura o valor do atributo “*data-field*” do botão clicado, que indica o campo a ser buscado (como nome, CPF ou telefone), e ajusta o nome e o *placeholder* do campo de entrada com base nesse valor. O campo de entrada é então limpo (valor zerado) e o *card* de busca é exibido, mudando seu estilo para *display: block*. Já para os botões de fechamento, identificados pela classe “.*close-search*”, é adicionado um evento de clique que, ao ser acionado, também previne a propagação do evento e oculta o *card* de busca, definindo o seu estilo para *display: none*.

Assim, o *script* cria uma interface dinâmica e amigável para que o usuário possa abrir diferentes campos de busca e fechá-los facilmente, personalizando o *placeholder* conforme o tipo de dado que deseja consultar. No final, o código PHP fecha a conexão com o banco de dados, indicando o fim do *script* do lado servidor.

3.5.4.7 - Tela de Adicionar Aluno

Figura 123 – Tela de adicionar aluno da secretaria



A imagem mostra a interface de usuário para adicionar um novo aluno. O título da tela é "Adicionar Aluno". No topo, há um ícone de perfil de usuário e um link "Alterar foto". Abaixo, há campos de entrada para "Nome do aluno", "CPF", "Turma" (com o valor "1a-Avançado" selecionado), "Email", "Telefone" e "Senha". Cada campo possui um ícone de categoria à direita. No final da tela, há um botão azul "Cadastrar Aluno".

FONTE: AUTORIA PRÓPRIA, 2025

A tela na figura 123 exibida corresponde ao formulário de cadastro de alunos no sistema. Nela, a secretaria pode inserir os dados principais do estudante, incluindo nome completo, CPF, turma associada, *e-mail*, telefone e senha de acesso. A interface também disponibiliza a opção de alterar a foto de perfil do aluno, tornando o registro mais personalizado. Na parte inferior, encontra-se o botão "Cadastrar Aluno", que ao ser acionado confirma a inclusão do novo estudante no sistema. Essa tela foi desenvolvida para facilitar o processo de registro, garantindo que todas as

forem aprovadas, o código tenta executar uma transação no banco de dados. Ele lê o conteúdo binário da foto enviada e inicia a transação. Depois, prepara uma consulta SQL para inserir os dados do usuário na tabela usuário, incluindo o perfil, nome, *email*, CPF, telefone, senha e tipo, vinculando a foto ao registro. Após a execução, obtém o ID do novo usuário inserido. Depois, realiza uma inserção na tabela aluno, associando o novo usuário como aluno.

Depois, consulta a tabela turma para obter o ID da turma correspondente ao nome fornecido, validando sua existência no banco. Caso a turma não exista, lança uma exceção. Finalmente, insere o relacionamento entre aluno e turma na tabela aluno_turma, registrando também a data da matrícula automaticamente. Se tudo ocorrer sem erros, a transação é confirmada (*commit*) e o usuário é redirecionado para a página de visualização dos alunos, com um alerta de sucesso. Em caso de erro durante a transação, o código realiza um *rollback* para desfazer as alterações e exibe a mensagem do erro, garantindo a integridade dos dados.

Além disso, esse processo exemplifica boas práticas de desenvolvimento em sistemas de gestão acadêmica, pois combina validação de dados, uso de transações e tratamento de exceções de forma estruturada, assegurando que apenas informações corretas e completas sejam registradas no banco. A abordagem de transações garante que todas as inserções relacionadas ao aluno ocorram de maneira atômica, evitando inconsistências entre tabelas e assegurando a integridade referencial.

O tratamento cuidadoso de erros, aliado à verificação da existência da turma antes da associação, previne registros inválidos e facilita a manutenção futura do sistema. Essa implementação também valoriza a experiência do usuário, pois fornece alertas claros sobre possíveis problemas no preenchimento dos campos, orientando a correção imediata. Ao combinar segurança, confiabilidade e usabilidade, o código não apenas automatiza o processo de cadastro de alunos, mas também contribui para uma gestão acadêmica mais eficiente e profissional.

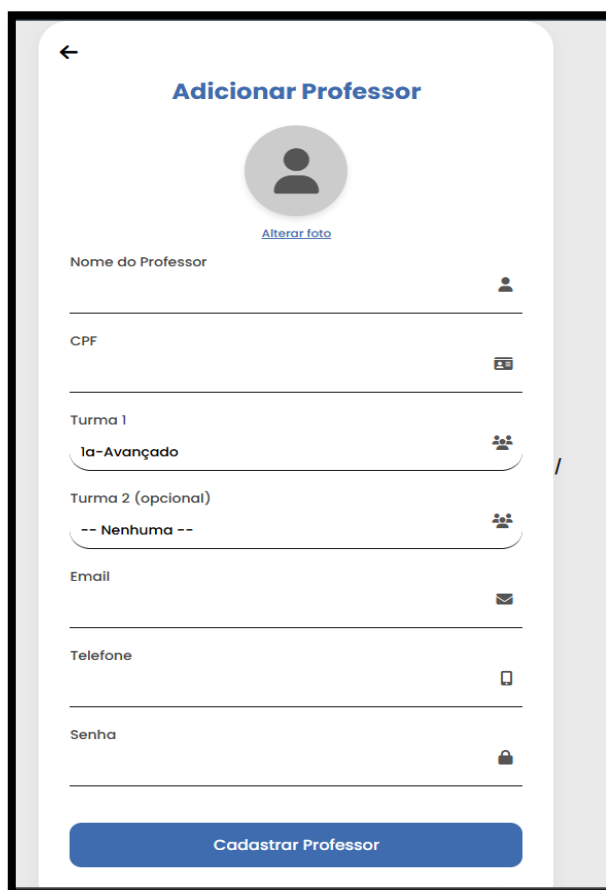
Adicionalmente, a estrutura adotada permite que o sistema seja facilmente expandido para suportar novos requisitos, como a inclusão de campos adicionais no cadastro, integração com outros módulos do sistema ou a implementação de regras de negócios mais complexas, sem a necessidade de reescrever a lógica principal. A

modularidade do código, aliada ao uso de *prepared statements* e transações, assegura que futuras modificações não comprometam a segurança ou a consistência dos dados.

Isso torna a solução escalável e sustentável a longo prazo, demonstrando como boas práticas de programação e planejamento arquitetural podem ser aplicadas para criar sistemas de gestão acadêmica robustos, confiáveis e adaptáveis às demandas crescentes das instituições de ensino

3.5.4.8 - Tela de Adicionar Professor

Figura 125 – Tela de adicionar professor da secretaria



A imagem mostra a tela de cadastro de um professor em um sistema. No topo, há um ícone de seta para voltar e o título "Adicionar Professor". Abaixo do título, há um ícone de perfil de usuário com o link "Alterar foto". O formulário contém os seguintes campos:

- Nome do Professor
- CPF
- Turma 1: 1a-Avançado
- Turma 2 (opcional): -- Nenhuma --
- Email
- Telefone
- Senha

Um botão azul "Cadastrar Professor" está localizado na base do formulário.

FONTE: AUTORIA PRÓPRIA, 2025

A tela exibida na figura 125 corresponde ao formulário de cadastro de professores no sistema. Nela, a secretaria pode inserir as principais informações do docente, incluindo nome completo, CPF, turmas associadas (sendo a primeira obrigatória e a segunda opcional), *e-mail*, telefone e senha de acesso. A interface

também oferece a possibilidade de alterar a foto de perfil do professor, proporcionando um registro mais personalizado.

Na parte inferior, encontra-se o botão “Cadastrar Professor”, que ao ser acionado confirma a inclusão do novo professor no sistema. Essa tela foi desenvolvida para simplificar o processo de registro, reunindo todas as informações necessárias em um único espaço de forma clara, objetiva e intuitiva.

Figura 126 – Tela de código adicionar professor da secretaria

```

$cpf = trim($_POST['cpf'] ?? '');

if (!$nome || !$turma || !$emailNovo || !$telefone || !$senhaUsuario || !$cpf) {
    echo "<script>alert('Preencha todos os campos obrigatórios.');

```

FONTE: AUTORIA PRÓPRIA, 2025

Este código exibido na figura 126 PHP é responsável pelo cadastro de um professor no sistema, incluindo a validação dos dados, inserção nas tabelas do banco de dados e o relacionamento com turmas. Inicialmente, o *script* obtém e limpa os dados enviados via *POST*, incluindo CPF, nome, turma, *email*, telefone e senha. Logo

em seguida, analisa se todos os campos obrigatórios estão preenchidos, verifica a validade do CPF e se uma foto válida foi enviada.

Caso alguma dessas validações falhe, uma mensagem de alerta é exibida. Se as validações forem aprovadas, o conteúdo binário da foto é obtido, e o *script* prepara a inserção dos dados do usuário na tabela usuário, incluindo foto, nome, *email*, CPF, telefone, senha e o tipo definido como "professor". Após executar a inserção, recupera o ID do novo usuário. Depois, verifica se o usuário já está cadastrado como professor na tabela professor. Se não estiver, insere o novo registro na tabela professor vinculando o ID do usuário. Caso já exista, apenas recupera o ID do professor.

Posteriormente, relaciona o professor com a primeira turma informada na tabela professor_turma. Se houver uma segunda turma válida, também a relaciona da mesma forma. Por fim, se tudo for executado com sucesso, o sistema exibe um alerta confirmando o cadastro do professor e redireciona para a página de visualização dos professores. Caso ocorra algum erro na inserção do usuário, é exibida uma mensagem informando o problema. O *script* encerra fechando a conexão com o banco de dados.

Essa implementação demonstra boas práticas de programação em sistemas de gestão acadêmica, pois combina validação de dados, tratamento de exceções e operações de banco de dados de forma estruturada, garantindo que apenas informações corretas e consistentes sejam registradas. O uso de verificações prévias, como a validação do CPF e a checagem de fotos válidas, assegura que o cadastro ocorra de forma confiável, evitando dados incorretos ou incompletos na base do sistema.

Além disso, a lógica adotada permite escalabilidade e manutenção facilitada, pois novas funcionalidades, como o vínculo com mais turmas, a inclusão de campos adicionais ou a integração com outros módulos do sistema, podem ser incorporadas sem a necessidade de reescrever a estrutura principal. A abordagem modular garante que alterações futuras não comprometam a segurança ou a consistência dos dados, mantendo a integridade do sistema mesmo com expansões.

Ao centralizar as operações críticas e fornecer *feedback* claro ao usuário, o sistema promove maior eficiência administrativa e reduz o risco de erros durante o cadastro. Essa organização contribui significativamente para uma gestão acadêmica

mais estruturada, permitindo que a secretaria execute suas atividades de forma ágil, confiável e alinhada às boas práticas de tecnologia aplicada à educação, além de facilitar a adaptação a novas demandas institucionais e regulamentações futuras.

3.5.4.9 - Tela de Adicionar Turma

Figura 127 – Tela de adicionar turma da secretaria

A imagem mostra a interface de usuário para 'Cadastrar Turma'. O formulário contém os seguintes campos:

- Nome da Turma (ícone de grupo de pessoas)
- Quantidade de Alunos (ícone de aluno)
- Quantidade de Atividades (ícone de lista)
- Quantidade de Aulas (ícone de sala de aula)
- Quantidade de Professores (ícone de professor)

Um botão azul 'Cadastrar Turma' está localizado na base do formulário.

FONTE: AUTORIA PRÓPRIA, 2025

A tela exibida na figura 127 corresponde ao formulário de cadastro de turmas no sistema. Nela, a secretaria pode inserir informações essenciais para a criação de uma nova turma, incluindo o nome da turma, a quantidade de alunos, a quantidade de atividades, a quantidade de aulas e o número de professores associados.

Na parte inferior, encontra-se o botão “Cadastrar Turma”, que ao ser acionado confirma a inclusão da nova turma na plataforma. Essa interface foi

projetada para organizar de maneira prática e clara os dados referentes às turmas, permitindo um registro rápido e eficiente dentro do sistema.

Figura 128 – Tela de código adicionar turma da secretaria

```

<?php
session_start();
if (!isset($_SESSION['email'])) {
    header("Location: ../../../../index.php");
    exit();
}

$host = "localhost";
$usuario = "u357936358_librinha";
$senha = "Librinh4tcc#";
$banco = "u357936358_scanhands";

$conn = new mysqli($host, $usuario, $senha, $banco);
if ($conn->connect_error) {
    die("Erro de conexão com o banco: " . $conn->connect_error);
}

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $nomeTurma = $_POST['nome'] ?? '';
    $qtdAluno = $_POST['qtd_aluno'] ?? '';
    $qtdAtividade = $_POST['qtd_atividade'] ?? '';
    $qtdAula = $_POST['qtd_aula'] ?? '';
    $qtdProf = $_POST['qtd_prof'] ?? '';

    if ($nomeTurma && is_numeric($qtdAluno) && is_numeric($qtdAtividade) && is_numeric($qtdAula) && is_numeric($qtdProf)) {
        $stmt = $conn->prepare("INSERT INTO turma (nome, qtd_aluno, qtd_atividade, qtd_aula, qtd_prof) VALUES (?, ?, ?, ?, ?)");
        $stmt->bind_param("siiii", $nomeTurma, $qtdAluno, $qtdAtividade, $qtdAula, $qtdProf);

        if ($stmt->execute()) {
            echo "<script>alert('Turma cadastrada com sucesso!'); window.location.href='visuTurma.php';</script>";
        } else {
            echo "<script>alert('Erro ao cadastrar turma: " . $stmt->error . "');</script>";
        }

        $stmt->close();
    } else {
        echo "<script>alert('Preencha todos os campos corretamente.');"</script>";
    }
}

$conn->close();
?>

<!DOCTYPE html>
<html lang="pt-br">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Cadastrar Turma</title>
    <link rel="stylesheet" href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;500;700&display=swap">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css">
    <link rel="icon" href="../../img/librinhaSemFundo.png" type="image/png">
    <style>
    * {
        box-sizing: border-box;
        font-family: 'Poppins', sans-serif;
        margin: 0;
        padding: 0;
    }
    body {
        background-color: #eaeaea;
    }

```

FONTE: AUTORIA PRÓPRIA, 2025

O código exibido na figura 128 inicia uma sessão PHP e verifica se a variável de sessão *email* está definida. Caso não esteja, o usuário é redirecionado para a página inicial (*index.php*) para evitar o acesso não autorizado. Em seguida, o código estabelece uma conexão com um banco de dados MySQL, utilizando os parâmetros de conexão como servidor (*localhost*), nome de usuário, senha e nome do banco de

dados. Se houver algum erro na conexão, o *script* interrompe a execução e exibe uma mensagem de erro.

Quando o método de requisição HTTP é *POST*, o código coleta os dados enviados pelo formulário: nome da turma, quantidade de alunos, quantidade de atividades, quantidade de aulas e quantidade de professores. Cada dado é obtido do *array* `$_POST`, utilizando o operador de coalescência nula para garantir que, se o campo não existir, um valor vazio será atribuído. Depois, o código verifica se todos os campos obrigatórios estão preenchidos corretamente, e se os campos numéricos realmente contêm valores numéricos. Se as validações forem bem-sucedidas, uma instrução SQL é preparada para inserir os dados da turma na tabela *turma* do banco de dados, e os parâmetros são vinculados com os valores recebidos, e a execução da *query* é realizada.

Se a inserção for realizada com sucesso, o usuário recebe uma mensagem de confirmação via *alert* JavaScript e é redirecionado para a página de visualização das turmas cadastradas. Caso haja algum erro durante a inserção, uma mensagem de erro é exibida na tela. Se os campos estiverem incorretos ou incompletos, uma mensagem alertando para o preenchimento correto dos campos é mostrada, e, finalmente, o código fecha a instrução preparada e a conexão com o banco de dados.

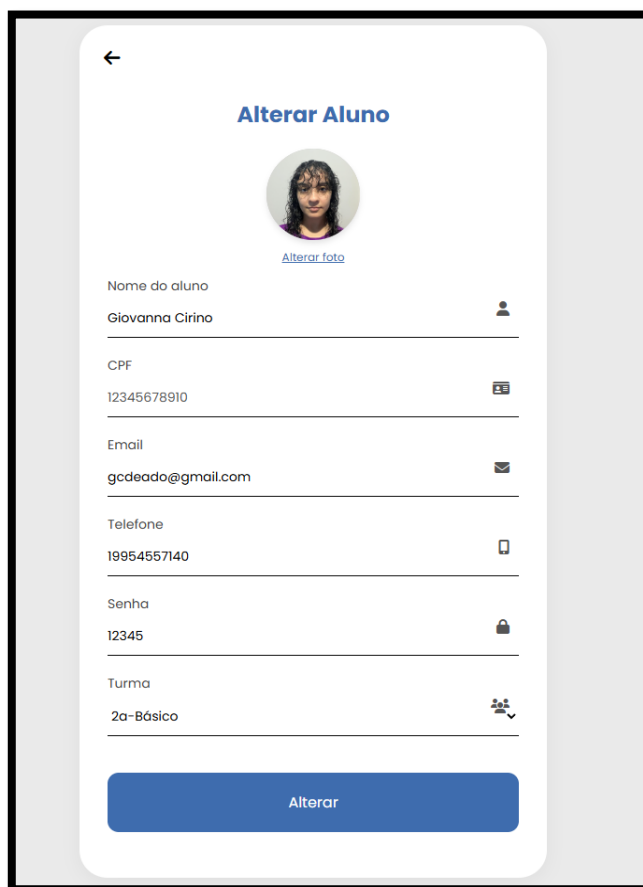
Além disso, essa implementação evidencia boas práticas de programação e gestão de dados, ao garantir validação completa dos campos antes da inserção, fornecer *feedback* imediato ao usuário e assegurar que apenas informações consistentes sejam armazenadas no banco. A combinação de validações do lado do servidor, mensagens de alerta claras e o uso de *prepared statements* aumenta a confiabilidade do sistema, reduzindo a possibilidade de erros ou dados incorretos. Essa abordagem também facilita a manutenção futura, permitindo ajustes na lógica de validação ou inclusão de novos atributos sem comprometer a integridade da aplicação.

Adicionalmente, o *design* da interface e a lógica do código promovem uma experiência de usuário mais eficiente, pois simplificam o processo de cadastro e minimizam retrabalho em caso de dados incompletos ou incorretos. A centralização da validação e da execução da *query* SQL garante consistência na inserção dos registros, enquanto o redirecionamento após a confirmação do cadastro mantém a

navegação fluida e intuitiva. Essa estrutura também permite escalabilidade do sistema, possibilitando futuras integrações com outros módulos acadêmicos, como gestão de professores e alunos, relatórios estatísticos ou ferramentas de acompanhamento de desempenho, sem comprometer a robustez ou segurança da aplicação.

3.5.4.10 - Tela de Alterar Aluno

Figura 129 – Tela alterar aluno da secretaria



←

Alterar Aluno

[Alterar foto](#)

Nome do aluno
Giovanna Cirino

CPF
12345678910

Email
gcdeado@gmail.com

Telefone
19954557140

Senha
12345

Turma
2a-Básico

Alterar

FONTE: AUTORIA PRÓPRIA, 2025

A tela exibida na figura 129 corresponde ao formulário de alteração de dados do aluno no sistema. Nela, a secretaria pode visualizar e atualizar as informações já cadastradas do estudante, incluindo nome completo, CPF, *e-mail*, telefone, senha de acesso e turma associada.

Além disso, a interface disponibiliza a opção de alterar a foto de perfil do aluno, garantindo que os registros permaneçam sempre atualizados e personalizados. Na parte inferior, encontra-se o botão “Alterar”, que ao ser acionado confirma a

atualização das informações no sistema. Essa tela foi desenvolvida para facilitar a edição de dados, mantendo o cadastro dos alunos organizado, confiável e sempre alinhado às necessidades da instituição.

Figura 130 – Tela de código alterar aluno da secretaria

```

#instanciar o objeto $stmtAluno;
$resultAluno = $stmtAluno->get_result();
$id_aluno = $resultAluno->fetch_assoc()['id_aluno'];
$stmtAluno->close();

// Buscar turma atual
$turmaAtual = null;
$stmtTurmaAtual = $conn->prepare("
    SELECT t.id_turma, t.nome
    FROM aluno_turma at
    JOIN turma t ON at.id_turma = t.id_turma
    WHERE at.id_aluno = ?
");
$stmtTurmaAtual->bind_param("i", $id_aluno);
$stmtTurmaAtual->execute();
$resultTurmaAtual = $stmtTurmaAtual->get_result();
if ($resultTurmaAtual->num_rows > 0) {
    $turmaAtual = $resultTurmaAtual->fetch_assoc();
}
$stmtTurmaAtual->close();

// Buscar todas as turmas
$turmas = [];
$resultTurmas = $conn->query("SELECT id_turma, nome FROM turma");
while ($row = $resultTurmas->fetch_assoc()) {
    $turmas[] = $row;
}

// Atualizar aluno
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $nome = $_POST['nome'];
    $email = $_POST['email'];
    $telefone = $_POST['telefone'];
    $senha = $_POST['senha'];
    $novaTurma = $_POST['turma'];

    if (!empty($_FILES['foto']['tmp_name'])) {
        $fotoData = file_get_contents($_FILES['foto']['tmp_name']);
        $updateSql = "UPDATE usuario SET nome=?, email=?, telefone=?, senha=?, foto_perfil=? WHERE cpf=?";
        $updateStmt = $conn->prepare($updateSql);
        $updateStmt->bind_param("sssss", $nome, $email, $telefone, $senha, $fotoData, $cpf);
    } else {
        $updateSql = "UPDATE usuario SET nome=?, email=?, telefone=?, senha=? WHERE cpf=?";
        $updateStmt = $conn->prepare($updateSql);
        $updateStmt->bind_param("sssss", $nome, $email, $telefone, $senha, $cpf);
    }

    $updateStmt->execute();
    $updateStmt->close();

    // Atualizar turma vinculada
    $conn->query("DELETE FROM aluno_turma WHERE id_aluno = $id_aluno");
    $stmtVinculo = $conn->prepare("INSERT INTO aluno_turma (id_aluno, id_turma) VALUES (?, ?)");
    $stmtVinculo->bind_param("ii", $id_aluno, $novaTurma);
    $stmtVinculo->execute();
    $stmtVinculo->close();

    header("Location: visuAlunos.php");
    exit();
}

```

FONTE: AUTORIA PRÓPRIA, 2025

O código exibido na figura 130 começa recuperando o identificador do aluno a partir do banco de dados, por meio de uma consulta preparada que utiliza o nome do usuário fornecido. Após executar a consulta, o código obtém o resultado e extrai o `id_aluno`. Em seguida, o código busca a turma atual à qual o aluno está vinculado. Para isso, ele executa uma consulta SQL que relaciona a tabela `aluno_turma` com a

tabela turma, utilizando o identificador do aluno para filtrar o registro correspondente. Se for encontrado algum resultado, ele armazena as informações da turma atual. O código também realiza uma consulta para buscar todas as turmas disponíveis, armazenando-as em um *array* para uso posterior.

Se o método da requisição HTTP for *POST*, indicando que um formulário foi submetido, o código coleta os dados atualizados do aluno, como nome, *email*, telefone, senha e a nova turma selecionada. Caso uma nova foto tenha sido enviada, ela é processada e convertida em dados binários para serem armazenados no banco. Com essas informações, o código prepara uma instrução SQL para atualizar os dados do usuário na tabela usuário. Caso uma nova foto tenha sido enviada, a atualização inclui o campo da foto; caso contrário, apenas os demais dados são atualizados.

Após executar a atualização, o código atualiza a associação entre o aluno e a turma no banco de dados. Primeiro, remove qualquer vínculo existente entre o aluno e suas turmas na tabela *aluno_turma*. Logo após, insere uma nova associação com a turma selecionada pelo usuário. Finalmente, o código redireciona o usuário para a página de visualização dos alunos (*visuAlunos.php*) e finaliza a execução do *script*.


Além dessas etapas principais, observa-se que o uso de consultas preparadas contribui para a segurança do sistema, reduzindo vulnerabilidades com a isenção de SQL e garantindo a integridade dos dados manipulados. O fluxo sequencial implementado facilita o entendimento e manutenção do código, deixando explícita a ordem das operações de leitura, atualização e vínculo de registros. Essa abordagem modular permite evoluir o sistema com mais facilidade, incorporando novas regras ou campos no formulário sem comprometer a lógica existente. O uso de variáveis claras e organizadas também favorece a rastreabilidade, ajudando desenvolvedores e administradores a identificarem rapidamente problemas ou pontos de melhoria. Por fim, essa estrutura robusta apoia tanto o gerenciamento eficiente dos dados dos alunos quanto a integração com outros módulos administrativos, fortalecendo o controle escolar e a experiência do usuário final.

3.5.4.11 - Tela de Alterar Professor

Figura 131 – Tela alterar professor da secretaria

←

Alterar Professor


[Alterar foto](#)

Nome do Professor
professor

CPF
45678978945

Email
professor@gmail.com

Telefone
19995865473

Senha
12345

Turma 1
2a-Básico

Turma 2 (opcional)
2a-Básico

Alterar

FONTE: AUTORIA PRÓPRIA, 2025

A tela exibida na figura 131 corresponde ao formulário de alteração de dados do professor no sistema. Nela, a secretaria pode visualizar e atualizar as informações já cadastradas do docente, incluindo nome completo, CPF, *e-mail*, telefone, senha de acesso e turmas associadas (com possibilidade de vincular até duas turmas). A interface também disponibiliza a opção de alterar a foto de perfil, permitindo manter o registro atualizado e personalizado. Na parte inferior, encontra-se o botão “Alterar”, que ao ser acionado confirma a atualização das informações no sistema. Essa tela foi projetada para facilitar a edição dos dados dos professores, garantindo que o cadastro permaneça sempre correto, organizado e em conformidade com as necessidades institucionais.

Além disso, o formulário de alteração proporciona maior segurança e confiabilidade na gestão acadêmica, pois permite que os dados sejam revisados e

corrigidos sempre que necessário, evitando inconsistências no banco de dados. O *layout* claro e funcional, aliado à validação dos campos, assegura que as informações atualizadas estejam completas e corretas, reduzindo o risco de erros e garantindo que a secretaria possa gerenciar os cadastros de forma eficiente. Outro ponto importante é que a interface permite maior flexibilidade e adaptabilidade, possibilitando a adição futura de novos campos ou funcionalidades sem comprometer a experiência do usuário.

A combinação de visualização, edição e atualização centralizada contribui para a manutenção da integridade dos registros, assegurando que a plataforma continue atendendo às demandas institucionais e promovendo uma gestão docente organizada, precisa e profissional. Dessa forma, o sistema fortalece o processo administrativo, otimiza a rotina da secretaria e garante maior transparência e confiabilidade nas informações, criando um ambiente digital mais seguro, moderno e preparado para evoluções futuras, com processos cada vez mais automatizados e integrados à realidade educacional.

Essa abordagem também melhora a experiência do usuário final, que passa a contar com um fluxo de trabalho mais intuitivo e rápido. Com a centralização das funcionalidades e a clareza visual, a probabilidade de falhas humanas diminui consideravelmente, contribuindo para a padronização das rotinas internas. Além disso, a implementação desse tipo de tela favorece a rastreabilidade e o histórico de alterações, permitindo que a instituição acompanhe as mudanças realizadas e mantenha um controle preciso sobre cada etapa do gerenciamento dos cadastros dos professores.

A interface clara e objetiva facilita o preenchimento correto dos dados, reduzindo inconsistências e retrabalho. A presença de campos obrigatórios e opcionais bem definidos orienta o usuário na tomada de decisão, promovendo maior eficiência no processo de cadastro. O botão de atualização visível e acessível reforça a ação final, garantindo que as alterações sejam salvas com segurança. A possibilidade de alteração da foto do professor também contribui para a personalização e identificação visual dos perfis. Com isso, o sistema se torna uma ferramenta estratégica para a gestão educacional, alinhando tecnologia à excelência administrativa.

Figura 132 – Tela de código alterar professor da secretaria

```

<?php
<!-- ... -->
$stmtTurmasVinc->execute();
$resultTurmasVinc = $stmtTurmasVinc->get_result();
while ($row = $resultTurmasVinc->fetch_assoc()) {
    $turmasVinculadas[] = $row['id_turma'];
}
$stmtTurmasVinc->close();

// Buscar todas as turmas
$turmas = [];
$resultTurmas = $conn->query("SELECT id_turma, nome FROM turma");
while ($row = $resultTurmas->fetch_assoc()) {
    $turmas[] = $row;
}

// Atualizar professor
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $nome = $_POST['nome'];
    $email = $_POST['email'];
    $telefone = $_POST['telefone'];
    $senha = $_POST['senha'];
    $turma1 = $_POST['turma1'] ?? '';
    $turma2 = $_POST['turma2'] ?? '';

    if (!empty($_FILES['foto']['tmp_name'])) {
        $fotoData = file_get_contents($_FILES['foto']['tmp_name']);
        $updateSql = "UPDATE usuario SET nome=?, email=?, telefone=?, senha=?, foto_perfil=? WHERE cpf=?";
        $updateStmt = $conn->prepare($updateSql);
        $updateStmt->bind_param("ssssss", $nome, $email, $telefone, $senha, $fotoData, $cpf);
    } else {
        $updateSql = "UPDATE usuario SET nome=?, email=?, telefone=?, senha=? WHERE cpf=?";
        $updateStmt = $conn->prepare($updateSql);
        $updateStmt->bind_param("sssss", $nome, $email, $telefone, $senha, $cpf);
    }

    $updateStmt->execute();
    $updateStmt->close();

    // Atualizar vínculos de turma
    $conn->query("DELETE FROM professor_turma WHERE id_professor = $id_professor");
    $stmtVinculo = $conn->prepare("INSERT INTO professor_turma (id_professor, id_turma) VALUES (?, ?)");

    if (!empty($turma1)) {
        $stmtVinculo->bind_param("ii", $id_professor, $turma1);
        $stmtVinculo->execute();
    }

    if (!empty($turma2) && $turma2 != $turma1) {
        $stmtVinculo->bind_param("ii", $id_professor, $turma2);
        $stmtVinculo->execute();
    }

    $stmtVinculo->close();

    header("Location: visuProf.php");
    exit();
}

$conn->close();

$fotoBase64 = !empty($professor['foto_perfil']) ? 'data:image/jpeg;base64,' . base64_encode($professor['foto_perfil']) : './img/iconPerfil.png';
?>

```

FONTE: AUTORIA PRÓPRIA, 2025

O código ilustrado na figura 132 inicia buscando as turmas vinculadas a um determinado professor a partir da tabela `professor_turma`. Ele executa uma consulta preparada para obter os identificadores das turmas associadas ao professor e armazena esses valores em um *array* chamado “`turmasVinculadas`”. Na sequência, o código busca todas as turmas disponíveis no sistema, executando uma consulta SQL simples para obter os identificadores e nomes das turmas, armazenando esses dados em um *array* chamado `turmas`.

Ao verificar se o método da requisição HTTP é *POST*, o código obtém os dados enviados por um formulário, como o nome, *email*, telefone, senha e as duas turmas selecionadas para o professor. Se um arquivo de foto foi enviado, o código

processa a imagem para armazenar os dados binários, que serão atualizados no banco de dados junto com as outras informações do professor.

O código então prepara uma instrução SQL para atualizar os dados do professor na tabela usuário. Caso uma foto tenha sido enviada, o campo da foto é incluído na atualização; caso contrário, somente os outros dados são atualizados. Após executar a atualização, o código atualiza os vínculos entre o professor e as turmas no banco de dados. Primeiro, ele remove todos os vínculos existentes do professor na tabela professor_turma. A seguir, ele insere novos vínculos conforme as turmas selecionadas no formulário. Se a primeira turma foi informada, o vínculo é criado; caso a segunda turma seja diferente da primeira e não esteja vazia, um segundo vínculo é criado.

Finalmente, o código redireciona o usuário para a página de visualização dos professores (visuProf.php) e encerra a execução do script. No final, o código gera uma variável contendo a foto do perfil do professor em formato *Base64*, que pode ser usada para exibição da imagem na interface. Caso o professor não possua uma foto cadastrada, é utilizada uma imagem padrão. Essa implementação evidencia boas práticas de desenvolvimento em sistemas de gestão acadêmica, pois garante a integridade dos dados ao atualizar tanto as informações pessoais do professor quanto os vínculos com as turmas de forma controlada.

O uso de consultas preparadas e a manipulação condicional da foto asseguram segurança e flexibilidade, permitindo que alterações futuras ou integrações com outros módulos sejam realizadas sem comprometer a consistência do banco de dados. Além disso, a geração da imagem em *Base64* facilita a exibição imediata da foto na interface, promovendo uma experiência mais intuitiva e visualmente organizada para o usuário.

Essa abordagem também demonstra preocupação com a escalabilidade e manutenção do sistema, visto que a estrutura modular e o tratamento sistemático dos dados tornam o código mais fácil de evoluir e integrar com novas funcionalidades ou outros sistemas institucionais no futuro, aumentando a confiabilidade geral da plataforma. Em conjunto, esses procedimentos criam um fluxo automatizado e seguro, garantindo que os dados sejam manipulados de maneira padronizada e aditável, fortalecendo a governança digital da instituição. Essa prática contribui diretamente

para um ambiente de gestão mais transparente, moderno e sustentável, alinhado às melhores recomendações técnicas para sistemas *web* e acadêmicos.

3.5.4.12 - Tela de Alterar Turma

Figura 133 – tela alterar turma da secretaria

A imagem mostra a interface de usuário para alterar uma turma. O formulário é dividido em seções horizontais, cada uma com um rótulo à esquerda, um campo de entrada no centro e um ícone de edição à direita. Os dados atuais são: Nome da Turma: 1a-Básico; Quantidade de Alunos: 1; Quantidade de Atividades: 0; Quantidade de Aulas: 0; Quantidade de Professores: 2. Um botão azul 'Salvar Alterações' está posicionado na base do formulário.

Nome da Turma	Quantidade de Alunos	Quantidade de Atividades	Quantidade de Aulas	Quantidade de Professores
1a-Básico	1	0	0	2

FONTE: AUTORIA PRÓPRIA, 2025

A tela exibida na figura 133 corresponde ao formulário de alteração de turmas no sistema. Nela, a secretaria pode revisar e atualizar as informações já cadastradas de uma turma, incluindo nome da turma, quantidade de alunos, quantidade de atividades, quantidade de aulas e número de professores vinculados. Na parte inferior, encontra-se o botão “Salvar Alterações”, que ao ser acionado confirma a atualização dos dados no sistema. Essa interface foi desenvolvida para manter as informações das turmas sempre corretas e atualizadas, assegurando praticidade, clareza e organização no gerenciamento acadêmico.

Além disso, o formulário facilita a manutenção da integridade e consistência das informações no banco de dados, permitindo que ajustes sejam feitos de maneira segura e eficiente. A interface intuitiva e o agrupamento lógico dos campos contribuem

para reduzir erros durante a edição, enquanto o *feedback* imediato ao salvar as alterações garante que o usuário tenha certeza de que os dados foram atualizados corretamente. Essa abordagem promove uma gestão acadêmica mais organizada, ágil e confiável.

Figura 134 – Tela de código alterar professor da secretaria

```

$stmtAluno->execute();
$resultAluno = $stmtAluno->get_result();
$id_aluno = $resultAluno->fetch_assoc()['id_aluno'];
$stmtAluno->close();

// Buscar turma atual
$turmaAtual = null;
$stmtTurmaAtual = $conn->prepare("
SELECT t.id_turma, t.nome
FROM aluno_turma at
JOIN turma t ON at.id_turma = t.id_turma
WHERE at.id_aluno = ?
");
$stmtTurmaAtual->bind_param("i", $id_aluno);
$stmtTurmaAtual->execute();
$resultTurmaAtual = $stmtTurmaAtual->get_result();
if ($resultTurmaAtual->num_rows > 0) {
    $turmaAtual = $resultTurmaAtual->fetch_assoc();
}
$stmtTurmaAtual->close();

// Buscar todas as turmas
$turmas = [];
$resultTurmas = $conn->query("SELECT id_turma, nome FROM turma");
while ($row = $resultTurmas->fetch_assoc()) {
    $turmas[] = $row;
}

// Atualizar aluno
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $nome = $_POST['nome'];
    $email = $_POST['email'];
    $telefone = $_POST['telefone'];
    $senha = $_POST['senha'];
    $novaTurma = $_POST['turma'];

    if (!empty($_FILES['foto']['tmp_name'])) {
        $fotoData = file_get_contents($_FILES['foto']['tmp_name']);
        $updateSql = "UPDATE usuario SET nome=?, email=?, telefone=?, senha=?, foto_perfil=? WHERE cpf=?";
        $updateStmt = $conn->prepare($updateSql);
        $updateStmt->bind_param("ssssss", $nome, $email, $telefone, $senha, $fotoData, $cpf);
    } else {
        $updateSql = "UPDATE usuario SET nome=?, email=?, telefone=?, senha=? WHERE cpf=?";
        $updateStmt = $conn->prepare($updateSql);
        $updateStmt->bind_param("sssss", $nome, $email, $telefone, $senha, $cpf);
    }

    $updateStmt->execute();
    $updateStmt->close();

    // Atualizar turma vinculada
    $conn->query("DELETE FROM aluno_turma WHERE id_aluno = $id_aluno");
    $stmtVinculo = $conn->prepare("INSERT INTO aluno_turma (id_aluno, id_turma) VALUES (?, ?)");
    $stmtVinculo->bind_param("ii", $id_aluno, $novaTurma);
    $stmtVinculo->execute();
    $stmtVinculo->close();

    header("Location: visuAlunos.php");
    exit();
}

```

FONTE: AUTORIA PRÓPRIA, 2025

O código exibido na figura 134 inicia buscando o identificador do aluno a partir do banco de dados, com base em uma consulta SQL preparada que usa o CPF do aluno para localizar seu ID único. Em seguida, ele busca a turma atual do aluno. Para isso, executa uma consulta preparada que faz um *JOIN* entre as tabelas *aluno_turma* e *turma*, para obter o ID e o nome da turma na qual o aluno está atualmente

matriculado. Se existir uma turma vinculada ao aluno, esses dados são armazenados na variável \$turmaAtual.

Logo após o código busca todas as turmas disponíveis no sistema, executando uma consulta SQL simples para obter os IDs e nomes das turmas, armazenando os resultados em um *array* \$turmas. Quando a requisição HTTP é do tipo *POST*, significa que um formulário foi enviado para atualizar as informações do aluno. O código então coleta os dados enviados no formulário, como nome, *email*, telefone, senha e a nova turma para a qual o aluno será vinculado. Se um arquivo de foto foi enviado, o código lê o conteúdo do arquivo para armazená-lo no banco de dados.

Em seguida, ele prepara uma instrução SQL para atualizar os dados do usuário, incluindo o campo da foto de perfil. Caso não haja uma nova foto, ele prepara a instrução sem atualizar a foto. Depois de preparar a instrução correta, o código executa a atualização dos dados do aluno no banco e fecha o *statement*. Subsequentemente, ele remove todos os vínculos antigos do aluno com turmas na tabela *aluno_turma* e insere um novo vínculo para a turma selecionada no formulário. Por fim, o código leva o usuário para a página de visualização dos alunos (*visuAlunos.php*) e encerra a execução do *script*.

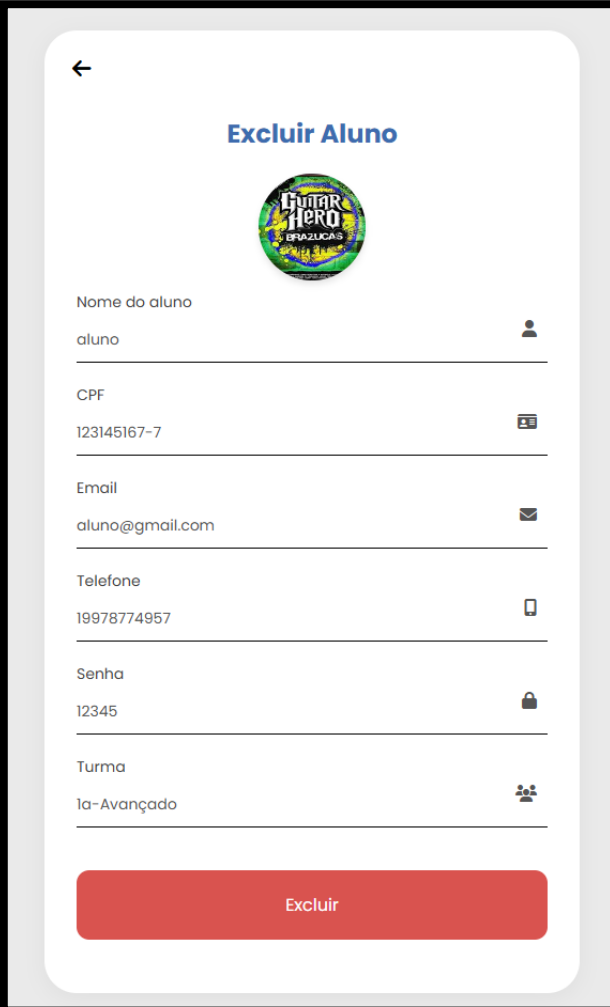
Essa implementação demonstra boas práticas de desenvolvimento em sistemas de gestão acadêmica, ao combinar validação, atualização segura de dados e manutenção de relacionamentos entre tabelas de forma estruturada. O uso de consultas preparadas e a lógica condicional para atualização da foto de perfil garantem segurança contra inserção de dados incorretos ou ataques de SQL injection, protegendo informações sensíveis de alunos. Além disso, a abordagem de remover os vínculos antigos antes de inserir o novo vínculo assegura que os registros permaneçam consistentes, evitando duplicidades ou inconsistências no banco de dados.

A organização do código também contribui para a clareza e manutenção futura, permitindo que novas funcionalidades, como a associação de múltiplas turmas, integração com módulos de notas ou acompanhamento de desempenho acadêmico, sejam implementadas sem comprometer a estrutura existente. Ao centralizar todas as operações críticas em uma única rotina, o sistema proporciona maior confiabilidade e

eficiência, permitindo que a secretaria realize alterações de forma segura, rápida e organizada, garantindo a integridade e a precisão das informações. Além disso, a geração de variáveis adicionais, como a foto em *Base64*, melhora a experiência do usuário na interface, permitindo que as alterações sejam refletidas imediatamente na plataforma e mantendo um padrão visual consistente. Essa abordagem integrada reforça a robustez do sistema, promovendo não apenas a manutenção correta dos dados, mas também a escalabilidade e a adaptabilidade a futuras demandas institucionais.

3.5.4.13 - Tela de Excluir Aluno

Figura 135 – Tela de excluir aluno da secretaria



A tela de exclusão de aluno apresenta o seguinte layout:

- Um ícone de seta para trás no canto superior esquerdo.
- O título "Excluir Aluno" em azul no topo central.
- Um logotipo circular "Guitar Hero BRAZUCAS" no centro.
- Formulário com os seguintes campos e ícones de validação:
 - Nome do aluno: "aluno" com ícone de pessoa.
 - CPF: "123145167-7" com ícone de documento.
 - Email: "aluno@gmail.com" com ícone de envelope.
 - Telefone: "19978774957" com ícone de telefone.
 - Senha: "12345" com ícone de cadeado.
 - Turma: "1a-Avançado" com ícone de grupo de pessoas.
- Um botão redondo "Excluir" em vermelho na base.

FONTE: AUTORIA PRÓPRIA, 2025

A tela exibida na figura 135 corresponde ao formulário de exclusão de aluno no sistema. Nela, a secretaria pode visualizar as informações já cadastradas do

estudante, incluindo nome completo, CPF, e-mail, telefone, senha de acesso e turma vinculada, além da foto de perfil associada ao registro. Na parte inferior, encontra-se o botão “Excluir”, destacado em vermelho, que ao ser acionado remove o aluno definitivamente da base de dados. Essa tela foi projetada para garantir clareza e segurança no processo de exclusão, permitindo a conferência das informações antes da remoção do registro.

Figura 136 – Tela do código da tela excluir aluno da secretaria

```

}
$stmtTurma->close();
// Excluir aluno e suas relações
if ($SERVER["REQUEST_METHOD"] === "POST") {
    $id_usuario = $aluno['id_usuario'];

    // Buscar id_aluno
    $stmtAluno = $conn->prepare("SELECT id_aluno FROM aluno WHERE id_usuario = ?");
    $stmtAluno->bind_param("i", $id_usuario);
    $stmtAluno->execute();
    $resultAluno = $stmtAluno->get_result();

    if ($resultAluno->num_rows > 0) {
        $id_aluno = $resultAluno->fetch_assoc()['id_aluno'];

        // Excluir relações com turma
        $stmtTurmaDel = $conn->prepare("DELETE FROM aluno_turma WHERE id_aluno = ?");
        $stmtTurmaDel->bind_param("i", $id_aluno);
        $stmtTurmaDel->execute();
        $stmtTurmaDel->close();

        // Excluir da tabela aluno
        $stmtDelAluno = $conn->prepare("DELETE FROM aluno WHERE id_aluno = ?");
        $stmtDelAluno->bind_param("i", $id_aluno);
        $stmtDelAluno->execute();
        $stmtDelAluno->close();
    }

    $stmtAluno->close();

    // Excluir da tabela usuario
    $deleteSql = "DELETE FROM usuario WHERE cpf=?";
    $deleteStmt = $conn->prepare($deleteSql);
    $deleteStmt->bind_param("s", $cpf);

    if ($deleteStmt->execute()) {
        header("Location: visuAlunos.php");
        exit();
    } else {
        echo "Erro ao excluir: " . $conn->error;
    }

    $deleteStmt->close();
}

$conn->close();

// Foto de perfil
$fotoBase64 = 'empty($aluno["foto_perfil"]) ? 'data:image/jpeg;base64,' . base64_encode($aluno['foto_perfil']) : './img/iconPerfil.png';
?>
<!DOCTYPE html>
<html lang="pt-br">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Excluir Aluno</title>
<link rel="icon" href="./img/librinhaSemFundo.png" type="image/png">
<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;700&display=swap" rel="stylesheet">

```

FONTE: AUTORIA PRÓPRIA, 2025

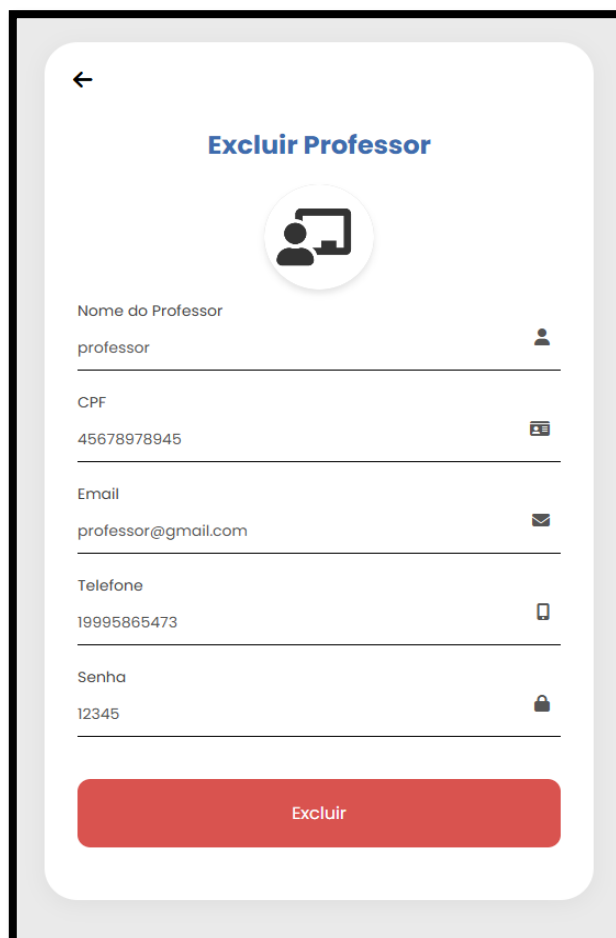
O código exibido na figura 136 é responsável por excluir um aluno do sistema, incluindo suas relações com turmas e sua conta de usuário. Primeiramente, verifica se a requisição HTTP é do tipo *POST*, permitindo que a ação de exclusão seja realizada por meio de um formulário ou requisição adequada. Após isso, recupera o identificador do usuário associado ao aluno a ser excluído. Depois, o código busca o

ID do aluno na tabela aluno usando o ID do usuário. Caso o aluno exista, o *script* prossegue para excluir as relações do aluno com as turmas, removendo todas as entradas correspondentes na tabela aluno_turma. Em seguida, exclui o próprio registro do aluno na tabela aluno.

Por fim, o código exclui o registro do usuário na tabela usuário utilizando o CPF do aluno. Se a exclusão for bem-sucedida, o sistema redireciona para a página que lista os alunos (visuAlunos.php). Se ocorrer algum erro durante o processo, uma mensagem de erro é exibida. Além disso, o código prepara a exibição da foto de perfil do aluno. Se o aluno possuir uma foto cadastrada, ela é convertida para o formato *base64* para ser exibida diretamente na página; caso contrário, uma imagem padrão é mostrada.


3.5.4.14 - Tela de Excluir Professor


Figura 137 – Tela de excluir professor da secretaria





←


Excluir Professor




Nome do Professor
professor 

CPF
45678978945 

Email
professor@gmail.com 

Telefone
19995865473 

Senha
12345 

Excluir

FONTE: AUTORIA PRÓPRIA, 2025

A tela exibida na figura 137 corresponde ao formulário de exclusão de professor no sistema. Nela, a secretaria pode visualizar os dados já cadastrados do docente, como nome completo, CPF, *e-mail*, telefone e senha de acesso, além da foto de perfil associada ao registro. Na parte inferior, encontra-se o botão “Excluir”, destacado em vermelho, que ao ser ativado remove o professor da base de dados do sistema. Essa tela foi elaborada para oferecer clareza e segurança no processo de exclusão, possibilitando a conferência das informações antes da remoção definitiva do registro.

Figura 138 – Tela do código de excluir professor da secretaria

```

<?php
session_start();
if (!isset($_SESSION['email'])) {
    header("Location: ../../index.php");
    exit();
}

$emailLogado = $_SESSION['email'];

// Conexão com o banco
$servername = "localhost";
$username = "u357936358_librinha";
$password = "Librinh4tcc#";
$dbname = "u357936358_scannhands";
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Erro na conexão: " . $conn->connect_error);
}

// Pega o CPF da URL
if (!isset($_GET['cpf'])) {
    die("CPF não informado.");
}
$cpf = $_GET['cpf'];

// Busca os dados do aluno (incluindo foto BLOB)
$sql = "SELECT * FROM usuario WHERE cpf = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("s", $cpf);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows == 0) {
    die("Professor não encontrado.");
}

$aluno = $result->fetch_assoc();
$stmt->close();

// Excluir aluno
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $deleteSql = "DELETE FROM usuario WHERE cpf=?";
    $deleteStmt = $conn->prepare($deleteSql);
    $deleteStmt->bind_param("s", $cpf);

    if ($deleteStmt->execute()) {
        header("Location: visuProf.php");
        exit();
    } else {
        echo "Erro ao excluir: " . $conn->error;
    }
    $deleteStmt->close();
}

$conn->close();

// Converte a foto atual para Base64 (se existir)
$fotoBase64 = !empty($aluno['foto_perfil']) ? 'data:image/jpeg;base64, ' . base64_encode($aluno['foto_perfil']) : './img/iconPerfil.png';
?>
<!DOCTYPE html>
<html lang="pt-br">

```

FONTE: AUTORIA PRÓPRIA, 2025

O código exibido na figura 138 inicia uma sessão e verifica se o usuário está autenticado, conferindo se a variável de sessão *email* está definida. Caso não esteja, o usuário é redirecionado para a página de *login*. Em seguida, o *script* estabelece uma conexão com o banco de dados MySQL, utilizando as credenciais e informações definidas para acessar a base de dados. O código então verifica se o CPF do professor foi fornecido via URL (*Uniform Resource Locator* – Localizador Uniforme de Recursos) (através do método *GET*). Se o CPF não estiver presente, o sistema interrompe a execução e exibe uma mensagem informando que o CPF não foi informado.

Com o CPF em mãos, o código prepara e executa uma consulta SQL para buscar os dados do professor na tabela usuário, incluindo a foto de perfil armazenada como *BLOB*. Se nenhum registro for encontrado, o *script* para a execução e exibe a mensagem "Professor não encontrado." Caso o professor seja encontrado, os dados são carregados para serem exibidos ou manipulados posteriormente.

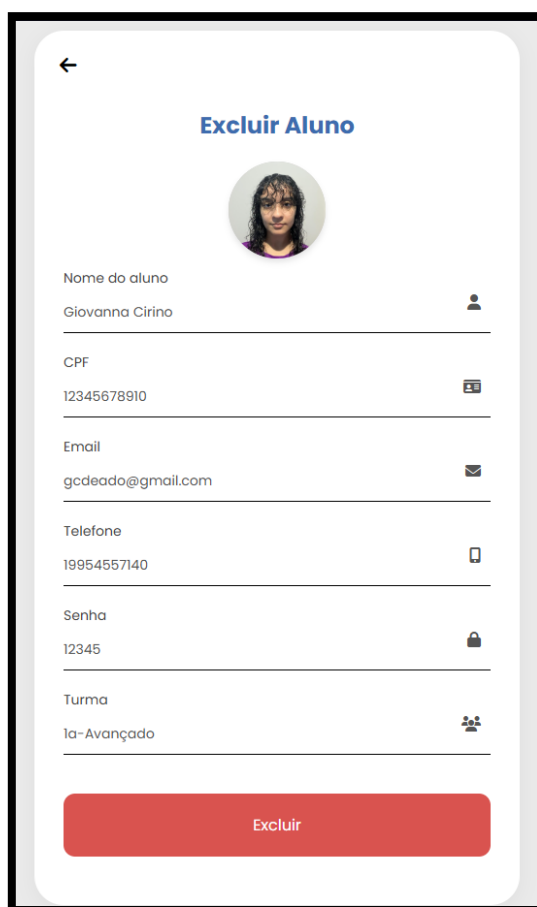
Quando o método HTTP da requisição for *POST*, o código executa a exclusão do professor na tabela usuário baseado no CPF informado. Se a exclusão for bem-sucedida, o usuário é levado para a página de visualização dos professores (*visuProf.php*). Se ocorrer algum erro durante a exclusão, uma mensagem de erro é exibida. Por fim, o código converte a foto de perfil do professor para o formato *Base64* para facilitar a exibição na página HTML.

Caso o professor não tenha uma foto cadastrada, uma imagem padrão é usada. Além disso, o uso de *prepared statements* com *bind_param* melhora a segurança do sistema, prevenindo ataques de SQL Injection. A separação entre os métodos *GET* e *POST* permite que a mesma página trate tanto da visualização quanto da exclusão, otimizando o fluxo. O uso de condicionais claras e mensagens de erro facilita a manutenção e depuração do sistema por parte dos desenvolvedores. A conexão com o banco é encerrada adequadamente após as operações, evitando consumo desnecessário de recursos.

Por fim, a utilização de codificação *Base64* para a imagem permite que ela seja incorporada diretamente no HTML, sem a necessidade de armazenar a foto em diretórios públicos.

3.5.4.15 – Tela de Excluir Turma

Figura 139 - Tela de excluir aluno



FONTE: AUTORIA PRÓPRIA, 2025

A tela “Excluir Aluno”, representada na imagem 139, autoriza o usuário com perfil administrativo a remover um estudante do sistema de forma segura e controlada. Nela, são exibidas informações importantes do aluno, como nome completo, turma, CPF, telefone e *e-mail*, todas apresentadas de maneira clara e organizada para que o usuário possa confirmar os dados antes de realizar a exclusão. No topo da tela, a foto de perfil do aluno reforça a identificação visual, reduzindo o risco de exclusões equivocadas.

Cada campo possui um ícone representativo ao lado, facilitando a compreensão rápida dos dados exibidos. A interface é minimalista, com cores suaves e elementos bem espaçados, proporcionando uma navegação intuitiva e agradável. O botão vermelho destacado com o texto “Excluir” está localizado na parte inferior da tela, evidenciando a ação principal, e seu tamanho favorece o toque em dispositivos móveis.

Além disso, o *layout* inclui um botão de retorno no canto superior esquerdo, permitindo que o usuário volte para a tela anterior sem confirmar a exclusão, o que ajuda a evitar ações acidentais. Essa tela foi projetada para garantir que o processo de remoção do aluno seja claro, seguro e que o usuário tenha todas as informações necessárias para tomar uma decisão consciente.

Figura 140 – Tela do código de excluir turma da secretaria

```
<?php
session_start();
if (!isset($_SESSION['email'])) {
    header("Location: ../../index.php");
    exit();
}

$emailLogado = $_SESSION['email'];

// Conexão com o banco
$servername = "localhost";
$username = "u357936358_librinha";
$password = "Librinh4tcc#";
$dbname = "u357936358_scanhands";
$conn = new mysqli($servername, $username, $password, $dbname);
if ($conn->connect_error) {
    die("Erro na conexão: " . $conn->connect_error);
}

// Pega o nome da turma da URL
if (!isset($_GET['nome'])) {
    die("Nome da turma não informado.");
}
$nomeTurma = $_GET['nome'];

// Busca os dados da turma
$sql = "SELECT * FROM turma WHERE nome = ?";
$stmt = $conn->prepare($sql);
$stmt->bind_param("s", $nomeTurma);
$stmt->execute();
$result = $stmt->get_result();

if ($result->num_rows == 0) {
    die("Turma não encontrada.");
}

$turma = $result->fetch_assoc();
$stmt->close();

// Excluir turma
if ($_SERVER["REQUEST_METHOD"] === "POST") {
    $deleteSql = "DELETE FROM turma WHERE nome=?";
    $deleteStmt = $conn->prepare($deleteSql);
    $deleteStmt->bind_param("s", $nomeTurma);

    if ($deleteStmt->execute()) {
        header("Location: visuTurma.php");
        exit();
    } else {
        echo "Erro ao excluir: " . $conn->error;
    }
    $deleteStmt->close();
}

$conn->close();
?>
</DOCTYPE html>
```

FONTE: AUTORIA PRÓPRIA, 2025

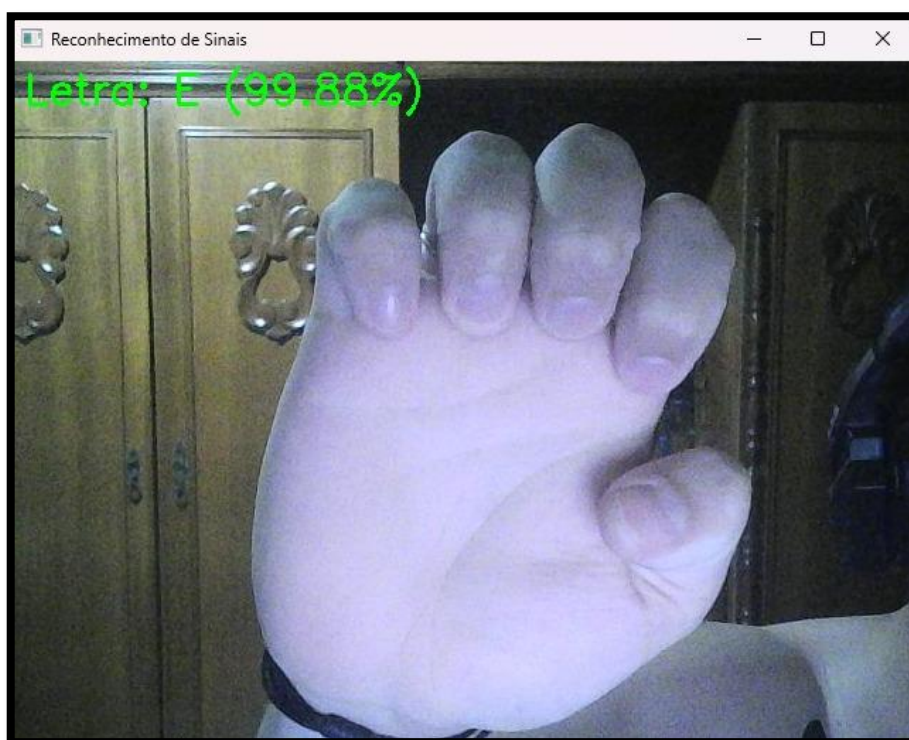
O código representado na imagem 140 inicia uma sessão e verifica se o usuário está autenticado, verificando se a variável de sessão *email* está definida. Caso

contrário, o usuário é redirecionado para a página inicial de *login*. Em seguida, o *script* estabelece uma conexão com o banco de dados MySQL, utilizando os dados do servidor, nome de usuário, senha e nome do banco. Depois, o código verifica se o parâmetro nome foi passado pela URL via método *GET*. Se não estiver presente, o *script* interrompe a execução exibindo uma mensagem informando que o nome da turma não foi informado. Com o nome da turma obtido, o código realiza uma consulta no banco para buscar todos os dados da turma correspondente ao nome informado. Se nenhuma turma for encontrada, o sistema exibe a mensagem "Turma não encontrada." Caso a turma seja localizada, os dados da turma são carregados para serem utilizados posteriormente.

O código então verifica se o método da requisição é *POST*, indicando que uma ação de exclusão foi solicitada. Neste caso, é preparado e executado um comando SQL para deletar a turma com o nome especificado. Se a exclusão for realizada com sucesso, o usuário é redirecionado para a página de visualização das turmas (*visuTurma.php*). Caso contrário, uma mensagem de erro é exibida informando o problema ocorrido. Por fim, a conexão com o banco de dados é encerrada.

3.5.5 – Tela do funcionamento da Câmera

Figura 141 – Tela da câmera em funcionamento



A Figura 141 mostra a interface do sistema SCAN HANDS, que utiliza visão computacional para o reconhecimento automático de sinais manuais. O sistema captura a imagem da mão do usuário em tempo real, processando a posição e forma dos dedos para identificar a letra correspondente. Essa tecnologia combina algoritmos de processamento de imagem, análise geométrica e aprendizado de máquina, permitindo que o sistema interprete gestos com alta precisão e rapidez.

Na parte superior da janela, a mensagem "Letra: E (99.88%)" indica que o algoritmo identificou o gesto como sendo a letra "E" com uma probabilidade de acerto de 99,88%. Essa taxa representa o nível de confiança do modelo, que varia de acordo com a semelhança entre o gesto capturado e o padrão armazenado no banco de dados do sistema. Quanto mais preciso for o gesto, maior será o índice de confiança. Valores próximos a 100% demonstram que a forma e a posição da mão estão praticamente idênticas ao padrão ideal utilizado no treinamento do modelo.

Dessa forma, o sistema demonstra sua capacidade de detectar, interpretar e responder instantaneamente aos gestos realizados, tornando o processo mais natural e interativo. A disposição central da mão na tela também contribui para a melhoria da leitura, garantindo que a área de interesse esteja devidamente alinhada com o campo de visão da câmera.

A elevada taxa de acerto de 99,88% demonstra a eficiência e precisão do modelo, que é capaz de identificar sinais manuais com alta confiabilidade, mesmo em condições normais de iluminação e ambiente. Isso evidencia a robustez do sistema, que contribui para a inclusão social e comunicação entre pessoas surdas e ouvintes por meio do reconhecimento de gestos na Língua Brasileira de Sinais (Libras). Além disso, o SCAN HANDS pode ser utilizado em ambientes educacionais, auxiliando no ensino de Libras, e em aplicações de acessibilidade digital, integrando-se a plataformas de tradução automática e interfaces inteligentes.

Em suma, o SCAN HANDS é uma ferramenta inovadora, acessível e confiável para o reconhecimento de sinais manuais, oferecendo precisão, rapidez e facilidade de uso. Seu desenvolvimento representa um importante avanço no campo da inteligência artificial aplicada à comunicação humana, reforçando o papel da tecnologia como instrumento de inclusão, aprendizado e interação social.

Figura 142 – Tela do código do funcionamento da câmera (parte 1)

```

1 import cv2
2 import numpy as np
3 import tensorflow as tf
4
5
6 interpreter = tf.lite.Interpreter(model_path="model.tflite")
7 interpreter.allocate_tensors^(
8     (função) get_input_details: Any
9 input_details = interpreter.get_input_details()
10 output_details = interpreter.get_output_details()
11
12
13 cap = cv2.VideoCapture(0)
14
15
16 if not cap.isOpened():
17     print("Erro ao abrir a câmera!")
18     exit()
19
20
21 labels = [
22     "A", "B", "C", "D", "E", "F", "G", "H", "I", "J",
23     "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T",
24     "U", "V", "W", "X", "Y"
25 ]
26
27 while True:
28
29     ret, frame = cap.read()
30     if not ret:
31         print("Falha ao capturar o frame!")
32         break
33
34     image = cv2.resize(frame, (224, 224))
35     image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
36     image_normalized = np.array(image_rgb, dtype=np.float32) / 255.0
37     input_data = np.expand_dims(image_normalized, axis=0)
38
39     interpreter.set_tensor(input_details[0]['index'], input_data)
40     interpreter.invoke()
41
42
43     output_data = interpreter.get_tensor(output_details[0]['index'])[0]
44
45
46     prediction = np.argmax(output_data)
47
48

```

FONTE: AUTORIA PRÓPRIA, 2025

Na figura 142, a porção de código apresentada realiza o reconhecimento de letras por meio de gestos capturados pela câmera do computador. O *script* inicia com a importação das bibliotecas *cv2*, *numpy* e *tensorflow*, que permitem, respectivamente, a captura e exibição de imagens, o processamento e normalização dos dados, e a execução do modelo de aprendizado de máquina. Em seguida, o modelo treinado é carregado por meio do comando `interpreter = tf.lite.Interpreter(model_path="model.tflite")`, e `interpreter.allocate_tensors()` prepara os tensores necessários para a execução das inferências.

Logo após, a câmera é ativada com `cap = cv2.VideoCapture(0)` e verificada por meio da condição `if not cap.isOpened():`, garantindo que o dispositivo foi inicializado corretamente. O código define, então, a lista *labels*, contendo as letras de A a Y, que representam as classes que o modelo pode reconhecer.

Durante a execução, o programa entra em um *loop while True*, onde cada *frame* capturado pela câmera é processado. A captura é realizada com *ret, frame = cap.read()*, e caso falhe, uma mensagem de erro é exibida e o *loop* é interrompido. O *frame* é redimensionado para 224x224 *pixels* com *cv2.resize(frame, (224, 224))*, convertido de BGR para RGB com *cv2.cvtColor(image, cv2.COLOR_BGR2RGB)* e normalizado com *np.array(image_rgb, dtype=np.float32) / 255.0*.

Em seguida, é adicionada uma dimensão extra com *np.expand_dims(image_normalized, axis=0)* para formar um lote compatível com o modelo. A inferência do modelo é realizada com *interpreter.set_tensor(input_details[0]['index'], input_data)* e *interpreter.invoke()*, retornando um vetor de probabilidades com *output_data = interpreter.get_tensor(output_details[0]['index'])[0]*.

A classe com maior probabilidade é identificada pelo comando *prediction = np.argmax(output_data)*, e a confiança da previsão é calculada com *confidence = output_data[prediction] * 100*. O valor do índice é associado à letra correspondente na lista *labels*, e caso esteja fora do intervalo, o código retorna um "?".

O resultado é exibido sobre o *frame* capturado com *cv2.putText(frame, f"Letra: {letter} ({confidence:.2f}%)", (10, 40), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)* e apresentado na janela "Reconhecimento de Sinais" por meio de *cv2.imshow()*. O *loop* pode ser encerrado pressionando a tecla ESC ou fechando a janela, condição verificada por *cv2.waitKey(1) == 27* e *cv2.getWindowProperty("Reconhecimento de Sinais", cv2.WND_PROP_VISIBLE) < 1*. Ao finalizar, a câmera é liberada com *cap.release()* e todas as janelas são fechadas com *cv2.destroyAllWindows()*.

Embora o código funcione corretamente para o reconhecimento em tempo real, é importante observar que a precisão do modelo depende da qualidade da captura da câmera e do treinamento prévio das imagens.

Recomenda-se a utilização de ambientes com boa iluminação e a inclusão de um número adequado de imagens para cada letra no treinamento, garantindo maior acurácia e confiabilidade nos resultados apresentados.

Figura 143 – Tela do código do funcionamento da câmera (parte 2)

```

✓ if prediction < len(labels):
    letter = labels[prediction]
✓ else:
    letter = "?"

✓ cv2.putText(frame, f"Letra: {letter} ({confidence:.2f}%)",
              (10, 30), cv2.FONT_HERSHEY_SIMPLEX,
              1, (0, 255, 0), 2)

cv2.imshow("Reconhecimento de Sinais", frame)

✓ if cv2.waitKey(1) == 27: # Tecla ESC
    break

✓ if cv2.getWindowProperty("Reconhecimento de Sinais", cv2.WND_PROP_VISIBLE) < 1:
    break

cap.release()
cv2.destroyAllWindows()

```

FONTE: AUTORIA PRÓPRIA, 2025

A Figura 143 apresenta um trecho de código em Python, desenvolvido com a biblioteca OpenCV (cv2), utilizado no sistema de reconhecimento de sinais manuais SCAN HANDS. O objetivo principal desse código é exibir em tempo real a letra identificada e o nível de confiança do modelo, com base na imagem capturada pela câmera.

O trecho inicia com uma verificação condicional que garante a validade da previsão feita pelo modelo. Caso o valor de *prediction* esteja dentro do intervalo da lista *labels*, a variável *letter* recebe a letra correspondente; caso contrário, é atribuído o caractere "?", indicando que o gesto não foi reconhecido. Essa etapa evita erros e assegura que apenas resultados válidos sejam exibidos.

A seguir, a função *cv2.putText()* é responsável por exibir na tela a letra reconhecida e a porcentagem de confiança. O texto é formatado para mostrar, por exemplo, "Letra: E (99.88%)", e é desenhado sobre a imagem com fonte simples, cor verde e espessura definida, proporcionando uma visualização clara e intuitiva. Essa informação é sobreposta diretamente ao vídeo, permitindo o acompanhamento imediato do reconhecimento.

Logo depois, o comando *cv2.imshow()* exibe o vídeo processado em uma janela chamada "Reconhecimento de Sinais", onde a imagem da câmera é atualizada

continuamente. O sistema permanece ativo até que o usuário pressione a tecla ESC (detectada por `cv2.waitKey(1) == 27`) ou feche a janela manualmente, o que é verificado por meio de `cv2.getWindowProperty()`. Em ambos os casos, o laço é interrompido de forma segura.

Por fim, os comandos `cap.release()` e `cv2.destroyAllWindows()` encerram a captura de vídeo e fecham todas as janelas abertas, liberando os recursos utilizados pelo sistema.

Figura 144 – Tela do código do funcionamento da câmera (parte 3)

```

)
    znpzef=,Λ9J1q9f1ou,
    c1922~woqe=,c9f680w1c9J,
    p9fcm~z1ze=35,
    f9w8ef~z1ze=(554, 554),
    q9f92ef~b9fμ,
Λ9J~q9f9 = q9f98eu~t1om~tlow~q1wecfouλ(
# Λ9J1q9c9o

)
    znpzef=,fΛ9J1u1u8,
    c1922~woqe=,c9f680w1c9J,
    p9fcm~z1ze=35,
    f9w8ef~z1ze=(554, 554),
    q9f92ef~b9fμ,
fΛ9J1u~q9f9 = q9f98eu~t1om~tlow~q1wecfouλ(
# 1Λ9J1u0

)
    t11J~woqe=,u69w82f,
    μ0w1zouf9J~t11Jb=1Λ96,
    z00w~Λ9w8e=0.J,
    zμ9w~Λ9w8e=0.J,
    μ618μf~zμ1tf~Λ9w8e=0.J,
    w1qfμ~zμ1tf~Λ9w8e=0.J,
    wof9f1ou~Λ9w8e=12,
    Λ9J1q9f1ou~z11Jf=0.5,
    w82c9Jf=J.522,
q9f98eu = 1w98e89f9c8eu8e9fou(
# 8w8e~8woc8229w8eu8e 8 9w8eu8e 8e 89802

q9f92ef~b9fμ = „D9f92ef„
# c9w1u1u0 89w8 82 1w98eu82

1w80w8f 02
tlow f8u8ouλ1om~k8e82~8w8e8woc822u8e~1w98e 1w80w8f 1w98e89f9c8eu8e9fou
1w80w8f f8u8ouλ1om 82 f1

```

A figura 144 mostra um trecho de código em Python, desenvolvido com a biblioteca OpenCV (`cv2`), cuja finalidade é capturar imagens em tempo real a partir da câmera do computador e aplicar transformações básicas de processamento. Esse tipo de rotina é muito comum em sistemas de visão computacional, servindo como etapa inicial para reconhecimento de objetos, gestos ou padrões visuais.

O programa inicia com a criação de um objeto de captura de vídeo através do comando `cv2.VideoCapture(0)`, o qual acessa a câmera padrão do dispositivo. Esse objeto, denominado comumente como *cap*, permite que os quadros (*frames*) sejam lidos continuamente dentro de um laço de repetição (*while True:*). A cada iteração do *loop*, o método `cap.read()` é chamado, retornando dois valores: um booleano que indica se a leitura foi bem-sucedida (*ret*) e o próprio frame capturado (*frame*).

Após a captura, o código realiza uma série de operações sobre o frame. Primeiramente, ocorre a conversão da imagem original (colorida) para escala de cinza, utilizando o comando `cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)`. Essa etapa é fundamental para reduzir a complexidade da imagem e facilitar a aplicação de algoritmos de detecção, pois elimina as informações de cor e mantém apenas os valores de intensidade luminosa.

Em seguida, o código aplica um filtro gaussiano por meio da função `cv2.GaussianBlur(gray, (5, 5), 0)`. Esse filtro tem a função de suavizar a imagem e reduzir o ruído, tornando as bordas mais definidas e facilitando o trabalho de algoritmos de segmentação e reconhecimento. Logo depois, é utilizada a função `cv2.Canny(blur, 50, 150)` para realizar a detecção de bordas, uma técnica amplamente usada em visão computacional para identificar contornos de objetos.

O resultado dessas transformações é exibido na tela por meio da função `cv2.imshow()`. Geralmente, são abertas duas janelas: uma mostrando o vídeo original capturado pela câmera e outra apresentando o resultado processado com o filtro de bordas aplicado. Isso permite ao usuário observar, em tempo real, como cada etapa do processamento afeta a imagem. O laço principal do programa é mantido até que o usuário pressione a tecla 'q', condição detectada pela instrução `cv2.waitKey(1) & 0xFF == ord('q')`. Quando isso ocorre, o programa interrompe o *loop*, libera o dispositivo de captura com o comando `cap.release()` e fecha todas as janelas abertas utilizando `cv2.destroyAllWindows()`. Essa finalização garante que os recursos de *hardware* e

software utilizados sejam devidamente liberados, evitando erros ou travamentos futuros.

De forma geral, esse trecho de código representa uma estrutura básica de captura e processamento de vídeo em tempo real, sendo um ponto de partida comum para aplicações mais complexas, como sistemas de reconhecimento facial, identificação de gestos manuais, rastreamento de objetos ou monitoramento de movimento. O uso combinado das funções `cv2.cvtColor()`, `cv2.GaussianBlur()` e `cv2.Canny()` demonstra claramente o fluxo de pré-processamento de imagem dentro da *pipeline* de visão computacional, permitindo que o desenvolvedor visualize, compreenda e ajuste cada etapa de acordo com as necessidades do projeto.

Figura 145 - Tela do código do funcionamento da câmera (parte 4)

```
# Modelo simples (baseado no MobileNetV2)
base_model = tf.keras.applications.MobileNetV2(
    input_shape=(224, 224, 3),
    include_top=False,
    weights='imagenet')
(variable) base_model: Any
base_model.trainable = False

model = tf.keras.Sequential([
    base_model,
    tf.keras.layers.GlobalAveragePooling2D(),
    tf.keras.layers.Dense(len(train_data.class_indices), activation='softmax')
])

model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Treinar
history = model.fit(train_data, validation_data=val_data, epochs=10)

# Salvar modelo em TensorFlow Lite
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()

with open("model.tflite", "wb") as f:
    f.write(tflite_model)
```

FONTE: AUTORIA PRÓPRIA, 2025

A figura 145 mostra um trecho de código em Python desenvolvido com o TensorFlow e a biblioteca Keras. Esse código tem como objetivo criar, treinar e converter um modelo de rede neural convolucional. Esse tipo de modelo é muito usado em aplicações de visão computacional, pois é leve, eficiente e pode ser executado até mesmo em dispositivos móveis.

O código começa carregando a rede MobileNetV2 já pré-treinada com o conjunto de dados *ImageNet*, um grande banco de imagens usado para reconhecimento visual. O tamanho das imagens de entrada é definido como 224x224 *pixels* com três canais de cor (vermelho, verde e azul). A configuração “*include_top=False*” remove as camadas originais de saída do modelo, permitindo que novas camadas personalizadas sejam adicionadas depois. Logo em seguida, a linha “*base_model.trainable = False*” congela as camadas da rede base, ou seja, seus pesos não serão alterados durante o treinamento. Essa técnica é conhecida como aprendizado por transferência e é usada para aproveitar o conhecimento que o modelo já tem, ajustando apenas as partes finais para a nova tarefa.

Depois dessa etapa, o código cria um novo modelo sequencial no Keras. Esse novo modelo combina o MobileNetV2 com outras duas camadas adicionais. A primeira é a camada *GlobalAveragePooling2D*, que reduz o tamanho do mapa de características gerado pela rede, resumindo as informações mais importantes. A segunda é uma camada densa com ativação *softmax*, que realiza a classificação final. A quantidade de neurônios dessa camada é definida de acordo com o número de classes presentes nos dados de treinamento, o que permite ao modelo prever a probabilidade de cada classe nas imagens analisadas.

O modelo é então compilado usando o otimizador Adam, a função de perda *categorical_crossentropy*, e a métrica de precisão (*accuracy*). O otimizador Adam ajusta automaticamente a taxa de aprendizado, tornando o treinamento mais eficiente, enquanto a função de perda mede o quanto as previsões do modelo estão distantes dos resultados reais.

Na etapa seguinte, o modelo é treinado com o comando “*model.fit*”, usando os conjuntos de dados de treinamento e validação. Esse processo ocorre durante dez épocas, ou seja, o modelo passa dez vezes por todo o conjunto de dados, ajustando gradualmente seus parâmetros para melhorar o desempenho. Durante o treinamento, o código avalia a acurácia e a perda, permitindo verificar se o modelo está realmente aprendendo de forma consistente.

Depois que o treinamento é concluído, o código converte o modelo treinado para o formato TensorFlow Lite, que é otimizado para rodar em dispositivos com recursos limitados, como celulares, microcontroladores e sistemas embarcados. Essa

conversão é feita por meio do conversor, que transforma o modelo completo em uma versão mais leve e rápida, sem comprometer tanto a precisão.

Por fim, o modelo convertido é salvo em um arquivo chamado “model.tflite”. Esse arquivo pode ser usado em aplicativos e dispositivos que utilizam o TensorFlow Lite para fazer previsões em tempo real, mesmo fora de um ambiente de alto desempenho.

De forma geral, o código apresentado representa um fluxo completo de aprendizado de máquina, passando pelas etapas de criação do modelo, treinamento, avaliação e conversão para uma versão otimizada. Ele é um exemplo típico de como modelos de *deep learning* modernos são desenvolvidos e implantados em aplicações práticas, como reconhecimento de imagens, detecção de objetos e classificação visual automatizada.

CONSIDERAÇÕES FINAIS

Esta sessão tem como objetivo apresentar uma análise dos resultados obtidos ao longo do processo, destacando tanto os objetivos alcançados quanto aqueles que, por diferentes razões, ainda não foram plenamente atingidos. Essa reflexão é fundamental para compreender o desempenho geral do projeto e identificar os pontos fortes que devem ser mantidos, bem como as fragilidades que exigem atenção e ajustes. Os resultados positivos evidenciam o comprometimento da equipe envolvida, a eficácia das estratégias adotadas e a relevância das metas previamente estabelecidas. No entanto, reconhece-se que ainda há desafios a serem superados, especialmente em relação a determinadas metas que não foram completamente atingidas, seja por limitações operacionais, de tempo ou recursos.

Além disso, esta análise permite discutir possíveis melhorias futuras, com o intuito de aprimorar continuamente os processos e resultados. Entre essas melhorias, destacam-se a necessidade de uma maior integração entre os setores, o fortalecimento da comunicação interna, bem como o investimento em capacitação e tecnologias que possam otimizar as etapas de execução. O comprometimento com a melhoria contínua deve permanecer como um princípio orientador, garantindo não apenas a eficiência, mas também a sustentabilidade dos resultados a longo prazo.

OBJETIVOS ALCANÇADO

Ao longo da execução do projeto, foi possível atingir uma série de objetivos técnicos e funcionais, os quais contribuíram diretamente para a consolidação de uma plataforma funcional, moderna e voltada à inclusão. Os resultados obtidos evidenciam o comprometimento da equipe e a viabilidade do projeto dentro do escopo proposto. Um dos primeiros marcos foi a criação da interface do *site*, projetada com foco na experiência do usuário. A interface foi desenvolvida de maneira responsiva, intuitiva e visualmente clara, permitindo que usuários com diferentes níveis de familiaridade tecnológica consigam navegar e interagir com as funcionalidades disponíveis. A usabilidade foi uma das prioridades desde o início, considerando o caráter educativo e inclusivo da plataforma.

Outro objetivo atingido com sucesso foi o desenvolvimento de um sistema de *login* funcional, que possibilita a criação de perfis individuais, controle de acesso e segurança dos dados do usuário. Essa funcionalidade garante um ambiente mais seguro e personalizado, essencial para acompanhar o progresso de cada pessoa dentro da plataforma. A implementação de uma área de atividades interativas também foi concluída com êxito. Nessa seção, os usuários têm acesso a exercícios e desafios que estimulam o aprendizado por meio da prática. Essa abordagem contribui para um processo educacional mais dinâmico e eficiente, além de manter o engajamento dos participantes de maneira mais ativa.

A integração com inteligência artificial (IA) é outro destaque importante do projeto. A IA foi empregada em funcionalidades como reconhecimento de sinais e respostas automatizadas, agregando valor tecnológico ao sistema. Essa integração demonstra o potencial de aplicação de tecnologias emergentes na área da educação e da acessibilidade, ampliando o alcance e a sofisticação da solução. Complementando as ferramentas de apoio ao usuário, foi implementado um sistema de *feedback* automatizado, que fornece respostas imediatas com base no desempenho dos usuários nas atividades realizadas. Essa funcionalidade é essencial para o processo de autoaprendizagem, permitindo que o usuário identifique rapidamente seus acertos e erros, e promova correções de forma autônoma.

Por fim, mesmo que ainda em estágio inicial, foi possível implementar um sistema de reconhecimento de sinais por meio da câmera, o qual demonstrou capacidade de identificar alguns gestos básicos da Língua Brasileira de Sinais (Libras). Esse resultado, ainda que parcial, é extremamente relevante, pois comprova a viabilidade técnica da proposta e serve como base para futuros aprimoramentos.

OBJETIVOS INALCANÇADOS

Apesar do progresso significativo, alguns objetivos previstos inicialmente não puderam ser concluídos dentro do prazo ou da estrutura disponível para o desenvolvimento do projeto. Esses pontos foram identificados como prioridades para fases futuras e são considerados extensões naturais do sistema. Um dos principais itens não concluídos foi a implementação de um sistema de pontuação e notas. Esse recurso seria responsável por mensurar de forma objetiva o desempenho dos usuários nas atividades interativas, permitindo a construção de uma trilha de aprendizado mais clara e a visualização do progresso individual. A ausência dessa funcionalidade limita,

em parte, a capacidade da plataforma de atuar como uma ferramenta completa de avaliação educacional.

Outro objetivo não alcançado foi a tradução de Libras para o português e vice-versa, o que representaria um avanço substancial no quesito acessibilidade. A tradução automática entre línguas diferentes, especialmente quando uma delas é visual-gestual como a Libras, exige tecnologias mais complexas, como redes neurais especializadas, além de uma base de dados mais robusta e diversa. A equipe optou por postergar essa etapa para que seja realizada com a precisão e o cuidado necessários.

A adaptação do sistema para diferentes tipos de sinais, incluindo variações regionais e sinais personalizados da Libras, também não foi possível neste momento. A riqueza e diversidade da Libras, que varia conforme o local, idade e contexto dos usuários, requer um treinamento muito mais amplo e especializado para garantir que o sistema reconheça e interprete corretamente esses sinais. A inclusão dessa funcionalidade exigiria uma fase específica de coleta e análise de dados gestuais em maior escala.

Por fim, embora o reconhecimento de sinais por câmera tenha sido iniciado e demonstrado funcionalidade, a limitação no número de sinais reconhecidos indica que o sistema ainda não está apto para uma aplicação em larga escala. O modelo atual precisaria ser aprimorado com mais ciclos de treinamento, aumento do conjunto de dados e ajustes nos algoritmos de processamento de imagem para atingir uma performance mais precisa e confiável.

ANEXOS

REFERÊNCIAS BIBLIOGRÁFICAS

3PM3. Figma. PM3, 2024. Disponível em: <<https://pm3.com.br/blog/figma/>>. Acesso em: 20 maio 2025.

ADRIANO. PHP: quem é você? DevMedia, 2006. Disponível em: <<https://www.devmedia.com.br/php-quem-e-voce/1797/>>. Acesso em: 03 abr. 2025.

AGATEPEPE. Web. Agatetepe, 2023. Disponível em: <<https://www.agatetepe.com.br/web/>>. Acesso em: 04 jun. 2025.

ALFF, F. Técnicas de levantamento de requisitos. Análise de Requisitos, 2018. Disponível em: <<https://analisederequisitos.com.br/tecnicas-levantamento-requisitos/>> . Acesso em: 17 jun. 2025.

ANIMATION. Sobre. Animation, 2024. Disponível em: <<https://animation905708322.wordpress.com/about/>>. Acesso em: 04 jun. 2025.

ATRA. História das linguagens de programação. Atra Informática, 2024. Disponível em: <<https://www.atrainformatica.com.br/2023/04/05/historia-das-linguagens-de-programacao/>>. Acesso em: 03 abr. 2025.

AWARI. Versões do Python. Awari, 2023. Disponível em: <<https://awari.com.br/python-version-a-evolucao-da-linguagem-de-programacao-python-2/>>. Acesso em: 16 abr. 2025.

AWS. Diferença entre modelo lógico e físico. AWS, 2025. Disponível em: <<https://aws.amazon.com/pt/compare/the-difference-between-logical-and-physical-data-model/>>. Acesso em: 07 abr. 2025.

AWS. O que é Python. AWS, 2025. Disponível em: <<https://aws.amazon.com/pt/what-is/python/>>. Acesso em: 09 mai. 2025.

BAZZARA. Vantagens e desvantagens do *Microsoft Teams*. Barrazacarlos, 2023. Disponível em: <<https://barrazacarlos.com/pt-br/vantagens-e-desvantagens-das-equipes-da-microsoft/>>. Acesso em: 17 jun. 2025.

BEKTAS, A. *User Guiding*: Conheça 16 ferramentas de documentação de *software* essenciais. Bektas, 2025. Disponível em: <<https://userguiding.com/pt-br/blog/ferramentas-documentacao-software/>>. Acesso em: 15 ago.2025.

BOOTSTRAP. Overview. Bootstrap, 2024. Disponível em: <<https://getbootstrap.com.br/docs/4.1/about/overview/>>. Acesso em: 03 abr. 2025.

BRAVEHUB. Inteligência Artificial no YouTube. Bravehub, 2024. Disponível em: <<https://www.youtube.com/watch?v=kBG5dQe394s/>>. Acesso em: 14 maio 2025.

CANVA. Como usar o Canva. Canva, 2025. Disponível em: <<https://ebaonline.com.br/blog/como-usar-o-canva#title4/>>. Acesso em: 20 maio 2025.

CANVA. Linha do tempo 10 anos. Canva, 2023. Disponível em: <https://www.canva.com/pt_br/midia/novidades/canva-10-year-timeline/>. Acesso em: 20 maio 2025.

CAMARADOSDEPUTADOS. Lei nº 13.005, de 25 de junho de 2014. Câmara dos Deputados, 2025. Disponível em: <<https://www2.camara.leg.br/legin/fed/lei/2014/lei-13005-25-junho-2014-778970-publicacaooriginal-144468-pl.html/>>. Acesso em: 01 ago. 2025.

CAMARADOSDEPUTADOS. Lei nº 12.764/2012. Câmara dos Deputados, 2025. Disponível em: <<https://www2.camara.leg.br/legin/fed/lei/2012/lei-12764-27-dezembro-2012-774838-publicacaooriginal-138466-pl.html/>>. Acesso em: 04 ago. 2025.

CAMARADOSDEPUTADOS. Lei nº 13.005, de 25 de junho de 2014. Câmara dos Deputados, 2025. Disponível em: <<https://www2.camara.leg.br/legin/fed/lei/2014/lei-13005-25-junho-2014-778970-publicacaooriginal-144468-pl.html/>>. Acesso em: 01 ago. 2025.

CARDOSO, R. Python. Locaweb, 2024. Disponível em: <<https://www.locaweb.com.br/blog/temas/codigo-aberto/python/>>. Acesso em: 20 maio 2025.

CARDOSO, R. Python Matplotlib. Locaweb, 2022. Disponível em: <<https://www.locaweb.com.br/blog/temas/codigo-aberto/python-matplotlib-conheca-a-biblioteca-de-visualizacao-de-dados/>>. Acesso em: 09 mai. 2025.

CBM. História do Canva. CBM, 2025. Disponível em: <<https://canvasbusinessmodel.com/pt/blogs/brief-history/canva-brief-history/>>.

Acesso em: 20 maio 2025.

CLEMENTE, P. Evolução das linguagens de programação. Antlia, 2025. Disponível em: <<https://antlia.com.br/artigos/evolucao-das-linguagem-de-programacao/>>.

Acesso em: 03 abr. 2025.

CLICKUP. 10 melhores ferramentas de documentação de *software* em 2025.

CLICKUP, 2025. Disponível em: <<https://clickup.com/pt-BR/blog/110939/ferramentas-de-documentacao-de-software/>>.

Acesso em: 15 ago. 2025.

COMPARASOFTWARE. O que é tabela em base de dados. Comparasoftware, 2023.

Disponível em: <<https://blog.comparasoftware.com.br/o-que-e-tabela-em-base-de-dados/>>.

Acesso em: 07 abr. 2025.

CONFEA. Acessibilidade de acordo com a norma ABNT NBR 9050:2020. CONFEA,

2021. Disponível em: <<https://www.confes.org.br/acessibilidade-de-acordo-com-norma-abnt-nbr-90502020/>>.

Acesso em: 04 ago. 2025.

CONTROLENET. Diferença entre servidor local e remoto. ControleNet, 2025.

Disponível em: <<https://www.controle.net/faq/qual-a-diferenca-entre-servidor-local-e-servidor-remoto#:~:text=O%20que%20é%20servidor%20local,na%20rede%20local%20ou%200LAN/>>.

Acesso em: 14 mai. 2025.

COODESH. O que é MariaDB. Coodesh, 2025. Disponível em:

<<https://coodesh.com/blog/dicionario/o-que-e-mariadb/>>.

Acesso em: 16 abr. 2025.

COOPERSYSTEM. Melhores linguagens *mobile*. Coopersystem, 2023. Disponível

em: <<https://www.coopersystem.com.br/as-7-melhores-linguagens-de-programacao-para-desenvolvimento-mobile/>>.

Acesso em: 20 maio 2025.

CURSO DE DISEÑO GRÁFICO ONLINE. O que é o Figma. Curso Diseño Gráfico

Online, 2025. Disponível em: <<https://curso-diseno-grafico.org/pt/o-que-e-o-figma-principais-vantagens-e-caracteristicas/>>.

Acesso em: 04 jun. 2025.

DEVMEDIA. Técnicas de levantamento de requisitos. Devmedia, 2025. Disponível em: <<https://www.devmedia.com.br/tecnicas-para-levantamento-de-requisitos/9151/>>.

Acesso em: 17 jun. 2025.

DIAS, F. BPMN 2.0 – Novos Diagramas e Elementos: Introdução a Coreografia. IPROCESS, 2013. Disponível em: <<https://blog.iprocess.com.br/2013/08/bpmn-2-0-novos-diagramas-e-elementos-introducao-a-coreografia/>>. Acesso em: 15 ago. 2025.

DIDATICATECH. O que é TensorFlow. Didatica Tech, 2024. Disponível em: <<https://didatica.tech/o-que-e-tensorflow-para-que-serve/>>. Acesso em: 09 mai. 2025.

DIDATICATECH. Comando DELETE em SQL. DidaticaTech, 2024. Disponível em: <<https://didatica.tech/tudo-sobre-o-comando-delete-em-sql/>>. Acesso em: 04 jun. 2025.

DOCUMENT360. Top 20 Ferramentas de Documentação de *Software* de 2025. DOCUMENT360, 2025. Disponível em: <<https://document360.com/pt-br/blog/ferramentas-de-documentacao-de-software/>>. Acesso em: 15 ago. 2025.

EDULIBRAS. O que é EduLibras. Edulibras, 2025. Disponível em: <<https://www.edulibras.com.br/>>. e <<https://aws.amazon.com/pt/what-is/text-analysis/>>. Acesso em: 17 jun. 2025.

ETECSP. NSA. EtecSP, 2025. Disponível em: <<https://etecsp.cps.sp.gov.br/nsa/>>. Acesso em: 17 jun. 2025.

FIGMA. Verificação. Figma, 2025. Disponível em: <<https://www.figma.com/verify?fuid=1487244549182506393/>>. Acesso em: 20 maio 2025.

FOTOGRAFIAEBELEZA. Mito do megapixel. Fotografia e Beleza, 2014. Disponível em: <<https://fotografiaebeleza.wordpress.com/2014/08/10/mito-do-megapixel/>>. Acesso em: 04 jun. 2025.

FRANÇA, T. Modelagem de banco de dados. Medium, 2023. Disponível em: <<https://medium.com/@francethais/modelagem-de-banco-de-dados-entidades-relacionamentos-e-atributos-cada1a6e63d0/>>. Acesso em: 07 abr. 2025.

FRANCISCATO, A. A evolução do PHP. DIO, 2023. Disponível em: <<https://www.dio.me/articles/a-evolucao-do-php-da-sua-origem-a-lideranca-na-web-moderna/>>. Acesso em: 03 abr. 2025.

GARCIA, G. XAMPP. Mercado *Online* Digital, 2024. Disponível em: <<https://mercadoonlinedigital.com/blog/xampp/?srsId=AfmBOoov3277Y7ZFy8kjS8yLhnMpa0IASMEAVQpFtCJN2iaZuUAXnfzR/>>. Acesso em: 20 maio 2025.

GITHUB. Langflow. Github, 2025. Disponível em: <https://github.com/langflow-ai/langflow?utm_source=chatgpt.com/>. Acesso em: 14 maio 2025.

GOVBR. VLibras. Gov.br, 2025. Disponível em: <<https://www.gov.br/governodigital/pt-br/ acessibilidade-e-usuario/vlibras/>>. Acesso em: 17 jun. 2025.

GOVBR. Sobre o Centro Paula Souza. GOVBR, 2025. Disponível em: <<https://www.cps.sp.gov.br/institucional/sobre-o-centro-paula-souza/#:~:text=Atualmente%2C%20o%20CPS%20tem%20mais,n%C3%ADvel%20m%C3%A9dio%20e%20superiores%20tecnol%C3%B3gicos/>>. Acesso em: 01 ago. 2025.

GOVBR. Etecs – Centro Paula Souza. GOVBR, 2025. Disponível em: <<https://www.cps.sp.gov.br/etec/etecs/>>. Acesso em: 04 ago. 2025.

GOVBR. Etec Pedro Ferreira Alves. GOVBR, 2025. Disponível em: <<https://www.cps.sp.gov.br/etecs/etec-pedro-ferreira-alves/>>. Acesso em: 04 ago. 2025.

GUILHERME. História do Matplotlib. Phylos, 2021. Disponível em: <[HANASHIRO, M. Evolução do JavaScript. Locaweb, 2024. Disponível em: <<https://www.locaweb.com.br/blog/temas/codigo-aberto/evolucao-do-javascript/>>. Acesso em: 20 maio 2025.](https://phylos.net/2021-09-17/matplotlib#:~:text=site%20de%20Matplotlib.-,Um%20pouco%20de%20história,em%20sua%20pesquisa%20sobre%20epilepsia./>Acesso em: 14 mai. 2025.</p></div><div data-bbox=)

HANDTALK. Como usar o *Hand Talk*. *Handtalk*, 2025. Disponível em: <<https://www.handtalk.me/br/blog/como-usar-o-hand-talk-app-tradutor-de-linguas-de-sinais/>>. Acesso em: 17 jun. 2025.

HONÓRIO, R. O que é *OpenCV* e para que serve. *Asimov Academy*, 2024. Disponível em: <<https://hub.asimov.academy/blog/o-que-e-opencv-e-para-que-serve/>>. Acesso em: 09 mai. 2025.

IBM. Data type fields. IBM Docs, 2025. Disponível em: <<https://www.ibm.com/docs/pt-br/tivoli-netcoolimpact/7.1?topic=types-data-type-fields/>>. Acesso em: 07 abr. 2025.

INKSCAPE. Lançamento Inkscape 1.4. Inkscape, 2025. Disponível em: <<https://inkscape.org/release/inkscape-1.4/windows/64-bit/msi/?redirected=1/>>. Acesso em: 20 maio 2025.

INTUITMAILCHIMP. Formatos de vídeo. Mailchimp, 2024. Disponível em: <<https://mailchimp.com/pt-br/resources/video-formats/>>. Acesso em: 04 jun. 2025.

IBM. *Fix list* for IBM WebSphere Application Server traditional V9.0. Disponível em: <<https://www.ibm.com/support/pages/fix-list-ibm-websphere-application-server-traditional-v9-0/>>. Acesso em: 19 set. 2025.

JUMP. *Machine Learning*: a evolução do aprendizado de máquina. *Jump*, 2025. Disponível em: <<https://jump.tec.br/blog/machine-learning-a-evolucao-do-aprendizado-de-maquina/>>. Acesso em: 14 maio 2025.

JUSBRASIL. Lei nº 12.907/2008. JusBrasil, 2025. Disponível em: <<https://www.jusbrasil.com.br/legislacao/156600/lei-12907-08/>>. Acesso em: 01 ago. 2025.

KOVACS, L. O que é e como funciona o Microsoft Teams. Tecnoblog, 2020. Disponível em: <<https://tecnoblog.net/responde/o-que-e-e-como-funciona-o-microsoft-teams/>>. Acesso em: 17 jun. 2025.

LANGFLOW. Documentação *Langflow*. *Langflow*, 2025. Disponível em: <<https://docs.langflow.org/>>. Acesso em: 14 maio 2025.

LANGFLOW. *Helm Charts*. *Langflow*, 2025. Disponível em: <<https://github.com/langflow-ai/langflow-helm-charts/>>. Acesso em: 20 maio 2025.

LEGISLAÇÃO DIGITAL. Lei Ordinária nº 5.784/2015. Legislação Digital, 2025. Disponível em: <<https://legislacaodigital.com.br/Sumare-SP/LeisOrdinarias/5784-2015/>>. Acesso em: 04 ago. 2025.

LEONVANZYL. Introdução ao *Langflow*. YouTube, 2024. Disponível em: <https://www.youtube.com/watch?v=VbqfZjT_PLE/>. Acesso em: 20 maio 2025.

LESCANO, J. Canva e Inkscape. O Segredo, 2023. Disponível em: <https://osegredo.com.br/canva/#google_vignette e https://wiki.inkscape.org/wiki/Release_notes/0.35#Inkscape_Release_0.35/>. Acesso em: 20 maio 2025.

LESSENKO, C. BPMN 2.0: a padronização da modelagem de processos. LESSENKO, 2024. Disponível em: <<https://www.neomind.com.br/blog/bpmn-2-0-a-padronizacao-da-modelagem-de-processos/>>. Acesso em: 15 ago. 2025.

LIMA, G. Bootstrap. Alura, 2023. Disponível em: <https://www.alura.com.br/artigos/bootstrap?srsIid=AfmBOorwsU0XjoJC_UOawnJV Dq1zFIOYse2tOSUcPj0DB55ZjeGilF8l/>. Acesso em: 03 abr. 2025.

LUCID. Estudo de viabilidade. Lucid, 2025. Disponível em: <<https://lucid.co/pt/blog/como-fazer-um-estudo-de-viabilidade/>>. Acesso em: 17 jun. 2025.

LUCIDCHART. O que é um diagrama UML? LUCIDCHART, 2025. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-uml/>>. Acesso em: 15 ago. 2025.

LUCIDCHART. O que é um diagrama UML? Por que usar um diagrama UML. LUCIDCHART, 2025. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-uml/>>. Acesso em: 22 ago. 2025.

MAIA, F. *Microsoft Teams. Lean Solutions*, 2024. Disponível em: <<https://www.leansolutions.com.br/blog/microsoft-teams/>>. Acesso em: 17 jun. 2025.

MARQUES, R. MySQL 8: novidades. Homehost, 2024. Disponível em: <<https://www.homehost.com.br/blog/tutoriais/mysql/mysql-8-as-novidades-da-nova-versao-do-mysql/>>. Acesso em: 16 abr. 2025.

MDN. JavaScript. MDN, 2025. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/>>. Acesso em: 20 maio 2025.

MELO, D. O que é HTML: guia para iniciantes. Tecnoblog, 2021. Disponível em: <<https://tecnoblog.net/responde/o-que-e-html-guia-para-iniciantes/>>. Acesso em: 03 abr. 2025.

MIRANDA, J. *Hand Tracking com OpenCV*. Alura, 2024. Disponível em: <[MIRO. O guia definitivo para diagramas UML. MIRO, 2025. Disponível em: <\[MONTOVANI, I. O que é Gimp. MKT *Esports*, 2025. Disponível em: <\\[MYSQL. O que é MySQL. UDESC, 2025. Disponível em: <\\\[NAPOLEAN. O que é Gimp. Napoleon, 2024. Disponível em: <\\\\[NETCODERS. História do HTML. Netcoders, 2024. Disponível em: <\\\\]\\\\(https://napoleon.com.br/glossario/o-que-e-gimp/>. Acesso em: 04 jun. 2025.</p></div><div data-bbox=\\\\)\\\]\\\(https://ceavi.udesc.br/arquivos/id_submenu/487/o_que_e_o_mysql.pdf/>. Acesso em: 16 abr. 2025.</p></div><div data-bbox=\\\)\\]\\(https://mktesports.com.br/blog/stream/o-que-e-gimp/>. Acesso em: 04 jun. 2025.</p></div><div data-bbox=\\)\]\(https://miro.com/pt/diagrama/o-que-e-uml/>. Acesso em: 15 ago. 2025.</p></div><div data-bbox=\)](https://www.alura.com.br/conteudo/visao-computacional-hand-tracking-opencv#:~:text=O%20MediaPipe%20é%20um%20framework,disponível%20para%20a%20linguagem%20Python/>. Acesso em: 09 mai. 2025.</p></div><div data-bbox=)

ORACLE. O que é *Machine Learning*. Oracle, 2025. Disponível em: <[ORACLE. Python para desenvolvedores. Oracle, 2025. Disponível em: <\[O'REILLY. *Mastering OpenCV 4*. O'Reilly, 2025. Disponível em: <\\[OTSUKA, J. O que é Numpy e onde é utilizado. EBAC, 2025. Disponível em: <\\]\\(https://www.oreilly.com/library/view/mastering-opencv-4/9781789533576/2de0893d-e450-417d-b336-a4143799b43d.xhtml/>. Acesso em: 09 mai. 2025.</p></div><div data-bbox=\\)\]\(https://www.oracle.com/br/developer/what-is-python-for-developers/>. Acesso em: 20 maio 2025.</p></div><div data-bbox=\)](https://www.oracle.com/br/artificial-intelligence/machine-learning/what-is-machine-learning/>. Acesso em: 14 maio 2025.</p></div><div data-bbox=)

PACIEVITCH, Y. *Cascading Style Sheets (CSS)*. InfoEscola, 2020. Disponível em: <<https://www.infoescola.com/informatica/cascading-style-sheets-css/>>. Acesso em: 03 abr. 2025.

PCDEMFOCO. *Hand Talk*. PC em Foco, 2025. Disponível em: <<https://pcdemfocoblog.com/hand-talk/>>. Acesso em: 17 jun. 2025.

PINHEIRO, A. O que é e como funciona algoritmo e lógica de programação. *Estude Sem Fronteiras*, 2024. Disponível em: <<https://blog.estudesemfronteiras.com/o-que-e-e-como-funciona-algoritmo-e-logica-de-programacao/>>. Acesso em: 03 abr. 2025.

PLANALTO. Lei nº 13.005/2014. Presidência da República, 2014. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2014/lei/l13005.htm/>. Acesso em: 01 ago. 2025.

PLANALTO. Lei nº 13.709/2018. Presidência da República, 2018. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm/>. Acesso em: 01 ago. 2025.

PLANALTO. Lei nº 13.146/2015. Presidência da República, 2015. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm/>. Acesso em: 01 ago. 2025.

PLANALTO. Lei de Diretrizes e Bases da Educação Nacional – Lei nº 9.394/1996. Presidência da República, 2015. Disponível em: <https://www.planalto.gov.br/ccivil_03/leis/l9394.htm/>. Acesso em: 01 ago. 2025.

PLANALTO. Lei nº 10.436/2002. Presidência da República, 2018. Disponível em: <https://www.planalto.gov.br/ccivil_03/leis/2002/l10436.htm/>. Acesso em: 01 ago. 2025.

PLANALTO. Decreto nº 9.854/2019. Presidência da República, 2019. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2019-2022/2019/decreto/d9854.htm/>. Acesso em: 01 ago. 2025.

PLANALTO. Lei nº 12.764/2012. Presidência da República, 2014. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/lei/l12764.htm/>. Acesso em: 01 ago. 2025.

PLANALTO. Lei nº 9.279/1996. Presidência da República, 2014. Disponível em: <https://www.planalto.gov.br/ccivil_03/leis/l9279.htm/>. Acesso em: 01 ago. 2025.

PLANALTO. Lei nº 9.610/1998. Presidência da República, 2015. Disponível em: <https://www.planalto.gov.br/ccivil_03/leis/l9610.htm/>. Acesso em: 01 ago. 2025.

PLANALTO. Código de Defesa do Consumidor – Lei nº 8.078/1990. Presidência da República, 2015. Disponível em: <https://www.planalto.gov.br/ccivil_03/leis/l8078compilado.htm/>. Acesso em: 01 ago. 2025.

PLANALTO. Lei nº 14.133/2021. Presidência da República, 2021. Disponível em: <https://www.planalto.gov.br/ccivil_03/_ato2019-2022/2021/lei/l14133.htm0/>. Acesso em: 01 ago. 2025.

PNE. Plano Nacional de Educação – Lei nº 13.005/2014. PNE, 2025. Disponível em: <<https://pne.mec.gov.br/18-planos-subnacionais-de-educacao/543-plano-nacional-de-educacao-lei-n-13-005-2014/>>. Acesso em: 01 ago. 2025.

PORTAL DE CONTEÚDOS. História do Canva. O Novo Mercado, 2025. Disponível em: <<https://onovomercado.com/blog/historia-canva/>>. Acesso em: 20 maio 2025.

PRETTO, R. Artigo HTML. Faccat, 2019. Disponível em: <<https://fit.faccat.br/~sorgetz/ArtigoHTML.pdf/>>. Acesso em: 03 abr. 2025.

REIS, A. Evolução do CSS. DIO, 2023. Disponível em: <<https://www.dio.me/articles/evolucao-do-css/>>. Acesso em: 03 abr. 2025.

RICARDO. Conceitos fundamentais de banco de dados. *DevMedia*, 2006. Disponível em: <<https://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649/>>. Acesso em: 03 abr. 2025.

SILVA, F. História do PHP. *KingHost*, 2024. Disponível em: <<https://king.host/blog/tecnologia/historia-do-php/>>. Acesso em: 03 abr. 2025.

SILVA, B. Python: o que é. *Hostinger*, 2024. Disponível em: <<https://www.hostinger.com.br/tutoriais/python-o-que-e/>>. Acesso em: 20 maio 2025.

SILVÉRIO, R. JavaScript. Alura, 2023. Disponível em: <<https://www.alura.com.br/artigos/javascript?srsId=AfmBOopK15Qyi72F3LX2eXcWUg-clg1inaQ2Qrk-n9vSRWylrRwrEzGt/>>. Acesso em: 03 abr. 2025.

SILVEIRA, M. *Figma*. Alura, 2024. Disponível em: <<https://www.alura.com.br/artigos/figma/>>. Acesso em: 20 maio 2025.

SINGLIBRAS. *Hand Talk*. SingLibras, 2020. Disponível em: <<https://www.singlibras.com/hand-talk/>>. Acesso em: 17 jun. 2025.

TECHTUDO. *Google Chrome*. Techtudo, 2025. Disponível em: <<https://www.techtudo.com.br/tudo-sobre/google-chrome/>>. Acesso em: 14 maio 2025.

TECHTUDO. *ProDeaf*. Techtudo, 2025. Disponível em: <<https://www.techtudo.com.br/tudo-sobre/prodeaf/>>. Acesso em: 17 jun. 2025.

TECNOBLOG. *Microsoft Edge*. Tecnoblog, 2025. Disponível em: <<https://tecnoblog.net/sobre/microsoft-edge/>>. Acesso em: 14 maio 2025.

TOTVS. BPMN: entenda o que é a modelagem de processos de negócios, como fazer e sua importância. TOTVS, 2023. Disponível em: <<https://www.totvs.com/blog/gestao-industrial/bpmn/>>. Acesso em: 15 ago. 2025.

UNIFIQUE. VLibras. Unifique, 2025. Disponível em: <<https://unifique.com.br/acessibilidade/vlibras/>>. Acesso em: 17 jun. 2025.

UNINORTE. Aplicativo *ProDeaf* móvel. Uninorte, 2018. Disponível em: <<https://www.uninorte.com.br/aplicativo-prodeaf-movel-facilita-comunicacao-entre-alunos-conheca/>>. Acesso em: 17 jun. 2025.

VIEIRA, D. Conheça o HTML. Hostgator, 2023. Disponível em: <<https://www.hostgator.com.br/blog/conheca-o-html/>>. Acesso em: 04 jun. 2025.

VIEIRA, V. VLibras premiado. UFPB, 2024. Disponível em: <<https://www.ufpb.br/ufpb/contents/noticias/ferramentas-vlibras-desenvolvida-por-pesquisadores-da-ufpb-e-uma-das-vencedoras-de-premiacao-internacional/>>. Acesso em: 17 jun. 2025.

WEBPARATODOS. Lançamento da norma de acessibilidade em aplicativos móveis. WEBPARATODOS, 2022. Disponível em: < <https://mwpt.com.br/abnt-lanca-norma-de-acessibilidade-em-aplicativos-moveis/>>. Acesso em: 04 ago. 2025.

WEBPARATODOS. WCAG – As diretrizes de acessibilidade para o conteúdo da web de forma descomplicada. WEBPARATODOS, 2020. Disponível em: <<https://mwpt.com.br/wcag-as-diretrizes-de-acessibilidade-para-o-conteudo-da-web-de-forma-descomplicada/>>. Acesso em: 15 ago. 2025.