UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO

CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE

DEPARTAMENTO DE COMPUTAÇÃO

SISTEMAS DE INFORMAÇÃO

Eriani Moreira da Silva

SISTEMA WEB PARA A ESCOLA PIO XII COM RECURSOS DE ACESSIBILIDADE

Alegre, ES

2025

Eriani Moreira da Silva

SISTEMA WEB PARA A ESCOLA PIO XII COM RECURSOS DE ACESSIBILIDADE

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.
Orientador: Prof.º M.Sc. Giuliano Prado De Morais Giglio

Alegre, ES

2025

Eriani Moreira da Silva

Sistema Web para a escola PIO XII com recursos de acessibilidade

Trabalho de Conclusão de Curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado em 20 de agosto de 2025.

COMISSÃO EXAMINADORA

Prof. MSc. Giuliano Prado de Morais Giglio
Universidade Federal do Espírito Santo
Orientador

Prof.ª Dra. Simone Dornelas Costa
Universidade Federal do Espírito Santo

Prof.ª Dra. Valéria Alves da Silva
Universidade Federal do Espírito Santo

Àqueles que, com esforço e dedicação, se empenham em tornar o mundo um lugar melhor para todos.

Aos meus pais, Paulo Sérgio e Luciana, e minha irmã, Elizandra, por serem fonte constante de inspiração, pelo apoio incansável e pelo amor incondicional.

Aos amigos e colegas, pelo incentivo, parceria e companheirismo durante toda a caminhada acadêmica.

A todos os professores e mentores que, com sabedoria e dedicação, contribuíram significativamente para minha formação.

Agradecimentos

Em primeiro lugar, agradeço a Deus, que foi minha força em todos os momentos de dúvida, cansaço e medo. Foi Nele que encontrei refúgio, coragem e serenidade para continuar mesmo quando tudo parecia difícil.

Agradeço imensamente ao meu orientador, Prof. M.Sc. Giuliano Prado de Morais Giglio, por ter acreditado no meu trabalho desde o início. Sua orientação atenta, suas palavras de incentivo e sua disponibilidade em compartilhar conhecimento fizeram toda a diferença no desenvolvimento desta pesquisa.

Aos professores do Departamento de Computação, deixo minha sincera gratidão por cada aula, cada explicação paciente, cada desafio proposto. Sei que o que levo comigo vai muito além do conteúdo técnico — foram anos de crescimento intelectual, profissional e pessoal. Cada um de vocês contribuiu de forma única para minha formação.

Aos meus pais, Paulo Sérgio e Luciana, meu amor mais profundo e minha eterna admiração. Obrigada por nunca medirem esforços para me apoiar, mesmo nos dias mais difíceis. Por cada gesto de carinho, cada palavra de consolo, cada sacrifício feito em silêncio. Esta vitória é de vocês também — vocês são parte essencial dela.

À minha irmã Elizandra, que esteve ao meu lado mesmo nos momentos em que pensei em desistir. Sua presença firme, suas palavras de incentivo e seu jeito acolhedor foram essenciais para que eu não perdesse a fé em mim mesma. Nos dias mais difíceis, quando tudo parecia demais, foi ela quem me lembrou da importância da paciência, da leveza e de continuar, um passo de cada vez. Seu apoio foi meu abrigo, meu alívio, meu porto seguro em meio ao caos de prazos, dúvidas e cansaços.

Às professoras M.Sc. Ana Carolina de Melo Loureiro e M.Sc. Prisciliana Costa Ventura, meu muito obrigada pela confiança no meu trabalho e pela oportunidade de desenvolver o sistema vinculado ao projeto “Parede da Memória”. Foi uma experiência enriquecedora, que levarei com carinho e orgulho.

Aos meus amigos e colegas, que caminharam comigo durante essa formação. Obrigada por cada incentivo, cada parceria, cada momento de cumplicidade. Sem vocês, tudo teria sido mais difícil e muito menos divertido.

E por fim, mas não menos importante, agradeço a todas as pessoas que, direta ou indiretamente, contribuíram para a conclusão desta pesquisa.

Se não conseguir de primeira coma um pedaço de bolo.

Charlie Mackesy

Resumo

Este trabalho apresenta detalhes acerca do desenvolvimento de um sistema Web para a escola PIO XII, que possui como funcionalidade postagens de conteúdos, principalmente de fotografias, contendo recordações da cidade de São Mateus - ES. O planejamento e desenvolvimento do sistema envolveu a utilização de recursos de acessibilidade, mostrando a importância, principalmente, na inclusão social de disponibilizar sites acessíveis para a utilização de qualquer usuário, sem que os mesmos tenham dificuldade de acesso à informação disponível. Para o desenvolvimento, foi feito uso de diretrizes e tecnologias assistivas que são padronizadas para melhorar a acessibilidade de qualquer sistema. No âmbito mais técnico, utilizou-se de tecnologias de desenvolvimento web, como Spring Boot, Angular e BootStrap. O trabalho também contribui tecnicamente na aplicação de técnicas de acessibilidade em aplicações web, além de ampliar a conscientização sobre a preocupação com a acessibilidade, dentro e fora da Web e, dessa forma, dando acesso a qualquer usuário que desejar acessar as informações de um site.

Palavras-chave: Acessibilidade digital; Inclusão social; W3C; Aplicação Web.

Abstract

This work presents the development of a web system for the PIO XII school, designed to allow the publication of content—especially photographs—related to memories of the city of São Mateus, ES. The planning and implementation of the system prioritized accessibility features, emphasizing the importance of promoting social inclusion through digital platforms accessible to all users, regardless of their limitations. To achieve this, standardized guidelines and assistive technologies were employed to enhance the system's usability and accessibility. From a technical perspective, the system was developed using modern web technologies such as Spring Boot, Angular, and Bootstrap. In addition to its technical contributions, this work promotes awareness of digital accessibility practices both within and beyond the web environment, enabling broader access to information for all users.

Keywords: Digital accessibility; Social inclusion; W3C; Web application.

Lista de Figuras

Figura 1 – Representação do WCAG 2.2 27
Figura 2 – Representação de uma API 34
Figura 3 – Desenho da Arquitetura 36
Figura 4 – Ciclo de Vida do SCRUM 37
Figura 5 – Desenvolvimento de um sistema - Ciclo de Vida 38
Figura 6 – Processo de engenharia de requisitos 39
Figura 7 – Diagrama de Caso de Uso 43
Figura 8 – Diagrama de Classe 47
Figura 9 – Modelo lógico do Sistema 48
Figura 10 – Modelo físico do Sistema 49
Figura 11 – Tela do quadro de organização das atividades do sistema no Notion . . 54
Figura 12 – Estrutura do projeto *back-end* 55
Figura 13 – Exemplo de login realizado pelo *back-end* 64
Figura 14 – Estrutura do projeto *front-end* 65
Figura 15 – Configuração do serviço de envio de e-mail 78
Figura 16 – Configuração do template de envio de e-mail 79
Figura 17 – Imagem da tela inicial do sistema. 81
Figura 18 – Imagem da tela inicial do sistema com menu expandido 82
Figura 19 – Imagem da tela "Fale Conosco"com informações de contato 82
Figura 20 – Imagem da tela "Fale Conosco"com formulário de visita 83
Figura 21 – Imagem da tela da modal apresentando a mensagem de sucesso ao enviar solicitação de agendamento 83
Figura 22 – Imagem do e-mail enviado pelo sistema ao realizar solicitação de agendamento. 84
Figura 23 – Imagem da tela de listagens dos projetos 84
Figura 24 – Imagem da tela de listagens de publicações 85
Figura 25 – Imagem da tela de detalhes da publicação 86
Figura 26 – Imagem da tela de informações sobre o projeto 86
Figura 27 – Imagem da tela de listagem dos colaboradores 87
Figura 28 – Imagem da tela de login do sistema 87
Figura 29 – Imagem da tela da modal apresentando a mensagem de sucesso ao logar. 88
Figura 30 – Imagem da tela “Fale Conosco” com opção de edição. 88
Figura 31 – Imagem da tela “Fale Conosco” com os campos de Telefone e E-mail. . 89
Figura 32 – Imagem da tela “Fale Conosco” com os campos de Rede Social 89
Figura 33 – Imagem da tela “Nossos Projetos” com acesso de administrador master. 90
Figura 34 – Imagem da tela para adicionar novo projeto. 90

Figura 35 – Imagem da tela para editar projeto. 91
Figura 36 – Imagem da tela da modal de confirmação de exclusão. 91
Figura 37 – Imagem da tela “Publicações” com acesso de administrador master e/ou coadministrador. 92
Figura 38 – Imagem da tela de adicionar publicação. 92
Figura 39 – Imagem da tela de visualização de publicação com acesso de administrador 93
Figura 40 – Imagem da tela de visualização de "Quem somos"com acesso de administrador 93
Figura 41 – Imagem da tela de adicionar colaborador. 94
Figura 42 – Imagem da tela para editar colaborador. 94
Figura 43 – Imagem da tela de gerenciamento dos agendamentos de visitas 95
Figura 44 – Imagem da tela da modal de confirmação de agendamento de visita. . . 96
Figura 45 – Imagem da tela da modal de confirmação de exclusão. 96
Figura 46 – Imagem da tela de gerenciamento dos usuários 97
Figura 47 – Imagem da tela da modal de confirmação de exclusão 97
Figura 48 – Imagem da tela de adicionar usuário 98
Figura 49 – Imagem da tela do menu principal apresentando foco ao navegar utilizando o teclado. 100
Figura 50 – Imagem da tela de projetos apresentando texto alternativo para logotipo. 101
Figura 51 – Imagem da tela de projetos apresentando hierarquia da página. 101
Figura 52 – Imagem da tela principal na versão mobile. 102
Figura 53 – Imagem da tela da modal apresentando a mensagem clara de sucesso ao logar. 104
Figura 54 – Imagem da tela principal com o zoom aplicado. 104
Figura 55 – Imagem da tela da modal solicitando a confirmação antes de excluir usuário. 105
Figura 56 – Imagem da modal de recuperação de dados. 106
Figura 57 – Taxa de Contraste e seus níveis. 107
Figura 58 – Imagem da tela principal com alterações de contraste. 107
Figura 59 – Imagem da tela “Sobre o Projeto” com interação do VLibras. 108
Figura 60 – Imagem da tela “Sobre o Projeto” com botão de "Retornar ao topo". . 109
Figura 61 – Imagem da tela “Quem Somos” com lista de itens disponível através de links. 110
Figura 62 – Imagem da tela “Nossos projetos” com lista de itens disponível através de links. 110
Figura 63 – Imagem da tela de visualização de publicação com foco no Breadcrumb. 111
Figura 64 – Imagem da tela “Nossos Projetos” com aba do navegador descritiva. . . 111
Figura 65 – Tela do Lighthouse 115
Figura 66 – Tela inicial do sistema com extensão WAVE 116

Figura 67 – Tela da aplicação Color Contrast Analyzer 117
Figura 68 – Tela da extensão Dark Reader 118
Figura 69 – Tela da extensão Colorblindly 119
Figura 70 – Tela do site AcessMonitor 120
Figura 71 – Tela do Validador W3C 121
Figura 72 – Tela da extensão Axe DevTools para testes automatizados 123
Figura 73 – Tela da extensão Axe DevTools para testes guiados 123
Figura 74 – Tela da extensão HeadingsMAp 125
Figura 75 – Tela da extensão Funkity 125
Figura 76 – Tela da aplicação NVDA 126
Figura 77 – Imagem da tela da modal de erro ao excluir projeto com publicações vinculadas. 167
Figura 78 – Imagem da tela da modal de erro ao excluir projeto com colaboradores vinculados. 167
Figura 79 – Imagem da tela de Publicações ao realizar uma filtragem sem retorno. . 168
Figura 80 – Imagem da tela da modal de erro ao preencher campo de telefone incorretamente. 168
Figura 81 – Imagem da tela da modal de confirmação ao sair da tela sem salvar as informações. 169
Figura 82 – Imagem da tela de cadastro de usuários demonstrando erros de campo obrigatório no formulário. 169
Figura 83 – Imagem da tela do formulário para solicitar uma visita demonstrando erro do campo data futura. 170
Figura 84 – Imagem da tela do formulário para adicionar um projeto demonstrando a alteração quando o limite de caracteres é alcançado. 170

Lista de Quadros

Quadro 1 – Usuários do Sistema. 40
Quadro 2 – Quadro do banco de dados - SubProjeto 50
Quadro 3 – Descrição da estrutura do projeto. 56
Quadro 4 – Descrição da estrutura do projeto *front-end*. 65
Quadro 5 – Descrição da estrutura da função emailjs.send() 81
Quadro 6 – Descrição de resultados dos testes usando Lighthouse. 115
Quadro 7 – Descrição de resultados dos testes usando extensão WAVE. 117
Quadro 8 – Descrição de resultados dos testes usando Color Contrast Analyzer, Colorblindly e Dark Reader. 119
Quadro 9 – Descrição de resultados dos testes usando AcessMonitor e Validador W3C. 121
Quadro 10 – Descrição de resultados dos testes de responsividade. 122
Quadro 11 – Resultados dos testes usando Axe DevTools 124
Quadro 12 – Descrição de resultados dos testes usando Funkify. 126
Quadro 13 – Descrição de resultados dos testes usando NVDA. 127
Quadro 14 – Resultado das validações dos critérios de sucesso WCAG - Perceptível 128
Quadro 15 – Descrição de resultados da reunião com orientador. 129
Quadro 16 – Descrição de resultados da primeira reunião com partes interessadas. . 130
Quadro 17 – Descrição de resultados da terceira reunião com partes interessadas. . 131
Quadro 18 – Quadro do banco de dados - Usuário 165
Quadro 19 – Quadro do banco de dados - Formulário 165
Quadro 20 – Quadro do banco de dados - Publicação 165
Quadro 21 – Quadro do banco de dados - Foto 165
Quadro 22 – Quadro do banco de dados - Colaborador 166
Quadro 23 – Quadro do banco de dados - Colaborador-Subprojeto 166
Quadro 24 – Quadro do banco de dados - Email 166
Quadro 25 – Quadro do banco de dados - Rede Social 166
Quadro 26 – Quadro do banco de dados - Telefone 166

Lista de abreviaturas e siglas

API	Application Programming Interface
ARIA	Accessible Rich Internet Applications
CAT	Comitê de Ajudas Técnicas
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DER	Diagrama Entidade-Relacionamento
ER	Entidade-Relacionamento
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
IBGE	Instituto Brasileiro de Geografia e Estatística
JSON	JavaScript Object Notation
PNAD	Pesquisa Nacional por Amostra de Domínios
PRO-TI	Projeto de Assessoria Socialmente Responsável em Tecnologia da Informação
REST	Representational State Transfer
SGBD	Sistema de Gerenciamento de Banco de Dados
UFES	Universidade Federal do Espírito Santo
UML	Unified Modeling Language
WAI	Web Accessibility Initiative
WCAG	Diretrizes de Acessibilidade para Conteúdo Web
WWW	World Wide Web
W3C	World Wide Web Consortium

Sumário

1 INTRODUÇÃO . . . 16
1.1 O problema e sua importância . . . 17
1.2 Objetivos . . . 18
1.2.1 Objetivo Geral . . . 18
1.2.2 Objetivos Específicos . . . 19
1.3 Motivação . . . 19

2 REVISÃO DA LITERATURA . . . 20
2.1 Projeto "Parede da Memória" . . . 20
2.2 Acessibilidade na Web . . . 21
2.2.1 Design Universal . . . 21
2.2.2 Tecnologias Assistivas . . . 22
2.2.3 Diretrizes de Acessibilidade na Web . . . 25
2.2.3.1 WCAG . . . 25
2.2.3.1.1 Principios da WCAG . . . 26
2.2.3.2 WAI-ARIA . . . 28
2.2.3.3 E-MAG . . . 28
2.2.4 Usabilidade . . . 28
2.3 Tecnologias Envolvidas . . . 29
2.3.1 HTML . . . 29
2.3.2 CSS . . . 29
2.3.3 BootStrap . . . 30
2.3.4 JavaScript . . . 31
2.3.5 Angular . . . 31
2.3.6 Spring Boot . . . 32
2.4 Arquitetura do Sistema . . . 32
2.4.1 API . . . 33
2.4.2 REST . . . 34
2.4.3 Modelo da arquitetura do sistema . . . 35
2.5 Metodologia Scrum . . . 37

3 METODOLOGIA . . . 38
3.1 Análise dos Resquisitos . . . 39
3.1.1 Descrição dos usuários . . . 40
3.1.2 Requisitos Funcionais . . . 41
3.1.3 Requisitos não funcionais . . . 42

3.1.4 Casos de Uso . . . 43
3.1.4.1 Narrativas de Caso de Uso . . . 44
3.1.5 Diagrama de Classe . . . 46
3.2 Modelagem Banco de Dados . . . 48
3.2.1 Modelo Lógico . . . 48
3.2.2 Modelo Físico . . . 49
3.2.3 Dicionário de dados . . . 50
3.3 Avaliação de sistemas web . . . 50
3.3.1 Avaliação de Acessibilidade na Web . . . 51
3.3.1.1 Ferramentas avaliadoras de Acessibilidade . . . 52

4 DESENVOLVIMENTO . . . 54
4.1 Organização dos processos . . . 54
4.2 A aplicação . . . 55
4.2.1 Back-end . . . 55
4.2.1.1 Exemplificação do Back-end . . . 56
4.2.1.1.1 Model . . . 56
4.2.1.1.2 DTO . . . 58
4.2.1.1.3 Controller . . . 58
4.2.1.1.4 Service . . . 59
4.2.1.1.5 Repository . . . 62
4.2.1.2 Conexão com o banco de dados . . . 62
4.2.1.3 Segurança do sistema . . . 64
4.2.2 Front-end . . . 64
4.2.2.1 Exemplificação do Front-end . . . 65
4.2.2.1.1 Component . . . 66
4.2.2.1.2 Service . . . 74
4.2.2.2 Rotas de navegação . . . 76
4.2.3 Serviço de envio de e-mail . . . 77
4.2.3.1 Configuração do EmailJS . . . 78
4.2.3.2 Integração no código Angular . . . 79
4.2.4 Interfaces . . . 81
4.2.4.1 Telas do sistema . . . 81
4.3 Acessibilidade . . . 98
4.3.1 Declaração do Idioma da Página . . . 98
4.3.2 Alinhamento de textos . . . 99
4.3.3 Navegação por Teclado e Foco Visual . . . 99
4.3.4 Texto Alternativo para Imagens (Alt Text) . . . 100
4.3.5 Estrutura Semântica do HTML . . . 101
4.3.6 Design Responsivo . . . 102

4.3.7 Rotulagem de Formulários (Labels) . . . 102
4.3.8 Mensagens de Status Acessíveis . . . 103
4.3.9 Funcionalidade de Zoom: Acessibilidade e Flexibilidade Visual . . . 104
4.3.10 Confirmação para Alterações Críticas: Prevenção de Erros Irreversíveis . . . 105
4.3.11 Recuperação de dados após logout inesperado . . . 106
4.3.12 Contraste de Cores . . . 106
4.3.13 Integração com VLibras: Acessibilidade para a Comunidade Surda . . . 108
4.3.14 Botão "Retornar ao Topo": Navegação Acessível e Facilitada . . . 108
4.3.15 Modal Acessível: Foco Preso e Fechamento por Teclado . . . 109
4.3.16 Navegação por Atalhos: Acesso Rápido a Conteúdos Repetidos . . . 110
4.3.17 Uso de Breadcrumbs: Orientação e Navegação Hierárquica . . . 110
4.3.18 Títulos de Páginas Intuitivos: Clareza na Navegação . . . 111
4.3.19 Uso de Atributos ARIA . . . 111

5 GARANTIA DE QUALIDADE E PROCESSO DE TESTES . . . 114
5.1 Testes com apoio de ferramentas . . . 114
5.1.1 Lighthouse . . . 114
5.1.2 WAVE (Ferramenta de Avaliação de Acessibilidade da Web) . . . 116
5.1.3 Contraste . . . 117
5.1.4 Validação de HTML . . . 120
5.1.5 Simulador para responsividade . . . 121
5.1.6 Axe DevTools . . . 122
5.1.7 HeadingsMAp . . . 124
5.1.8 Funkify . . . 125
5.1.9 NVDA . . . 126
5.2 Testes Manuais . . . 127
5.2.1 WCAG . . . 127
5.2.2 Reuniões . . . 129
5.2.2.1 Implantação . . . 131

6 DISCUSSÃO DOS RESULTADOS E CONSIDERAÇÕES FINAIS . . . 133
6.1 Trabalhos Futuros . . . 136

REFERÊNCIAS . . . 138

APÊNDICE A - Critérios de Sucesso - WCAG 2.2 . . . 144

APÊNDICE B - Narrativas de casos de uso . . . 151

APÊNDICE C - Dicionário de dados . . . 165

APÊNDICE D - Telas do Sistema . . . 167

1 Introdução

É inegável que o uso da tecnologia tem bastante presença na sociedade contemporânea, principalmente na organização das práticas sociais, seja de maneira complexa ou rudimentar (BENAKOUCHE, 1999). Dessa forma, os avanços da tecnologia ao longo do tempo se tornaram essenciais para a contribuição de uma gestão cada vez melhor quando se refere aos processos realizados e à disseminação de informação (LAMPKOWSKI; SILVA, 2014).

Assim é visto que, sendo uma ferramenta muito atual e com presença cada vez maior, a tecnologia possui uma grande influência quanto a expansão e conhecimento de práticas sociais. Dentre as diversas tecnologias que estão inseridas no cotidiano das pessoas no mundo atual, destaca- se a internet. Esta possui diversos serviços que são oferecidos, porém será dado foco a World Wide Web (WWW ou W3), mais conhecido como Web (W3C, 2025).

A Web possui um amplo registro de serviços e informações que são de completa importância no mundo atual, onde a informação é de grande valor. Dessa maneira, é indispensável que qualquer usuário ao utilizar algum sistema Web, consiga realizar todas as tarefas que necessite sem nenhuma dificuldade, mesmo que possua alguma limitação. Dessa forma, a Web se torna acessível ao ser capaz de dar suporte para que haja acesso e oportunidades iguais para indivíduos com diversas aptidões (W3C, 2025).

De acordo com o World Wide Web Consortium (W3C, 2022b), organização responsável por desenvolver os inúmeros padrões que se tem na internet, a definição de acessibilidade na Web é: “Acessibilidade na Web significa que sites, ferramentas e tecnologias são projetadas e desenvolvidas para que pessoas com deficiência possam usá-las. Mais especificamente, as pessoas podem:1)perceber, entender, navegar e interagir com a Web; 2)contribuir para a web”.

É importante salientar que a acessibilidade não é limitada a apenas pessoas com deficiência mental ou física, mas pode atingir outros grupos da sociedade, como pessoas idosas, com deficiência visual em alto grau, baixa visão, daltonismo, surdez, baixa audição ou qualquer outra limitação que possa vir a impedir de acessar o conteúdo que busca na Web, mostrando assim a importância de um ambiente acessível na internet (World Wide Web Consortium (W3C), 2022).

Através de um desenvolvimento de um sistema Web que contenha recursos de acessibilidade propõe-se a, não somente ter as pessoas como receptora de conteúdo e informações, mas que todas as pessoas que acessarem o mesmo, possam usufruir e colaborar com a Web (W3C, 2022b).

> "Quando se pensa em acessibilidade na web e seus benefícios, é natural associar acessibilidade com responsabilidade social, melhoria da imagem das empresas/instituições e a disponibilização democrática de serviços, produtos e informações para as pessoas com deficiência. As pessoas com deficiência são as maiores beneficiadas, pois, na falta da acessibilidade na web, podem ter seu acesso prejudicado ou até mesmo impedido pelas barreiras impostas pela inacessibilidade. Mas, quando os sítios são verdadeiramente acessíveis, as pessoas com deficiência podem usufruir de todas as informações e serviços disponíveis na web". (W3C, 2014).

Dessa maneira, para facilitar o desenvolvimento com recursos acessíveis, o W3C desenvolveu a *Web Accessibility Initiative*, que em tradução seria a Iniciativa para a Acessibilidade na Web, conhecida como WAI (1997), responsável por assegurar que se tenha uma elaboração de um conjunto de orientações e modelo técnicos, para serem utilizados no momento de concepção de um sistema com acessibilidade. Sendo assim, possibilitando que haja um recurso de padronização e documentação para os desenvolvedores implementarem em seu sistema.

Reinaldo Ferraz em seu livro "Acessibilidade na Web: Boas práticas para construir sites e aplicações" (FERRAZ, 2020), aborda sobre algumas deficiências e suas respectivas soluções, especificamente a tecnologia assistiva, sendo elas, para deficiência visual(cegueira e/ou baixa visão), deficiência auditiva, deficiência motora e para a deficiência cognitiva/-neurológica.

A acessibilidade na Web não deve ser contemplada apenas por questões legais, cumprindo de acordo com a Lei Nº 13.146, de 6 de julho de 2015 (BRASIL, 2015), que em seu artigo 63 afirma a obrigatoriedade a acessibilidade em sítios da internet, mas se tornar essencial no momento da concepção de qualquer sistema, seja Web ou não.

Com base na necessidade de se ter um ambiente disponível a todo o momento e que qualquer pessoa consiga acessar e usufruir das informações sem dificuldades, ou seja, tendo total acessibilidade, foi realizado o desenvolvimento de um sistema Web para o projeto "Parede da Memória", da escola PIO XII, que contém recursos acessíveis para uma maior e melhor interação usuário-sistema.

1.1 O problema e sua importância

A generalização de acesso a conteúdos que ficam disponíveis em sistemas Web torna-se cada vez mais essencial para qualquer pessoa, a despeito de possuir alguma restrição ou deficiência que possa impossibilitá-la. Muitos estudos apontam que, apesar de se ter leis federais abordando o tema acessibilidade na Web, ainda se tem muitos sites que não correspondem ao esperado para se ter um conteúdo acessível a todos os indivíduos. Essa falta de conscientização dos projetistas de desenvolvimento na Web no que diz respeito a acessibilidade e a não aplicação dos métodos e procedimentos devidos

para o desenvolvimento de aplicações acessíveis contribuem para uma sequela vultosa sobre a acessibilidade (FREIRE, 2008).

De acordo com o Censo(2010), realizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE), quase 25% da população brasileira possui algum nível de dificuldade em executar suas tarefas por possuírem algum tipo de deficiência. Essa porcentagem equivale a quase 46 milhões de brasileiros que não estariam recebendo e contribuindo com a disseminação e transparência do acesso à informação caso não ocorra a devida atenção e foco na acessibilidade da Web.

Segundo o PNAD,Pesquisa Nacional por Amostra de Domínios, realizada em 2014(IBGE, 2014) , cerca de 57% das pessoas brasileiras que possuem alguma deficiência utilizam a internet com frequência. Somado a isso, se tem um estudo realizado pelo Movimento Web para Todos em parceria com a BigData Corp que realizou testes em 16,89 milhões de sites brasileiros que estão ativos, buscando analisar à acessibilidade dos mesmo e chegaram ao resultado que menos de 1% desses sites possuem recursos considerados acessíveis para as pessoas com deficiência.

Analisando os resultados obtidos nas duas pesquisas citadas é possível observar a proporção da interferência que se pode ter ao não se preocupar com a acessibilidade no momento da elaboração do projeto de um sistema, fazendo com que os futuros usuários possam ter gravíssimas limitações ao utilizá-la, dessa forma, afetando diretamente a utilidade da internet, já que a mesma tem como um de seus principais objetivos transmitir e receber informações, funcionando como um meio de comunicação globalizado.

Sendo assim, para atender a demanda da escola PIO XII, foi realizado o desenvolvimento de um site que inclui recursos de acessibilidade visando atingir o público usuário da escola.

1.2 Objetivos

Nesta seção são abordados os objetivos do trabalho desenvolvido.

1.2.1 Objetivo Geral

O presente trabalho tem como objetivo principal objetivo o desenvolvimento de um sistema para o projeto “Parede da Memória” com o estudo e aplicação de técnicas de acessibilidade para cumprir com o requisito do cliente, a escola PIO XII. Este desenvolvimento integra-se ao projeto de extensão UFES PRO TI[1] e, através do sistema citado contribui para a disseminação das informações que foram coletadas através do projeto “Parede da Memória”. Tendo em mente que trata-se de um sistema Web aberto destinado

[1]Disponível em:<https://proti.alegre.ufes.br/index.html>

a uma escola, tem-se, como uma das prioridades, o acesso global a todos os interessados, sobretudo aos necessitados de acessibilidade, mostrando assim a importância de uma Web disponível em todo momento, situação ou condição.

Como mais uma demanda de extensão do projeto UFES PRO TI, objetiva-se também a ampliação da atuação na área de extensão universitária em Computação pela UFES, agregando mais um projeto ao seu portfólio em atendimento à comunidade em geral.

1.2.2 Objetivos Específicos

Além dos objetivos gerais e a conclusão do desenvolvimento web para a escola PIO XII, tem-se os seguintes objetivos específicos:

- Análise e planejamento do sistema;
- Estruturação do banco de dados do sistema;
- Implementação das funcionalidades do sistema;
- Estudo e aplicação de técnicas de acessibilidade, contribuindo para que a web seja mais acessível.
- Disponibilização das informações coletadas no Projeto "Parede da Memória" para todos os usuários;

1.3 Motivação

O desenvolvimento do sistema para a escola PIO XII faz parte integrante do projeto de extensão UFES PRO TI, o qual tem por objetivo anteder a comunidade em geral com a prestação de serviços na área de Computação, primordialmente no desenvolvimento e implantação de sistemas de informação que atendam às demandas internas administrativas.

Jorge Fernandes e Francisco Godinho (FERNANDES; GODINHO, 2003) , afirmaram que: "Para a maioria das pessoas, a tecnologia torna a vida mais fácil. Para uma pessoa com necessidades especiais, a tecnologia torna as coisas possíveis". Dessa maneira, analisando a importância de se preocupar com as práticas de acessibilidade no momento de concepção de um sistema, desejou-se apresentar um sistema Web para a escola PIO XII onde possibilitou se ter uma Web alcançável a todas as pessoas.

2 Revisão da Literatura

Neste capítulo serão apresentados os princípios teóricos que apoiaram o desenvolvimento deste trabalho.

2.1 Projeto "Parede da Memória"

O projeto "Parede da Memória" constitui um Ponto de Memória situado na Escola Estadual de Ensino Fundamental e Médio Pio XII, localizada no centro de São Mateus-ES. A iniciativa teve início no âmbito escolar, envolvendo diretamente os estudantes na pesquisa, registro e preservação de memórias significativas da comunidade local.

A proposta do projeto consiste em identificar e documentar memórias de indivíduos que desempenharam papéis relevantes na construção cotidiana da história de São Mateus, especialmente aqueles cujas histórias são pouco abordadas pelos registros oficiais ou materiais didáticos. Para isso, os alunos, atuando como Jovens Pesquisadores, realizam visitas aos membros da comunidade mateense, entrevistando-os e coletando suas memórias, experiências e relatos de vida. Dessa forma, o projeto estabelece um vínculo direto entre escola e comunidade, promovendo a valorização de histórias locais e fortalecendo a conexão dos estudantes com sua própria cidade.

Atualmente, a "Parede da Memória" apresenta 64 (sessenta e quatro) imagens, acompanhadas das memórias de seus doadores, fixadas diretamente em uma parede da escola, tornando o espaço físico um ponto de encontro entre memória, educação e comunidade. Essas imagens representam sessenta e quatro formas de contar outras histórias, ou seja, narrativas que muitas vezes não são contempladas pelos livros didáticos, monumentos ou placas históricas, permitindo a construção de múltiplas perspectivas sobre a história de São Mateus.

O projeto valoriza indivíduos que contribuíram de maneira significativa para a manutenção de tradições culturais, preservação do patrimônio histórico, prática artística, atividades agrícolas, mobilização social e resistência cotidiana. Muitos desses sujeitos já não se encontram mais entre nós, mas deixaram legados relevantes, que agora são preservados e homenageados na "Parede da Memória".

Portanto, a "Parede da Memória" cumpre uma função dupla: por um lado, atua como instrumento de registro e preservação das memórias coletivas; por outro, constitui uma homenagem à história e à resistência do povo mateense, reconhecendo e valorizando aqueles que contribuíram para a construção da identidade local. Desde sua criação, em 2021, o projeto tem envolvido os estudantes em todas as etapas, desde a pesquisa e visitação

da comunidade até a seleção, registro e fixação das memórias na parede, consolidando-se como uma estratégia pedagógica, cultural e social capaz de aproximar jovens e visitantes das múltiplas dimensões da história de São Mateus.

2.2 Acessibilidade na Web

> "Acessibilidade é o grau em que produtos, sistemas, serviços, ambientes e instalações podem ser utilizados por pessoas de uma população com a mais ampla gama de necessidades, características e capacidades de usuário para atingir objetivos identificados em contextos de uso identificados."(INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2021).

Com o grande avanço da Web, ela se tornou parte do dia a dia de grande parte da população e dessa forma a maioria dos dispositivos tecnológicos se tornaram onipresente, ao mesmo tempo o uso desse meio por pessoas com necessidades especiais vem provocando a preocupação e conhecimento acerca do desenvolvimento de métodos para a produção de uma Web mais acessível (WADDELL et al., 2003). Com a disseminação rápida de informações que acarretou a globalização, é necessário uma atenção aos projetos de desenvolvimento de sistemas para Web, que constantemente esbarram nos desafios de contemplar todos os usuários, independente de possuírem alguma restrição ou necessidade. Sendo assim, a acessibilidade é a essência que abrange os fundamentos para se solucionar esses desafios (FREIRE, 2008).

Em muitos países, as entidades de leis que regularizam a disponibilização de materiais acessíveis têm se preocupado e dado ênfase em iniciativas que auxiliem a produção de técnicas para desenvolvimento de sistemas que possam estar cada vez mais acessíveis, mas é importante salientar que, primeiramente, os desenvolvedores devem estar cientes sobre a importância de se preocupar com a acessibilidade em seus sistemas (FREIRE, 2008).

Segundo Lazar(2003), apesar de possuir um grande número de usuários com múltiplas demandas, o número de sites que se preocupam em implementar corretamente as técnicas de acessibilidade é muito baixo. Lazar(2003) diz: “Websites precisam ser acessíveis a todos os usuários, incluindo aqueles com deficiência”.

Dessa forma, nos próximos tópicos, serão introduzidos termos e tecnologias que contribuíram para o desenvolvimento de um sistema acessível.

2.2.1 Design Universal

O Design Universal aborda sobre a tecnologia que é desenvolvida, onde a mesma deve ser ajustável para abranger o mais variável nível de habilidade das pessoas, sem que

seja necessário sacrificar o custo, eficácia ou estética. Apesar de terem leis que definem a obrigatoriedade no desenvolvimento de tecnologias com Design Universal, grande parte do incentivo vem da recepção positiva das pessoas acerca das vantagens que o mesmo oferece, visto que o designer atende as necessidades de pessoas com deficiência (FREIRE, 2008).

O conceito implícito ao Design Universal é que ambientes e produtos precisam ser apropriados, diretamente, a um grande número de pessoas, independente se possuírem alguma deficiência. Quando esse tipo de ambiente não estiver à disposição, é necessário se adaptar para oferecer alternativas, como acessórios padronizados, utilização de tecnologias assistivas (PUPO; MELO; FERRÉS, 2006).

Dessa forma, fazer uso de um Design Universal proporciona uma potencialização e participação de todos em igualdade, ocasionando diretamente uma inclusão para todos (PUPO; MELO; FERRÉS, 2006).

2.2.2 Tecnologias Assistivas

Segundo o Comitê de Ajudas Técnicas (CAT), Tecnologia Assistiva pode ser definida como:

> "É uma área do conhecimento, de característica interdisciplinar, que engloba produtos, recursos, metodologias, estratégias, práticas e serviços que objetivam promover a funcionalidade, relacionada à atividade e participação de pessoas com deficiência, incapacidades ou mobilidade reduzida, visando sua autonomia, independência, qualidade de vida e inclusão social" (CAT, 2007).

Sendo assim, a expressão Tecnologia Assistiva é empregada para a identificação de todos os métodos e técnicas que auxiliam no fornecimento e ampliação das habilidades funcionais de todas as pessoas, contribuindo para uma maior inclusão social e a autonomia (FREIRE, 2008). O termo *Assistive Technology*, que em tradução é Tecnologia Assistiva, é determinado como uma coleção de serviços, dispositivos, táticas e padrões desenvolvidos e utilizados para aliviar os obstáculos encontrados por pessoas que possuem necessidades especiais (PRESTON, 2003).

> "Entende-se por tecnologia assistiva, conforme definições amplamente reconhecidas, o conjunto de produtos, dispositivos, equipamentos, recursos, metodologias e serviços que têm como objetivo ampliar a funcionalidade, a autonomia, a independência e a inclusão social de pessoas com deficiência ou mobilidade reduzida, favorecendo sua participação plena na sociedade."(INTERNATIONAL ORGANIZATION FOR STANDARDIZATION, 2021).

Ao colocar em prática o desenvolvimento e utilização de Tecnologias Assistivas é possível atender uma enorme demanda das pessoas que vai desde tarefas básicas até atividades no trabalho. Assim, é possível observar que dar atenção a utilização dessas tec-

nologias possibilita que as pessoas possam alcançar autonomia e independência, atingindo diretamente o objetivo que se tem de inclusão social (FREIRE, 2008).

Apesar de possuir diversas vantagens é muito importante que esteja atento a utilização e desenvolvimento correto das Tecnologias Assistivas em cada situação, tomando o devido cuidado para que a tecnologia não se torne uma barreira para as pessoas que a utilizam. Sendo assim, existem diversos sistemas e dispositivos que proporcionam o auxílio na independência dos indivíduos que a utilizam (FREIRE, 2008).

Dentre as Tecnologias Assistivas que estão disponíveis, Pupo et al(2006) traz algumas definições para melhor entendimento:

- Sistemas para entrada de voz: Essa tecnologia baseia-se em utilizar o computador por comando de voz, dessa maneira pessoas que possuem algum impeditivo na mobilidade, podem usufruir de programas do computador que suportem esse tipo de Tecnologia Assistiva;
- Dispositivos apontadores alternativos: Essa tecnologia baseia-se em alternativas ao mouse, onde o usuário pode acionar elementos através da seleção do conteúdo de uma interface gráfica;
- Teclados Alternativos: Essa tecnologia baseia-se em alternativas ao teclado, onde é oferecido o acionamento das teclas através de programas no computador ou dispositivos físicos;
- Ponteiras de cabeça: Essa tecnologia baseia-se em dispositivos presos à cabeça, auxiliando as pessoas que têm dificuldade de utilizar, por exemplo, um teclado de maneira convencional;
- Ampliadores de tela: Essa tecnologia baseia-se em softwares que possibilitam ampliar conteúdos que sejam apresentados na tela de um computador, auxiliando as pessoas que possuem baixa visão;
- Leitores de tela com síntese de voz: Essa tecnologia baseia-se em dispositivos que realizam a leitura da tela, do conteúdo textual.

Conhecendo algumas das Tecnologias Assistivas é possível observar o impacto que elas podem ter na vida das pessoas. Reinaldo Ferraz (2020) em seu livro "Acessibilidade na Web: Boas práticas para construir sites e aplicações", aborda sobre algumas deficiências e suas respectivas soluções, sendo elas:

- Deficiência Visual(cegueira): Leitores de tela, destacando-se:

  - JAWS [1]: Grátis somente para teste, funciona no Windows;
  - NVDA [2]: Totalmente grátis, funciona somente no Windows;
  - ORCA [3]: Totalmente grátis, funciona somente no Linux;
  - CHROMEVOX [4]: Totalmente grátis, funciona no Chromebook e também como plugin para o Google Chrome;
  - VOICEOVER [5]: Leitor de tela somente para IOs;
  - TALKBACK [6]: Totalmente grátis, leitor de tela somente para Android.

- Deficiência Visual(baixa visão): Recursos que ampliem e melhorem a leitura de uma página Web, destacando-se:
  - LENTEPRÓ [7]: Totalmente gratis, funciona somente no Windows;
  - MAGIC [8]: Pago, funciona somente no Windows;
  - HIGH CONTRAST [9]: Totalmente grátis, funciona como um plugin para o Google Chrome;
  - TEXT LEGIBILITY [10]: Totalmente grátis, funciona como um plugin para o Firefox.

- Deficiência auditiva: Recursos como transmissão por áudio, tradutores de LIBRAS e transcrição, destacando-se:
  - HANDTALK [11]: Tradutor de LIBRAS;
  - VLIBRAS [12]: Tradutor de LIBRAS;
  - WEBCAPTIONER [13]: Ferramenta de transcrição.

- Deficiência motora: Recursos controlados com a boca ou utilizados na testa, mouses e teclados adaptados, softwares que reconheçam voz ou movimento, destacando-se:
  - CONFIGURAÇÃO DE NAVEGAÇÃO DA APPLE: Permite que um usuário seja capaz de controlar o dispositivo (Iphone ou Ipad) através de movimentos simples de cabeça;

[1] https://www.tecassistiva.com.br/catalogo/jaws/
[2] https://www.nvaccess.org/
[3] https://help.gnome.org/users/orca
[4] https://chromewebstore.google.com/detail/screen-reader/kgejglhpjiefppelpmljglcjbhoiplfn?authuser=4&pli=1
[5] https://support.apple.com/pt-br/guide/iphone/iph3e2e415f/ios
[6] https://support.google.com/accessibility/android/answer/6007100?hl=pt-BR
[7] https://intervox.nce.ufrj.br/dosvox/dosvox.html
[8] https://support.freedomscientific.com/Products/Blindness/MAGic
[9] https://chromewebstore.google.com/detail/high-contrast/djcfdncoelnlbldjfhinnjlhdjlikmph?hl
[10] https://addons.mozilla.org/en-US/firefox/addon/text-legibility/
[11] https://www.handtalk.me/br/
[12] https://www.gov.br/governodigital/pt-br/acessibilidade-e-usuario/vlibras
[13] https://webcaptioner.online/

  - SIRI [14]: Permite a realização de ações no dispositivo(Iphone ou Ipad) através de comandos de voz, através de um assistente pessoal da Apple;
  - EVA FACIAL MOUSE [15]: concede o controle de uma ponteira do mouse através de movimentos da cabeça, apenas para dispositivos Android;
  - PESQUISA POR VOZ DO GOOGLE: Permite que se controle o dispositivo (Android) através de comandos de vozes, utilizando o "OK Google".

- Deficiência cognitiva/neurológica: Recursos que melhoram a experiência das pessoas, tornando a Web mais fácil e legível, destacando-se:
  - WEBHELP [16]: Um plugin do Google Chrome que disponibiliza itens para customizar a tela, tornando a leitura de uma pessoa com dislexia mais confortável;
  - DYSLEXIA FORMATTER [17]:Um plugin do Google Chrome que disponibiliza alterações em recursos do navegador para facilitar a navegação de pessoas que possuem dislexia;

Esses são alguns dos recursos que estão disponíveis atualmente no ramo tecnológico para facilitar a realização de tarefas por pessoas que tenham limitações.

2.2.3 Diretrizes de Acessibilidade na Web

Para auxiliar na interação que ocorre entre um humano e um computador é realizado o desenvolvimento de interfaces, que muitas vezes possuem diretrizes. A utilização destas corresponde à enumeração de sentidos e princípios para a resolução de adversidades que possam ser encontradas em problemas já conhecidos. Dessa forma, existem múltiplas diretrizes que auxiliam no desenvolvimento de sistemas cada vez mais acessíveis (FREIRE, 2008).

2.2.3.1 WCAG

Com o intuito de tornar os conteúdos que são disponibilizados na Web mais acessíveis, foi desenvolvido pela W3C as Diretrizes de Acessibilidade para Conteúdo Web (WCAG [18]). Ao seguir as informações que são disponibilizadas pelas diretrizes é possível disponibilizar o conteúdo da web de maneira mais abrangente, atingindo um maior número de usuários (W3C, 2018).

Existem, no mundo inteiro, diversos indivíduos e instituições utilizando a WCAG,

[14]https://www.apple.com/br/siri/
[15]https://orientatech.es/pt/eva-facial-mouse-pro/
[16]https://chrome-stats.com/d/pjnhjelpkdoihfjeeemmahpdbmgliboo
[17]https://dyslexiefont.com/en/
[18]https://www.w3.org/WAI/standards-guidelines/wcag/

de diversas áreas, desde programadores até professores. Para serem capazes de acomodar as necessidades deste grupo, a WCAG utiliza de níveis de orientação para conseguir ser capaz de atender a todos que carecerem. Essas camadas são listadas abaixo, de acordo com a WCAG (W3C, 2018):

- Princípios: Essa camada de orientação abrange quatro princípios que sustentam a acessibilidade na Web: operável, robusto, perceptível e compreensível;
- Diretrizes: Posteriormente se têm as diretrizes. Sendo constituídas de 13 diretrizes, elas disponibilizam propósitos básicos para que os autores possam atingir e desenvolver conteúdos mais acessíveis;
- Critérios de Sucesso - Complementando as diretrizes, se tem os critérios de sucesso, que contribuem para a definição dos requisitos e testes de conformidade;
- Técnicas de tipo Necessária e de tipo Sugerida - E por fim, cada critério de sucesso e cada diretriz contribuem para a formação de uma documentação, que fornece técnicas. Estas permitem diminuir as barreiras apresentadas pela falta de acessibilidade.

Cada parte da orientação citada acima é complementar, sendo assim, funcionam em conjunto e acabam por oferecer orientações para ser disponibilizado conteúdos mais acessíveis.

2.2.3.1.1 Principios da WCAG

Para se conseguir sucesso completo no desenvolvimento de um sistema acessível, é preciso cumprir com os quatros princípios da acessibilidade, sendo eles (CALDWELL et al., 2008):

- Perceptível - Os usuários que utilizarem o site devem ser capazes de perceberem e captarem as informações que são fornecidas através do site;
- Operável - Os usuários que utilizarem o site devem ser capazes de manejar a interface do sistema;
- Compreensível - Os usuários que utilizarem o site devem conseguir compreender a informação e o funcionamento do sistema;
- Robusto - Os usuários do sistema que utilizarem o site devem conseguir acessar todo o conteúdo do sistema, mesmo que as tecnologias avancem.

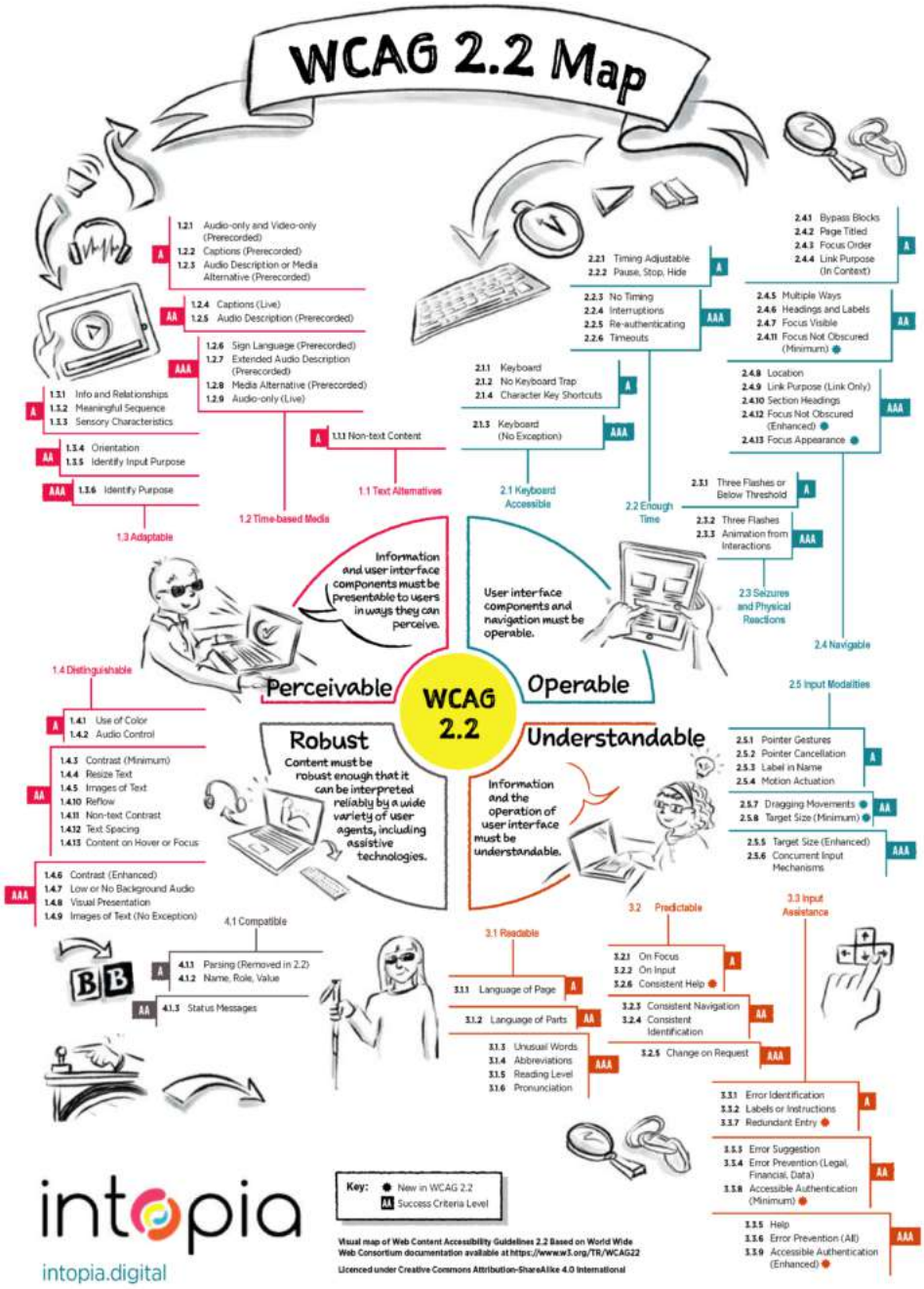

Figura 1 – Representação do WCAG 2.2

Fonte: (MORROW, 2023)

A Figura 1 representa os princípios de acessibilidade do WCAG 2.2, junto aos critérios de sucesso, sendo A (Essencial - o mínimo para se ter uma utilização por tecnologias assistivas), AA (Ideal - meta de acessibilidade) e AAA (Especializado - envolvem sites que possuem um público-alvo mais especializado).

A descrição mais detalhada com todos os critérios de sucesso podem ser encontradas no apêndice A.

2.2.3.2 WAI-ARIA

O Accessible Rich Internet Applications (ARIA[19]) foi desenvolvido pela W3C para conseguir tornar conteúdos dinâmicos e controle de interfaces a possuírem mais acessibilidade. Sem a utilização do WAI-ARIA, muitas das funcionalidades dos sites não estariam disponíveis para alguns usuários, principalmente os que fazem uso de leitores de tela e/ou não utilizam o mouse (W3C, 2006).

Dentre as características que o ARIA fornece, destaca-se (DPW, 2020):

- Melhoria na acessibilidade e interatividade com o teclado;
- Melhoria na definição de landmarks da página;
- Melhoria na definição de live regions atualizadas dinamicamente;
- Melhoria na acessibilidade de controles interativos.

2.2.3.3 E-MAG

Sites do Governo Brasileiro são extremamente importantes para transmissão e coleta de informações dentro do país, dessa forma o E-MAG[20] é o grupo de recomendações e diretrizes para uma padronização dentro desses sites. As sugestões abordadas neste modelo de diretriz, possibilitam uma concretização da acessibilidade na Web com uma fácil implementação, que atenda a necessidade do povo brasileiro e seguindo uma padronização semelhante aos padrões internacionais. É considerável evidenciar que o E-MAG possui características semelhantes ao WCAG, porém é voltado para o âmbito governamental brasileiro (EMAG, 2014).

2.2.4 Usabilidade

Para Nielsen(1993, apud (ROCHA, 2013)) a definição de usabilidade é: " a usabilidade é um atributo de qualidade que avalia quão fácil uma interface é de usar" e completa com "a medida de qualidade da experiência de um usuário ao interagir com um produto ou um sistema". Sendo assim, é possível compreender que durante o desenvolvimento de um sistema, no âmbito tecnológico, a usabilidade oferece meios para se ter um sistema que seja de fácil uso (LESINHOVSKI, 2014).

Se relacionando com outros elementos, a usabilidade é ligada aos seguintes fatores, segundo Nielsen(1993, apud (ROCHA, 2013)):

- Satisfação: Os usuários do sistema devem ficar satisfeitos ao utilizarem o sistema;

[19]https://www.w3.org/WAI/standards-guidelines/aria/
[20]https://emag.governoeletronico.gov.br/

- Facilidade de aprendizagem: Os usuários do sistema devem conseguir trabalhar rapidamente no sistema, ou seja, o sistema deve proporcionar fácil assimilação para os mesmos;
- Segurança: O sistema deve estar preparado para o tratamento de erros. Caso os usuários cometam algum erro, o sistema deve proporcionar a correção do mesmo e até mesmo a recuperação da versão anterior;
- Facilidade de memorização: Os usuários devem conseguir utilizar o sistema com praticidade e intuitividade, mesmo depois de passarem um longo tempo sem utilizar o mesmo, ou seja, os usuários devem ter facilidade em memorizar as funcionalidades do sistema;
- Eficiência: Os usuários devem ser capazes de ter uma boa produtividade na utilização do sistema.

2.3 Tecnologias Envolvidas

Nesta seção será abordado as tecnologias envolvidas no desenvolvimento do sistema.

2.3.1 HTML

A Linguagem de Marcação de Hipertexto, conhecida como HTML[21], é o bloco mais básico que se tem na Web, responsável pela publicação de conteúdos multimídias (MUSCIANO; KENNEDY; NOVAK, 1998). Desenvolvida por Tim-Berners-Lee, na Suíça, sua primeira versão foi disponibilizada em 1991 com o objetivo de criação de um sistema para comunicação com a internet, navegadores. Com a grande popularidade que o HTML ganhou, ela se tornou paradigma oficial da web e constitui grande parte dos aplicativos online e das páginas da internet (HOSTINGER, 2022c).

Um fato extremamente importante sobre o HTML é que o mesmo não é considerado uma linguagem de programação, mas uma linguagem de marcação. Sendo assim, o HTML é composto de partes próprias que formam páginas Web através de códigos que indicam características visuais da página. A cada nova atualização do HTML, o mesmo se torna mais fácil de utilizar e adquire mais funcionalidades (INFOESCOLA, 2022).

2.3.2 CSS

Após o desenvolvimento de sistemas Webs, tornou-se necessário um meio de colocar estilo nas mesmas, dessa forma surgiu o Cascading Style Sheets (CSS[22]), sendo um grande

[21] https://www.w3.org/html/
[22] https://www.w3.org/Style/CSS/Overview.en.html

avanço no âmbito de designer de páginas Web, ao ampliar as possibilidades para o visual de uma página Web (LIE; BOS, 1997). Sendo, muitas vezes, utilizado junto ao HTML, o CSS é utilizado para dar estilo às páginas da Web. Sendo desenvolvido pelo W3C, em 1996, para complementar o HTML na composição visual do site (HOSTINGER, 2022b).

O CSS oferece um enorme diferencial no momento de concepção de um site, principalmente por se tratar da estilização do visual do site. Dentre as vantagens que o mesmo oferece, pode-se citar que, depois da criação do CSS, foi possível fazer toda a estilização em um arquivo separado da marcação hipertexto, deixando assim que se tivesse arquivos HTML e CSS mais limpos, em questão de organização visual (HOSTINGER, 2022b).

2.3.3 BootStrap

Sendo desenvolvida pelo Twitter, por volta de 2010, o BootStrap[23] foi ganhando espaço até se tornar um dos mais populares Frameworks front-end e de projetos com o código aberto em todo mundo (BOOTSTRAP, 2022). Esse Framework mescla JavaScript e CSS para conseguir personalizar os componentes da página em HTML, indo muito além de apenas estilizar botões e/ou cores da página (SOUZA, 2019).

O Bootstrap é composto por um conjunto de sintaxes que atuam funções específicas, dessa forma, o famoso Framework possui 3 arquivos básicos, sendo: Bootstrap.css, Bootstrap.js e Glyphicons. Essas sintaxes específicas permitem que os desenvolvedores consigam construir sistemas rápidos, sem se preocuparem com funções adicionais e/ou comandos básicos (HOSTINGER, 2024).

O BootStrap é compartilhado através do site oficial e npm. Além disso, também é possível utilizar o mesmo através do próprio código-fonte ou da distribuição por meio do CDN do próprio BootStrap. O CDN possibilita que um sistema web consiga recuperar arquivos utilizados em servidores distribuídos, frequentemente (ALURA, 2021).

Dentre as funcionalidades do Bootstrap, pode-se citar (SOUZA, 2019):

- Layout Responsivo: Sendo uma das principais características do BootStrap, possibilita a adaptação do layout da página de acordo com o tamanho e/ou dispositivo utilizado.
- Biblioteca de Componentes: Possibilita que seja utilizado uma grande variedade de componentes para proporcionar uma melhor comunicação com o usuário e melhor interatividade.

[23] https://getbootstrap.com/

2.3.4 JavaScript

Sendo desenvolvida pela Netscape Communications Corporation e criada, inicialmente, com o nome de Mocha , a linguagem Javascript[24] foi lançada oficialmente em 1995. Em pouco tempo a linguagem foi ficando bastante popular e grande parte disso se devia ao fato da linguagem possuir um nome bem similar ao Java, fazendo com que muitos desenvolvedores pensassem terem alguma relação, mesmo que as duas linguagens não tenham nenhuma ligação (GRILLO; FORTES, 2008).

O JavaScript possibilita o desenvolvimento de pequenos programas que podem ser feitos de modo híbrido junto ao código do HTML. Todos os processos do JavaScript ocorrem no lado do cliente, sendo conhecido como Client-Server, dessa forma poupando o peso de transitar com os processos para o servidor em qualquer requisição feita. Sendo assim, os processos realizados no JavaScript dependem exclusivamente do processamento local (GRILLO; FORTES, 2008).

É possível notar que o JavaScript se tornou uma linguagem plena e influente, possuindo diversas características, onde pode-se destacar (GRILLO; FORTES, 2008):

- Possibilidade de desenvolvimento de listas associativas;
- Tipagem dinâmica;
- Similaridade com a linguagem C/C++;
- A linguagem JavaScript é multiparadigma.

Além disso, JavaScript é uma linguagem de script, ou seja, uma linguagem interpretada. Dessa forma, basta digitar o código-fonte e o interpretador lerá o código, executando as instruções, linha por linha, cada vez que o script for rodado. Com a utilização de scripts não é necessário a utilização e criação de arquivos estáticos, basta alterar o código e rodar novamente o script.Assim, é possível notar a agilidade que o JavaScript possibilita para alterações no programa, apesar de perder um pouco o desempenho , ainda se mostra uma linguagem com diversas vantagens (GRILLO; FORTES, 2008).

2.3.5 Angular

Angular[25] é um framework voltado para o desenvolvimento da interface de aplicações, utilizando HTML, CSS e, especialmente, JavaScript, desenvolvido pela equipe do Google. Entre os elementos que o compõe, ressalta-se (ALGAWORKS, 2018):

[24]https://262.ecma-international.org/
[25]https://angular.dev/

- Componentes;
- Modelos;
- Diretivas;
- Roteamento;
- Módulos;
- Serviços
- Injeção de dependências;
- Ferramentas de infraestrutura.

A produtividade aumenta, o desenvolvedor tem a oportunidade de dividir o código em partes, com o uso de componentes, módulos e outros recursos. Com esta forma de estrutura, existe um ambiente favorável para um desenvolvimento mais organizado (DUARTE, 2022).

2.3.6 Spring Boot

O Java Spring[26] Framework é uma estrutura de nível empresarial de código aberto bem conhecida para a construção de aplicativos de produção independentes que são executados na Java Virtual Machine (JVM) (IBM, 2022).

O Spring Boot é um projeto da Spring que tem como principal objetivo facilitar todo o processo de configuração e publicação de aplicações. Dessa forma, os projetos são executados sem complicações, consequentemente, mais rápidos(ALGAWORKS, 2017).

Com o Spring Boot conseguimos abstrair etapas de configuração, como exemplo (DUARTE, 2022):

- Servidores;
- Gerenciamento de dependências;
- Configurações de bibliotecas.

2.4 Arquitetura do Sistema

Apesar dos estudos sobre arquitetura do sistema terem ganhado foco somente nos anos 90 (MELO, 2015), essa é uma definição que tem grande impacto, caso não seja bem

[26]https://spring.io/projects/spring-boot

planejada. No momento de concepção de um sistema é importante que se tenha um cuidado no preparo para o desenvolvimento do mesmo. Dessa forma, planejar e se preocupar com a arquitetura do sistema é fundamental para amenizar as complicações que se tem no desenvolvimento de um sistema.

A arquitetura de software pode ser compreendida como um grupo de componentes de um sistema que contém uma grande relevância para a representação da estrutura e da forma de toda a aplicação. Sendo assim, um componente é a representação abstrata de um item do sistema. Na arquitetura de software isso pode ser apresentado em, por exemplo, uma implementação para um software, onde os componentes seriam os relacionamentos e suas propriedades (PRESSMAN, 2006).

Segundo (BASS; CLEMENTS; KAZMAN, 2003), pode-se pensar em três vantagens que envolvem o cuidado com a arquitetura de software, sendo:

- Uma boa comunicação com os stakeholders visto que a arquitetura representa o sistema em um alto nível, facilitando o entendimento de pessoas que não sejam da área;
- Ter uma análise de sistema pois para tornar toda a arquitetura do sistema evidente na sua fase inicial é necessário que se tenha um estudo. Dessa forma, as decisões são mais cuidadosas, chegando a conclusões sobre as funcionalidades que o sistema vai oferecer e se o mesmo atende aos requisitos críticos;
- Ao se desenvolver um modelo de arquitetura de software se terá toda uma documentação detalhada sobre o mesmo. Dessa forma, caso seja desenvolvido um outro sistema que possua requisitos semelhantes, grande parte da documentação e arquitetura poderá ser reutilizada.

Dessa maneira, o planejamento que a arquitetura de software oferece se mostra valioso pois interfere no desempenho, manutenção, distribuição e na robustez de um sistema (BOSCH, 2000).

2.4.1 API

Uma *Application Programming Interface* (API) consiste em uma Interface de Programação de Aplicativos que é capaz de fazer a comunicação entre as partes de um sistema, consequentemente, facilitando a comunicação do desenvolvedor e do sistema. A empresa Red Hat(2017) , responsável por oferecer soluções de sistemas operacionais, define API como "API significa interface de programação de aplicações, um conjunto de definições e protocolos para criar e integrar softwares de aplicações".

Com o desenvolvimento de uma API, o sistema consegue estabelecer uma comuni-

cação com outros serviços, conforme a Figura 2, sem a necessidade de se saber como foi elaborada a implementação. Dessa maneira, ocorre uma descomplicação no momento do desenvolvimento, permitindo que o sistema seja implementado de forma privada. Portanto, as APIs proporcionam a integração entre tecnolgoias e arquiteturas já existentes (HAT, 2017).

Na Figura 2 são representados os componentes e o fluxo da API. A API é o componente intermediário entre o back-end e os demais aplicativos, fazendo com que tenham uma boa troca de informações e consequentemente o sistema tenha uma boa integração.

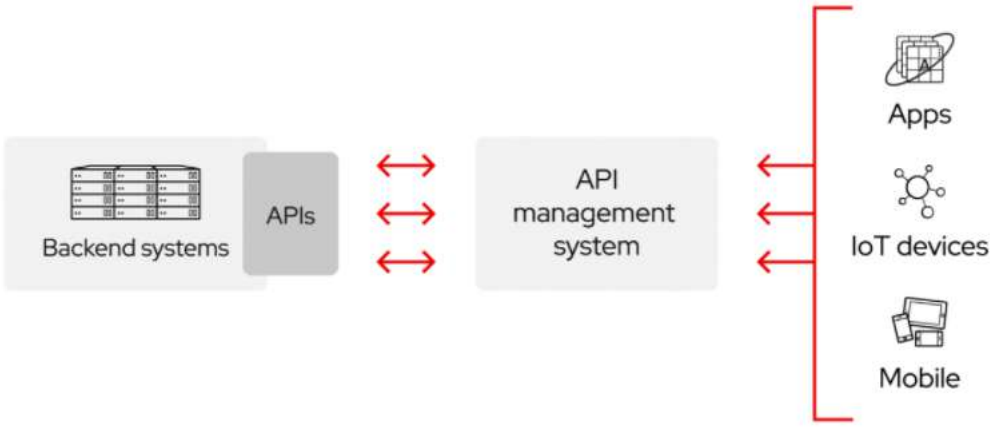

Figura 2 – Representação de uma API

Fonte: Red Hat

A utilização de APIs no desenvolvimento contribui muito para a facilidade no trabalho dos desenvolvedores e a padronização no momento de criar novos sistemas, visto que não é mais necessário desenvolver códigos personalizados. E é claro que as APIs também oferecem um grande suporte em relação a segurança, tendo a capacidade de monitorar acessos e permissões (TECHTUDO, 2020).

2.4.2 REST

Roy Fielding (2000) desenvolveu a *Representation State Transfer* (REST), em português sendo traduzido como transferência de estado representacional, que é definido como uma arquitetura para sistemas embasados em rede que une diversos outros estilos de arquitetura, aumentando assim as características do mesmo.

O REST, também conhecido como RESTful, funciona como uma coleção de limitações que a arquitetura necessita possuir, ou seja, REST não é nenhum padrão ou protocolo, mas pode ser utilizado de diversas maneiras no momento de criação de uma API (HAT, 2020). Sendo assim, o REST é capaz de fazer com que as requisições que são

solicitadas no sistemas Web, feitas através do protocolo HTTP, possam ser realizadas com mais simplicidade e padronizando os dados que são transferidos.

Segundo DEVMEDIA (2016) , um dos fundamentos do REST é que o sistema desenvolvido fornecerá uma *url* base e os verbos HTTP que serão responsáveis por apontar a ação que está sendo solicitada. Alguns dos verbos mais utilizados são:

- GET: Utilizado quando se deseja obter uma lista de dados, registros;
- POST: Utilizado quando se deseja um novo dado, registro;
- DELETE: Utilizado quando se deseja deletar um dado, registro;
- PUT: Utilizado quando se deseja editar um dado, registro;
- PATCH: Utilizado quando se deseja editar um dado, registro. Porém só são enviados os atributos que foram alterados.

As requisições realizadas podem ser entregues em diversos formatos, como JSON, HTML, XLT, entre outros. Porém o mais utilizado é o JSON, por ser independente da linguagem e de fácil compreensão para ser lido por humanos. Além disso, uma API deve cumprir com determinados parâmetros para ser considerada da categoria REST, sendo (HAT, 2020):

- Cliente-Servidor: É necessário ter uma arquitetura cliente/servidor que possui seus pedidos administrados pelo HTTP;
- *Stateless*: Não guardar informações do cliente entre os pedidos GET, além disso todos os pedidos são separados e desconectados;
- Cache: Possuir cache para conseguir otimizar as trocas de dados entre o servidor e cliente;
- Interface Uniforme: Possuir uma interface padrão para que todos os dados manipulados e trocados estejam em um formato uniforme;
- Layered: Possui organização entre os tipos de servidores, ou seja, possui um sistema em camadas.

2.4.3 Modelo da arquitetura do sistema

O modelo arquitetural MVC (*Model, View e Controller*) é amplamente utilizado no desenvolvimento de sistemas, especialmente aplicações web, por proporcionar uma clara separação de responsabilidades entre suas camadas. Nesse modelo, a camada *Model* é

responsável pela representação dos dados e regras de negócio; a camada *View* concentra-se na interface e na apresentação das informações ao usuário; e a camada *Controller* atua como intermediária, recebendo as requisições, processando-as e direcionando as interações entre o *Model* e a *View*. Essa divisão favorece a modularidade, reduz o acoplamento entre os componentes e torna o sistema mais flexível a mudanças (ARAUJO, 2024).

Com a evolução das práticas de engenharia de software, o padrão MVC passou a ser frequentemente estendido com a adição de novas camadas, como *Service*, *Repository* e *DTO*. Essas camadas complementares reforçam a organização do sistema, permitindo uma maior escalabilidade e facilitando a manutenção. O *Service* centraliza a lógica de negócio, o *Repository* abstrai o acesso ao banco de dados e o *DTO* (*Data Transfer Object*) otimiza o transporte de informações entre as diferentes partes da aplicação. Esse refinamento do modelo tradicional amplia suas vantagens, tornando-o uma referência sólida para arquiteturas modernas (ARAUJO, 2024).

Outro aspecto bastante comum nas arquiteturas atuais é a separação entre *frontend* e *backend*, muitas vezes associada à aplicação do modelo MVC. O *frontend* corresponde à camada de apresentação (*View*), sendo responsável pela interação com o usuário e pela exibição das informações. Já o *backend*, composto principalmente pelas camadas de *Model* e *Controller*, concentra-se na lógica de negócio, no processamento das requisições e na comunicação com o banco de dados.

Com base na Figura 3 pode-se compreender como funciona a comunicação do sistema. O frontend envia uma requisição para o backend, este que solicita as informações ao banco de dados, após a consulta a resposta retorna para o frontend, ficando disponível para o usuário.

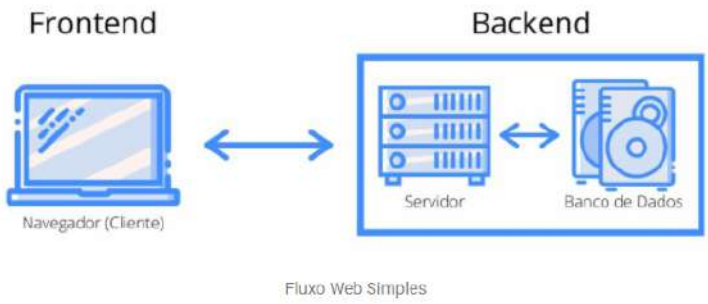

Figura 3 – Desenho da Arquitetura

Fonte: (FERNANDES, 2020)

2.5 Metodologia Scrum

O SCRUM se mostra essencial no controle de processos e possui em seus objetivos o aperfeiçoamento em previsibilidades, além disso contribui para conter os riscos que se tem durante o desenvolvimento de um projeto. Esta metodologia contém três bases que fazem parte de sua definição, sendo elas (SILVA; SOUZA; CAMARGO, 2013):

- Transparência: Todas as etapas do processo precisam ser claras para todas as partes comprometidas no projeto;
- Inspeção: Durante todo o processo do projeto é primordial que se tenha uma inspeção para que se evite conflitos;
- Adaptação: O projeto deve sempre estar preparado para mudanças e dessa forma ser capaz de se adaptar a elas.

Um aspecto importante da metodologia SCRUM é o desenvolvimento incremental, ou seja, cada etapa e processo do sistema é dividido e o projeto é construído de maneira a ser formado gradativamente. A divisão desses processos ocorre em sprints, que são os períodos de trabalho que se tem em cada fase da incrementação (SILVA; SOUZA; CAMARGO, 2013).

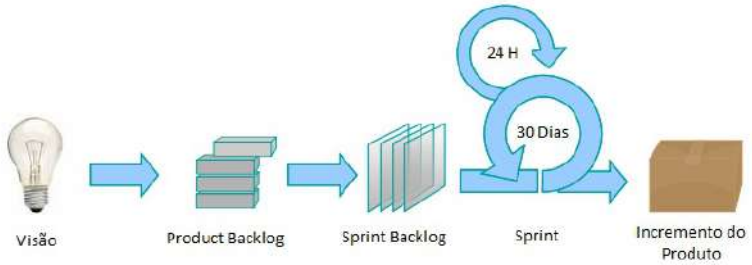

Figura 4 – Ciclo de Vida do SCRUM

Fonte: (HATANAKA, 2013)

Na Figura 4 é apresentado que a cada nova sprint é realizada uma nova implementação no projeto, sendo que cada implementação passa por todos os processos de codificação e testes (SILVA; SOUZA; CAMARGO, 2013).

3 Metodologia

O método de pesquisa utilizado no trabalho foi de âmbito exploratório e qualitativo, onde com base na análise do ambiente e o levantamento dos requisitos gerais foi possível o desenvolvimento de um sistema Web que atendesse as necessidades da escola PIO XII e ficasse disponível para acesso e interação de todos os usuários, usurfluindo das metodologias de desenvolvimento de software para a elaboração do sistema.

Analisando os problemas, foram coletados os principais requisitos para o desenvolvimento do sistema com entrevistas realizadas com as professoras Ana Carolina de Melo Loureiro e Prisciliana Costa Ventura, responsáveis pelo projeto “Parede da Memória”, além disso foi realizado um estudo na área de acessibilidade na Web para garantir o amplo acesso de todos os usuários.

Dessa maneira, o referido trabalho pode ser catalogado como uma pesquisa exploratória aplicada, quanto à natureza. Da perspectiva técnica esse trabalho pode ser categorizado como um estudo de caso.

No momento do desenvolvimento de um sistema é importante se preocupar com cada etapa, fazendo com que o produto final esteja completo e satisfaça o usuário que irá utilizá-lo.

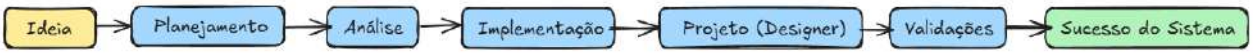

Figura 5 – Desenvolvimento de um sistema - Ciclo de Vida

Fonte: (DENNIS; WIXOM; ROTH, 2000)

Conforme a Figura 5, o projeto foi desenvolvimento seguindo o ciclo apresentando, onde, de acordo com a ideia inicial apresentada pelas professoras do projeto "Parede da Memória", foi realizado um planejamento, analisando as necessidades do projeto para um melhor levantamento dos requisitos. Após essa etapa, foi iniciada o desenvolvimento do sistema proposto, com atenção no designer e funcionalidade, para que atendesse as necessidades solicitadas pelas professoras do projeto, mas também os critérios de aprovação para um sistema com recursos de acessibilidade.

Seguindo diretrizes da WCAG foi possível realizar validações que garantissem a acessibilidade do sistema. Além disso, validações dos requisitos solicitados foram realizados pelas professoras do projeto, levando assim ao sucesso do sistema.

Cada etapa é fundamental e para contribuir em um bom planejamento, um projetista

de software pode contar com as metodologias de desenvolvimento de software, que são um grupo de técnicas que podem ser aplicadas no momento de criação de sistemas que processam dados. A escolha de uma metodologia adequada para o sistema, afeta diretamente no sucesso do projeto (MONITORA, 2020).

Dessa maneira, foi utilizada a metodologia de desenvolvimento que é classificada como ágil, conhecida como SCRUM, esta metodologia se sobressai entre as outras por conter um suporte para o desenvolvimento de aplicações que não contém requisitos estáveis e podem mudar a qualquer momento (SILVA; SOUZA; CAMARGO, 2013).

Durante todo processo de desenvolvimento, foram realizadas sprints com as funcionalidades a serem implementadas. Cada sprint realizada contou com a realização de testes, principalmente voltados para acessibilidade e, após a finalização completa da funcionalidade, era realizado a apresentação as partes interessadas.

Para dar suporte técnico à metodologia ágil e criar um sistema que fosse bem organizado e fácil de manter, foi escolhida a arquitetura REST. Essa opção foi ideal porque se encaixa perfeitamente no modo de trabalho do SCRUM, permitindo que cada parte do sistema (chamada de endpoint) fosse criada e testada separadamente a cada ciclo de desenvolvimento (sprint). Isso facilitou muito mostrar as novidades para as professoras a cada etapa e garantir que tudo se encaixasse perfeitamente no final.

3.1 Análise dos Resquisitos

Segundo Sommerville (2011) a análise dos requisitos consiste no processo onde os engenheiros de software junto aos usuários e clientes estabelecem informações sobre a aplicação, chegando assim aos requisitos que a mesma deve suportar, desde o desempenho até restrições de hardware. A Figura 6 demonstra uma visão em espiral deste processo:

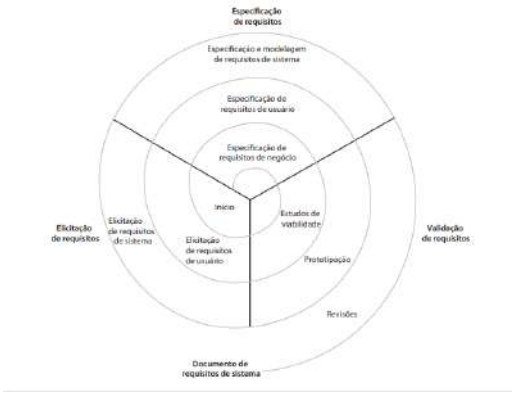

Figura 6 – Processo de engenharia de requisitos

Fonte: (SOMMERVILLE, 2011)

Dessa forma, a Figura 6 mostra que cada requisito novo é passado por todas as fases mostradas na Figura, para que no final se tenha uma maior certeza na definição do requisito. Sendo assim é possível se ter um feedback contínuo e um maior conhecimento a cada novo ciclo iniciado. O processo conta com as seguintes atividades (SOMMERVILLE, 2011): Descoberta de requisitos, Classificação e organização de requisitos, priorização e negociação de requisitos, e Especificação de requisitos.

A aplicação prática das técnicas recomendadas mostrou-se fundamental para que o site alcançasse um nível aceitável de acessibilidade, ao mesmo tempo em que manteve todos os requisitos funcionais e de layout solicitados pelo cliente.

Com base nas atividades descritas anteriormente, foi possível levantar de forma estruturada todos os requisitos do sistema. Durante esse processo, manteve-se um contato constante com as partes interessadas, garantindo um fluxo contínuo de feedback ao longo de cada etapa. Sempre que surgiam dúvidas ou pontos de incerteza, buscava-se inicialmente o esclarecimento por meio de mensagens; caso a questão demandasse maior aprofundamento, eram agendadas reuniões específicas para tratar do assunto. Essa proximidade entre a equipe e os stakeholders proporcionou maior clareza, alinhamento e segurança na definição dos requisitos, além de enriquecer a experiência no processo de engenharia de requisitos.

3.1.1 Descrição dos usuários

Em uma modelagem de sistema, os usuários são representados e vistos como atores. Sendo estes extremamente fundamentais em um caso de uso pois representam as formas que um usuário pode estabelecer uma interação com o sistema, uma funcionalidade que o sistema pode vir a ter.

No sistema desenvolvido, foi introduzido um ator Administrador Master que é encarregado de ter o papel de administrar o sistema, um Coadministrador que possue o papel de também administrar o sistema porém com acessos limitados e um ator Usuário Público que exerce o papel de todos os usuários que poderão acessar o sistema.

No Quadro 1 abaixo é possível ter a visualização de todos os atores do sistema e suas, respectivas, descrições.

Tipo	Descrição
Administrador Master	Representa o usuário que administra todas as funcionalidades do sistema
Coadministrador	Representa o usuário que administra somente algumas funcionalidades do sistema
Usuário Público	Representa o usuário que tem acesso ao sistema

Quadro 1 – Usuários do Sistema.

Fonte: Do Autor

3.1.2 Requisitos Funcionais

Segundo Sommerville (2011) os requisitos funcionais envolvem as declarações que um sistema pretende oferecer, quais serão suas reações e comportamentos ao interagir com entradas e situações específicas. Além disso, podem ocorrer situações em que os requisitos funcionais podem explicar o que um sistema não deve fazer.

Com base nisso, os requisitos funcionais do sistema são descritos abaixo:

RF01 — Cadastro de Coadministradores: O administrador master poderá cadastrar usuários do tipo coadministrador no sistema.

RF02 — Gerenciamento de Coadministradores: O administrador master poderá visualizar, atualizar e excluir usuários do tipo coadministrador.

RF03 — Login de Usuários Administrativos: O administrador master e os coadministradores poderão acessar o sistema por meio de autenticação (login).

RF04 — Cadastro de Subprojetos: O administrador master poderá cadastrar subprojetos no sistema.

RF05 — Gerenciamento de Subprojetos: O administrador master poderá atualizar e excluir subprojetos cadastrados.

RF06 — Visualização de Subprojetos: Usuários públicos poderão visualizar os subprojetos cadastrados no sistema.

RF07 — Cadastro de Publicações: O administrador master e os coadministradores poderão cadastrar publicações vinculadas a subprojetos.

RF08 — Gerenciamento de Publicações: O administrador master poderá atualizar e excluir publicações cadastradas.

RF09 — Visualização de Publicações: Usuários públicos poderão visualizar as publicações disponíveis no sistema.

RF10 — Cadastro de Colaboradores: O administrador master poderá cadastrar colaboradores que participam do projeto como bolsistas.

RF11 — Gerenciamento de Colaboradores: O administrador master poderá atualizar e excluir colaboradores cadastrados no sistema.

RF12 — Visualização de Colaboradores: Usuários públicos poderão visualizar os colaboradores do projeto.

RF13 — Cadastro de Informações de Contato: O administrador master poderá cadastrar informações de contato, como telefone, e-mail e redes sociais.

RF14 — Gerenciamento de Informações de Contato: O administrador master poderá atualizar e excluir as informações de contato cadastradas.

RF15 — Visualização de Informações de Contato: Usuários públicos poderão visualizar as informações de contato disponíveis no sistema.

RF16 — Solicitação de Agendamento de Visita: Usuários públicos poderão preencher e enviar um formulário para solicitar agendamento de visita.

RF17 — Gerenciamento de Agendamentos de Visita: O administrador master poderá visualizar, aprovar, reagendar ou excluir solicitações de visita.

RF18 — Envio Automático de E-mail ao Solicitar Visita: Ao receber uma solicitação de visita, o sistema deverá enviar automaticamente um e-mail de notificação.

3.1.3 Requisitos não funcionais

Segundo Sommerville (2011) os requisitos não funcionais envolvem as restrições às funções ou serviços que são ofertados pelo sistema. Estão inclusas as restrições no processo de desenvolvimento, restrições de timing e restrições impostas por normas.

Com base nisso, os requisitos não funcionais do sistema são descritos abaixo:

RNF01 – O sistema deverá atender às diretrizes de acessibilidade estabelecidas pela W3C (WCAG 2.2), garantindo recursos como contraste adequado, textos alternativos para imagens, navegação por teclado, descrição em links e formulários acessíveis, de modo a possibilitar o uso por pessoas com diferentes tipos de deficiência (visual, auditiva, motora e cognitiva).

RNF02 – O sistema deverá ser desenvolvido em plataforma web, disponível na internet, com acesso por meio dos principais navegadores.

RNF03 – O sistema deverá ter uma interface simples, intuitiva, de fácil entendimento e usabilidade para o usuário.

RNF04 – O sistema deverá ser responsivo.

RNF05 - Todas as fotos cadastradas deverão ter as informações: nome e localidade, pois serão utilizada no filtro da listagem.

RNF06 – O sistema terá o seguinte comportamento para os usuários do sistema:

- Administrador master: responsável por gerenciar o sistema sem limite de acessos, pode realizar o cadastro de usuários, subprojetos, colaboradores, informações de contato e publicações. Além dos cadastros, possue acesso de editar e excluir os registros.
- Coadministrador: responsável por gerenciar o sistema com limite de acessos, pode realizar o cadastro de publicações dentro do sistema.

- Usuário público: pode visualizar as informações de contato, os subprojetos e suas, respectivas, publicações e lista de colaboradores. Além disso, pode realizar o envio de um formulário solicitando uma visita a estrutura física do projeto.

3.1.4 Casos de Uso

A definição mais simples de um caso de uso é um diagrama que detecta quais são os atores e suas respectivas interações com o sistema, dando nome para o tipo de interação. Dessa maneira, os casos de uso podem ser utilizados para se conseguir detectar quais os requisitos do sistema, sendo uma das características básicas da linguagem de modelagem unificada, mais conhecida como UML (SOMMERVILLE, 2011). A Figura 7 apresenta, de forma diagramada, todos os requisitos do sistema por meio de um diagrama de casos de uso.

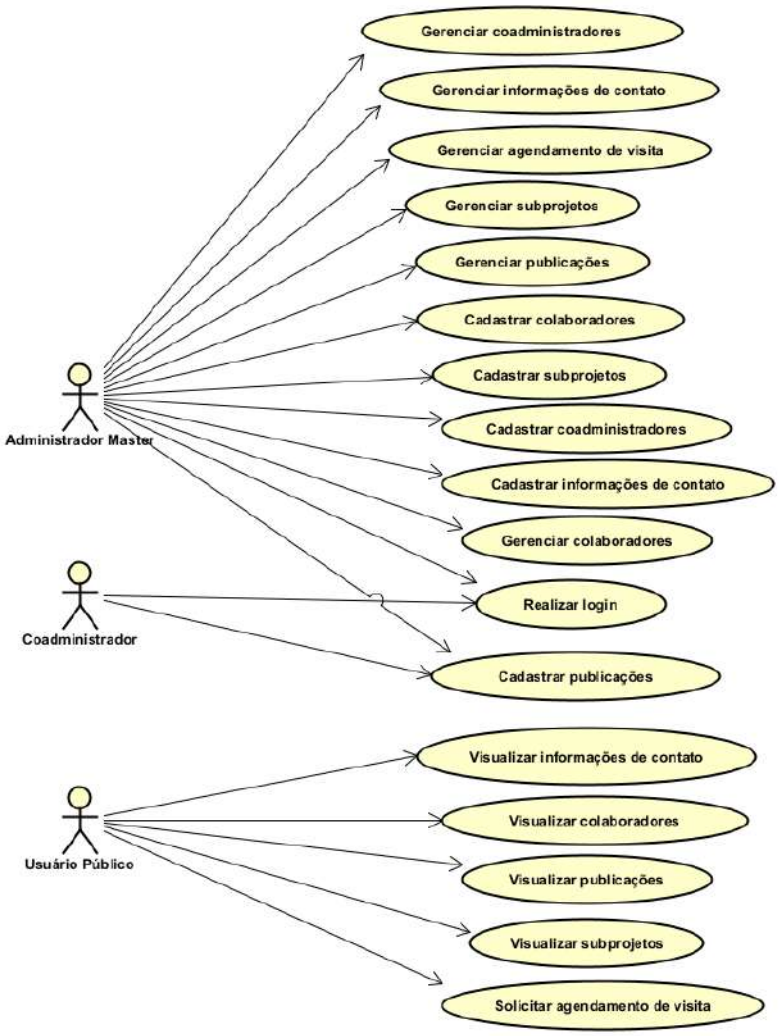

Figura 7 – Diagrama de Caso de Uso

Fonte: Do Autor

3.1.4.1 Narrativas de Caso de Uso

Somente a representação de diagramas de caso de uso resulta em pontos de atenção pois leva a uma falta de explicação. Dessa forma, para facilitar o entendimento é realizado o desenvolvimento de descrições que complementam o diagrama de casos de uso, conhecidas como narrativas de caso de uso (FILHO, 2011).

Uma narrativa é um exemplo sólido de um caso de uso em ação, mas deve-se atentar que uma narrativa não é o requisito completo, mas uma descrição mais generalizada e detalhada dos requisitos (COCKBURN, 2005).

A seguir, uma narrativa de caso de uso 001 para gerenciar subProjetos dentro do sistema. Demais narrativas dos casos de uso do sistema, encontram-se no Apêndice B.

UC001 - Gerenciar SubProjetos

Descrição: Este caso de uso descreve como o administrador master e o usuário público interagem com o sistema para gerenciar e visualizar subprojetos.

Atores: Administrador Master e Usuário Público.

Pré-condições:O administrador master deve estar logado no sistema.

Pós-condições: Um subprojeto é criado, editado, excluído ou exibido conforme a ação do ator.

Fluxo Normal (Administrador Master)

Criar Subprojeto

1. O sistema exibe a tela de gerenciamento de subprojetos.
2. O administrador master seleciona a ação desejada: criar, editar ou excluir subprojetos.
3. O sistema solicita o nome do subprojeto.
4. O sistema solicita o ícone representativo do subprojeto.
5. O sistema solicita a descrição do subprojeto.
6. O sistema solicita o resumo do subprojeto.
7. O administrador master clica no botão de salvar subprojeto.
8. O sistema salva as informações do novo subprojeto no banco de dados.
9. O sistema confirma a operação com uma mensagem de sucesso.

Editar Subprojeto

1. O sistema exibe a tela de gerenciamento de subprojetos.

2. O administrador master seleciona a ação desejada: criar, editar ou excluir subprojetos.
3. O sistema exibe a lista de subprojetos disponíveis.
4. O administrador master seleciona um subprojeto para editar.
5. O sistema exibe as informações atuais do subprojeto.
6. O administrador master altera os campos desejados (nome, ícone, descrição ou resumo).
7. O administrador master clica no botão de salvar alterações.
8. O sistema atualiza as informações do subprojeto no banco de dados.
9. O sistema confirma a operação com uma mensagem de sucesso.

Excluir Subprojeto

1. O sistema exibe a tela de gerenciamento de subprojetos.
2. O administrador master seleciona a ação desejada: criar, editar ou excluir subprojetos.
3. O sistema exibe a lista de subprojetos disponíveis.
4. O administrador master seleciona um subprojeto para excluir.
5. O sistema exibe uma mensagem de confirmação.
6. O administrador master confirma a exclusão.
7. O sistema remove o subprojeto do banco de dados.
8. O sistema confirma a operação com uma mensagem de sucesso.

Fluxo Normal (Usuário Público)

1. O sistema exibe a lista de subprojetos disponíveis e suas respectivas, informações de: nome, ícone e resumo.
2. O usuário público seleciona um subprojeto para visualizar.
3. O sistema exibe os detalhes do subprojeto: nome, descrição completa e publicações associadas.

Fluxos Alternativos

FA01 – Campo obrigatório não preenchido ao criar subprojeto.

1. O sistema exibe a mensagem MSG001.
2. O administrador master preenche o campo obrigatório.
3. Retorna ao fluxo normal no passo 07.

FA02 – Campo obrigatório não preenchido ao editar subprojeto.

1. O sistema exibe a mensagem MSG001.
2. O administrador master preenche o campo obrigatório.
3. Retorna ao fluxo normal no passo 07.

Fluxos de Exceção

FE01 – Falha ao salvar subprojeto no banco de dados.

1. O sistema exibe a mensagem MSG002.
2. O sistema encerra a operação.

FE02 – Falha ao salvar alterações no banco de dados.

1. O sistema exibe a mensagem MSG002.
2. O sistema encerra a operação.

FE03 – Falha ao excluir subprojeto do banco de dados.

1. O sistema exibe a mensagem MSG003.
2. O sistema encerra a operação.

Mensagens do Sistema:

- MSG001: "Preencha o campo obrigatório."
- MSG002: "Erro ao salvar subprojeto. Tente novamente!"
- MSG003: "Erro ao excluir subprojeto. Tente novamente!"

3.1.5 Diagrama de Classe

Durante o desenvolvimento de algum modelo de sistema que seja orientado a objetos, no estágio inicial, são utilizados diagramas de classes para representar o mundo

real. Assim, os "objetos" são capazes de espelhar pessoas e ações do mundo dentro do sistema. É importante salientar que os diagramas de classe em UML podem ser criados em diferentes graus de descrição (SOMMERVILLE, 2011).

Na maioria dos casos, o diagrama de classe é mais utilizado e sua forma de representação é em forma de classe, onde cada objeto tem sua respectiva classe, contendo também atributos, que correspondem aos dados que um objeto possui, e suas operações, que equivale às ações que um objeto pode realizar. Cada objeto pode se relacionar com os outros, formando assim as diversas relações que o diagrama proporciona (BEZERRA, 2007).

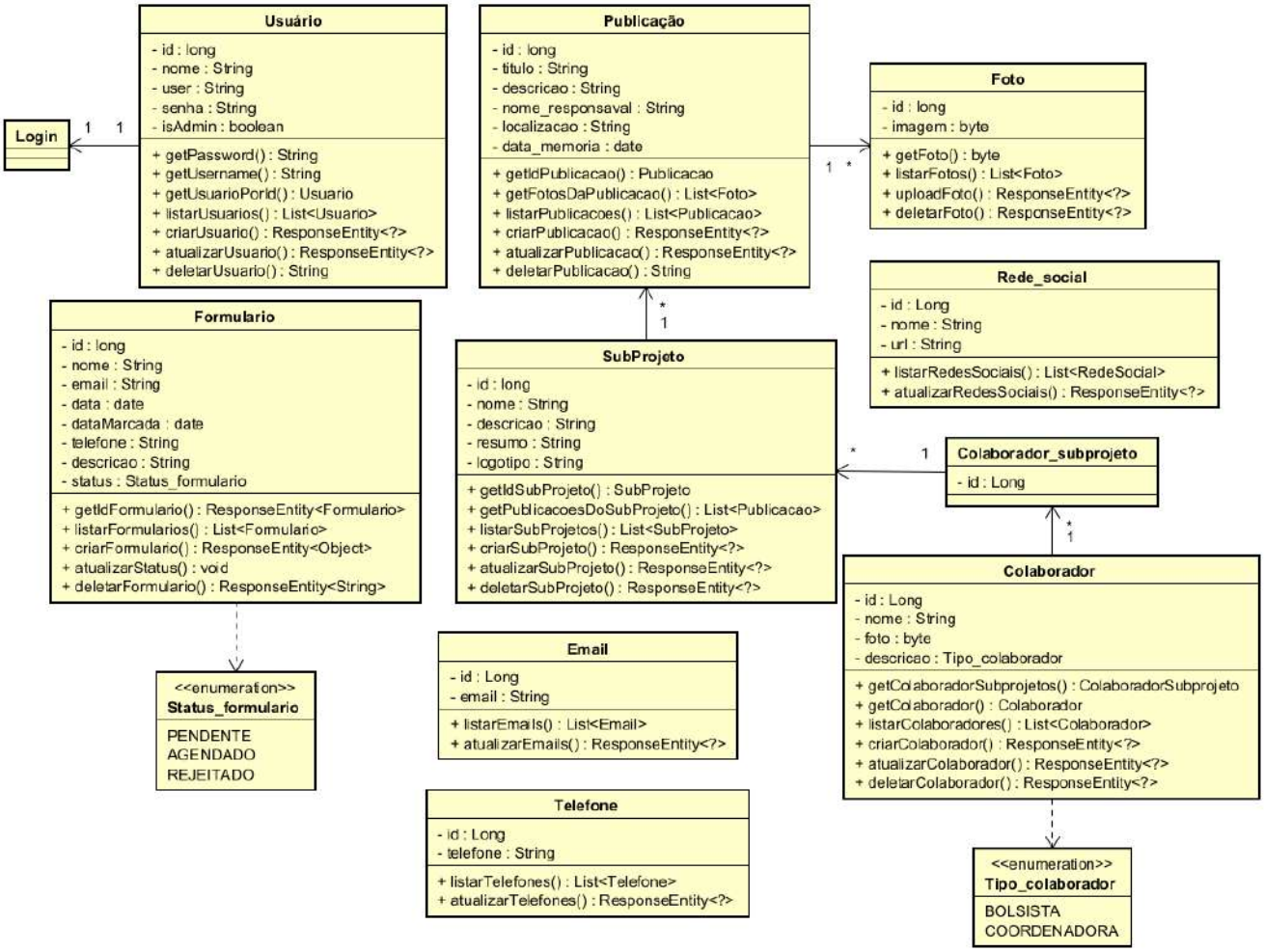

Figura 8 – Diagrama de Classe

Fonte: Do Autor

O diagrama de classes do sistema (Figura 8) apresenta a estrutura da aplicação, evidenciando as entidades, seus atributos, métodos e relacionamentos. Entre as principais classes destacam-se: Usuário, responsável pelo controle de acesso ao sistema; Formulário, que gerencia as solicitações recebidas de visita à escola PIO XII; Publicação e Foto,

que representam o acervo de memória e seus respectivos registros visuais; SubProjeto, que contém todos os projetos cadastrados no sistema; e Colaborador, que gerencia os participantes dos projetos.

Além disso, são utilizados tipos enumerados como Status_formulario e Tipo_colaborador para padronizar estados e papéis no sistema. Os relacionamentos estabelecem a forma como essas classes interagem, como a associação de um subprojeto a várias publicações e de um colaborador a diferentes subprojetos. Para os casos de Rede_social, Email e Telefone, se tratam de formas de contato da escola PIO XII, podendo ser uma listagem dos mesmos.

3.2 Modelagem Banco de Dados

Um modelo de banco de dados é utilizado para descrever qualquer informação que seja armazenada em um banco de dados. No desenvolvimento de um projeto de banco de dados, pode-se pensar em dois graus de abstração para ser capaz de compreender os dados, sendo o modelo lógico e o modelo físico (HEUSER, 2009).

3.2.1 Modelo Lógico

O modelo lógico de dados é uma versão mais detalhada do modelo conceitual. Ele representa graficamente as restrições dos dados, os nomes das entidades e suas conexões, preparando-as para uma implementação independente de qualquer plataforma específica (SERVICES, 2025). A Figura 9 ilustra o modelo lógico, do sistema desenvolvido.

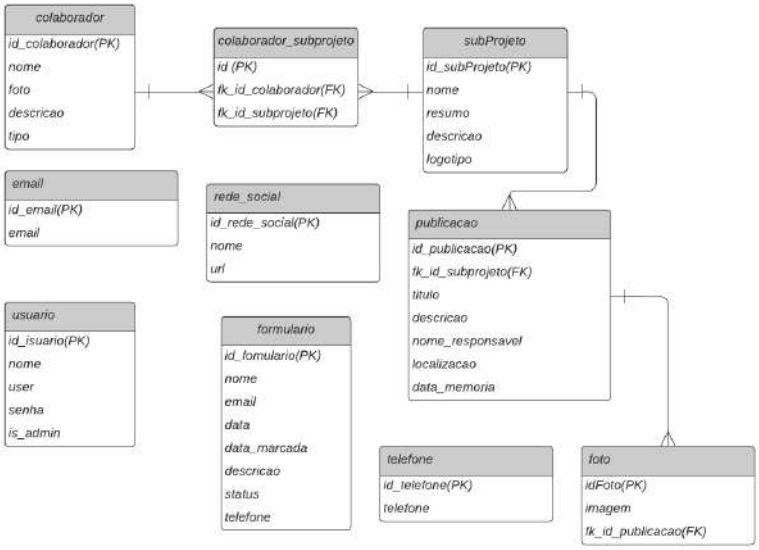

Figura 9 – Modelo lógico do Sistema

Fonte: Do Autor

3.2.2 Modelo Físico

Os modelos de dados físicos detalham como implementar a lógica de negócios em um banco de dados, incluindo atributos como acionadores, procedimentos armazenados e tipos de dados. Eles mapeiam os elementos para o banco real e seguem restrições da plataforma, como convenções de nomenclatura e palavras reservadas (SERVICES, 2025).

A Figura 10 apresenta o modelo físico do banco de dados utilizado no projeto. Ele foi estruturado para organizar as informações relacionadas ao sistema da escola PIO XII e seus subprojetos, contemplando entidades principais, relacionamentos e restrições.

Optou-se por separar as informações de contato (e-mail, telefone e redes sociais) em entidades próprias — email, telefone e rede_social. Essa decisão de modelagem tem como objetivo manter maior flexibilidade, normalização e possibilidade de expansão futura, além de facilitar a manutenção. Esses dados se referem especificamente aos meios de contato com a escola PIO XII e não a informações pessoais de usuários externos.

A entidade formulario foi criada de forma independente, pois representa os registros preenchidos por visitantes ou público externo que desejam agendar uma visita física ao projeto. Nesse contexto, cada formulário armazena dados como nome, e-mail, telefone, descrição da visita e datas relacionadas ao agendamento. As demais entidades representam a estrutura central do sistema.

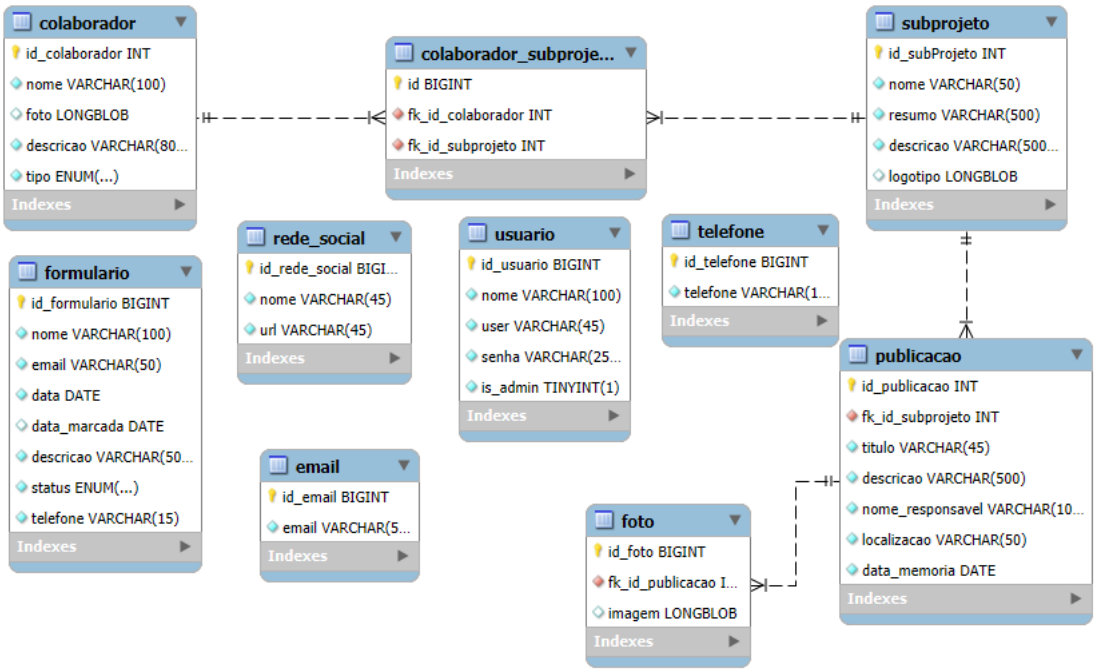

Figura 10 – Modelo físico do Sistema

Fonte: Do Autor

3.2.3 Dicionário de dados

Um dicionário de dados funciona como um glossário detalhado que descreve termos, métricas e variáveis relacionadas a um conjunto de dados. Ele é essencial para garantir que as informações sejam compreendidas de forma consistente, possibilitando a reprodutibilidade de pesquisas e a padronização no uso dos dados. Além de explicar o significado de cada variável e seus valores, o dicionário de dados facilita a comunicação entre equipes e assegura que todos compreendam o contexto de coleta, armazenamento e manipulação dos dados de maneira alinhada (DATAPSICO, 2021).

O dicionário de dados utilizado apresenta simultaneamente os modelos lógico e físico da tabela desenvolvida no sistema. A seguir, no Quadro 2 se tem a descrição dos dados de uma tabela desenvolvida no sistema.

Modelo Lógico			Modelo Físico	
Atributo	Tipo de atributo	Descrição	Nome da variável	Tipo de variável
ID do subProjeto	INTEIRO	Identificador único do subprojeto	id_subProjeto	INT
Nome do Subprojeto	TEXTO	Nome do subprojeto	nome	VARCHAR
Resumo do Subprojeto	TEXTO	Resumo do subprojeto	resumo	VARCHAR
Descrição do Subprojeto	TEXTO	Descrição do subprojeto	descricao	VARCHAR
Logotipo	IMAGEM	Arquivo de logotipo do subprojeto	logotipo	LONGBLOB

Quadro 2 – Quadro do banco de dados - SubProjeto

Fonte: Do Autor

Demais dicionários de dados das tabelas do sistema, encontram-se no Apêndice C.

3.3 Avaliação de sistemas web

Segundo Rocha e Baranauskas(2003), a avaliação é utilizada, de forma genérica, para entender quais dificuldades os usuários enfrentam e o que eles querem ao utilizarem determinada tecnologia. Com essas informações, os desenvolvedores e designer de sites são capazes de conhecer melhor os utilizadores de suas tecnologias, o público ao qual atende e suas limitações, conseguindo assim utilizar técnicas para melhorar a experiência do usuário (VIEIRA; BARANAUSKAS, 2003).

Ao realizar desenvolvimento de um sistema web é necessário que seja feito testes para garantir que as funcionalidades foram devidamente implementadas, além de verificar a robustez do mesmo, garantindo assim uma qualidade quanto ao uso do sistema. É importante ressaltar que, quanto mais cedo for identificado as correções necessárias no sistema, menor será o custo para ajustá-las (DIAS, 2014).

A realização de avaliações em sistemas de Internet para identificar áreas que podem ser melhoradas é um fator essencial. Se estas avaliações fossem realizadas, alguns problemas associados a interfaces de má qualidade poderiam ser evitados, como (DIAS, 2014):

- Insatisfação por parte dos usuários;
- Reduzi a produtividade dos utilizadores do sistema;
- Necessidade de treinamento excessivo;
- Causar confusão aos usuários;
- Desestimular o uso sistema.

Com base nos itens citados acima é possível perceber o quanto é essencial que um método de avaliação ajude os desenvolvedores a identificar, classificar e quantificar os problemas de utilização do sistema. Após a identificação, a equipe do projeto deve repensar a interface, priorizando a solução dos problemas mais críticos com base na sua gravidade e nos custos envolvidos na sua resolução (NIELSEN, 1994). Existem muitos métodos de avaliação que podem auxiliar na redução de problemas encontrados no sistema, que podem ser realizados em qualquer etapa de desenvolvimento (DIAS, 2014).

Para validar as funcionalidades implementadas no sistema, foram realizados testes com as professoras que utilizariam a plataforma. Durante esses testes, elas seguiam um checklist contendo todas as funcionalidades previstas, conferindo se cada recurso atendia de fato às necessidades e expectativas previamente especificadas. Esse processo permitiu identificar inconsistências ou ajustes necessários, garantindo que o sistema fosse desenvolvido conforme os requisitos definidos pelas usuárias.

Em resumo, as avaliações de softwares podem ser justificadas em três pontos: encontrar problemas singulares, qualificar a funcionalidades e analisar o resultado de uma interface em relação ao usuário que a utiliza (VIEIRA; BARANAUSKAS, 2003). No âmbito da avaliação da acessibilidade, o objetivo principal é reconhecer obstáculos de acessibilidade em sistemas e corrigi-los.

3.3.1 Avaliação de Acessibilidade na Web

A avaliação da acessibilidade Web é uma ação que pode ser realizada durante as fases de desenvolvimento de sistemas web ou para determinar o nível de acessibilidade obtido por determinados sistemas (DIAS, 2014).

Da mesma maneira que ocorre com a avaliação de usabilidade, a avaliação de acessibilidade utiliza-se de técnicas que envolvem a inspeção e testes com usuários reais. A inspeção equivale a avaliações realizadas por peritos em acessibilidade. Enquanto os testes com usuários devem ser feitos com usuários diversos, ou seja, com perfis distintos, que possuem diferentes deficiências, habilidades e dispositivos de acesso (FREIRE, 2008).

A W3C apresenta duas principais maneiras de periciar a acessibilidade na Web, sendo (W3C, 2022a):

- A Avaliação Preliminar de acessibilidade;
- A Avaliação de Conformidade.

Detalhando sobre esses métodos de avaliação, pode-se resumir a Avaliação Preliminar como uma forma mais rápida de identificar dificuldades de utilização de sites, sua acessibilidade. Por outro lado, a Avaliação de Conformidade é mais criteriosa utilizando-se de diretrizes, como WCAG. Este tipo de avaliação é muito parecido com a fórmula de revisão de diretrizes de usabilidade. Diferente do primeiro método de avaliação, este busca encontrar problemas mais específicos de acessibilidade, utilizando além de ferramentas, mas também empregando avaliações manuais de páginas características de maneira sistemática (FREIRE, 2008).

3.3.1.1 Ferramentas avaliadoras de Acessibilidade

Dentre as ferramentas que estão disponíveis na internet para validar a acessibilidade de um site, pode-se citar (HOSTINGER, 2022a):

- Axe DevTools[1] - essa ferramenta fornece testes automatizados que identifica e resolve problemas de acessibilidade durante o desenvolvimento de um site;
- WAVE[2] - essa ferramenta é bastante conhecida pelos desenvolvedores. Ela proporciona a identificação de problemas em sessões não acessíveis e disponibiliza soluções;
- NVACCESS[3] - essa ferramenta é um leitor de tela disponibilizado pelo Windows. Ela disponibiliza recursos de destaque e foco que ajudam os desenvolvedores a detectarem objetos-focos na tela;
- WebAim[4] - essa ferramenta é utilizada para verificação de planos de fundo e cor de textos com base nas diretrizes determinadas pelos padrões WCAG.

Com a utilização e cuidado na usabilidade do sistema, se tem um site com maior probabilidade de aceitação e, consequentemente, um grande número de acessos. Dessa forma, no desenvolvimento do sistema, foram realizados testes manuais extensivos para avaliar diferentes aspectos da acessibilidade, abrangendo desde a navegação até elementos visuais e interativos, garantindo que todos os usuários, independentemente de suas habilidades, pudessem utilizar a plataforma de forma eficiente. Esses testes incluíram a verificação da ordem de tabulação, compatibilidade com leitores de tela, contraste de cores, tamanho e

[1] https://www.deque.com/axe/devtools/
[2] https://wave.webaim.org/
[3] https://www.nvaccess.org/
[4] https://webaim.org/

legibilidade de fontes, além de análise da clareza e consistência de elementos visuais e botões de ação. Paralelamente, foi aplicado um checklist baseado nas diretrizes WCAG, permitindo uma avaliação criteriosa de conformidade com padrões reconhecidos internacionalmente.

Para complementar, foram utilizadas algumas das ferramentas citadas acima para auditoria de acessibilidade,como Axe DevTools e Wave, que auxiliaram na identificação de problemas que poderiam passar despercebidos em testes puramente manuais.

4 Desenvolvimento

Para aprimorar a gestão e a disseminação das informações do projeto "Parede da Memória", foi criado um sistema web. Esse sistema visa automatizar os processos de gerenciamento e tornar os dados mais acessíveis. Um pilar fundamental do desenvolvimento foi a acessibilidade, buscando remover barreiras para atingir o maior número de usuários possível, garantindo que o site seja inclusivo para diversos grupos de pessoas. Desenvolvido com base na metodologia ágil Scrum, o projeto utilizou o Notion[1] para organização e gerenciamento de tarefas através de um quadro de tarefas. Os requisitos e modelos detalhados no Capítulo 3 orientaram todo o desenvolvimento.

4.1 Organização dos processos

Para organizar as atividades do sistema, foi criado um quadro específico no Notion. Essa ferramenta, voltada para a organização tanto pessoal quanto profissional, oferece uma interface intuitiva com quadros e blocos que permitem estruturar e acompanhar o andamento das tarefas de maneira eficiente. Além disso, o Notion se destaca por sua flexibilidade, podendo ser totalmente personalizado de acordo com as necessidades do usuário (CANALTECH, 2023). O quadro utilizado para o desenvolvimento do site é apresentado na Figura 11.

Figura 11 – Tela do quadro de organização das atividades do sistema no Notion

Fonte: Do Autor

Estruturado com as colunas “A Fazer”, “Em Andamento” e “Concluído”, represen-

[1]Disponível em: <https://www.notion.com/>.

tando as fases do processo de desenvolvimento. Cada tarefa a ser realizada é transformada em um *card* no Notion. Esses *cards* são criados com base nos requisitos funcionais, padrões de acessibilidade e também incluem ajustes e melhorias. À medida que as funcionalidades são desenvolvidas, seus respectivos *cards* são movidos para os demais status.

4.2 A aplicação

O sistema foi desenvolvido com uma arquitetura de duas camadas de implementação: *front-end* e *back-end*. Para o desenvolvimento do *front-end*, que compreende a interface com o usuário, foram empregadas as tecnologias HTML5, CSS, JavaScript, Bootstrap e o framework Angular. Enquanto o *back-end*, responsável pela lógica e processamento de dados, foi construído utilizando Java e o framework Spring Boot. Por fim, o MySQL foi selecionado como o sistema de gerenciamento de banco de dados para armazenamento. A aplicação e a relevância de cada uma dessas tecnologias serão detalhadas nas seções seguintes.

4.2.1 Back-end

A essência da lógica de programação mais complexa e robusta em um site reside no *back-end*. Essa parte da aplicação web compreende as funcionalidades de infraestrutura que operam por trás do que é percebido pelo usuário, exigindo, portanto, um grau maior de abstração, uma vez que sua principal função é o gerenciamento da lógica do servidor (TERA, 2025).

Figura 12 – Estrutura do projeto *back-end*

Fonte: Do Autor

A Figura 12 ilustra a estrutura do *back-end* do projeto, evidenciando a arquitetura em camadas, MVC. Em complemento, o Quadro 3 detalha cada um dos pacotes e arquivos que compõem essa estrutura.

Pacotes/Arquivos	Função
Model	Responsável por representar as entidades, geralmente, mapeadas diretamente para as tabelas do banco de dados. Contém os atributos e o comportamento básico dos dados da aplicação.
DTO	São objetos simples usados para transferir dados entre as camadas. Responsáveis por controllar quais dados serão enviados e recebidos, garantido que não seja exposto informações desnecessárias da Model.
Controller	Responsável por receber as requisições HTTP do front-end (ou de outros clientes), validar a entrada, e delegar a lógica de negócio para a camada de Serviço. Sua principal função é rotear as requisições e retornar as respostas.
Service	Contém a lógica de negócio principal da aplicação. Responsável por garantir que as regras de negócio são aplicadas, validações complexas são realizadas e a orquestração entre diferentes Repositories acontece. O Controller chama o Serviço para executar uma determinada funcionalidade.
Repository	Responsável pela comunicação direta com o banco de dados. Abstrai as operações de persistência (como salvar, buscar, atualizar e deletar dados), permitindo que a camada de Serviço trabalhe com objetos Java sem se preocupar com os detalhes técnicos do banco.
Infra	Pasta contendo arquivos responsáveis pela configuração da segurança da aplicação.
Resources	Pasta contendo arquivos responsáveis pela configuração geral da aplicação.

Quadro 3 – Descrição da estrutura do projeto.

Fonte: Do Autor

4.2.1.1 Exemplificação do Back-end

Nesta seção serão apresentados os códigos de exemplificação dos arquivos que compõem o *back-end*.

4.2.1.1.1 Model

Para mostrar como o projeto foi estruturado, o código abaixo apresenta o *model* da classe 'Publicacao', que corresponde à tabela 'Publicacao' no banco de dados. Essa classe é usada pelo *back-end* para mapear e manipular os dados dessa tabela.

Publicacao.java

```
@Getter
@Setter
@Entity
```

```
@Table(name = "PUBLICACAO")
public class Publicacao {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "ID_PUBLICACAO")
    private Long id;

    @ManyToOne
    @JoinColumn(name = "FK_ID_SUBPROJETO")
    @JsonBackReference
    private SubProjeto subprojeto;

    @Column(name = "TITULO", length = 45)
    private String titulo;

    @Column(name = "DESCRICAO", length = 500)
    private String descricao;

    @Column(name = "NOME_RESPONSAVEL", length = 100)
    private String nomeResponsavel;

    @Column(name = "LOCALIZACAO", length = 50)
    private String localizacao;

    @Temporal(TemporalType.DATE)
    @Column(name = "DATA_MEMORIA")
    private LocalDate dataMemoria;

    @OneToMany(mappedBy = "publicacao", fetch = FetchType.LAZY)
    @Fetch(FetchMode.SUBSELECT)
    @JsonManagedReference
    private List<Foto> fotos;
}
```

Cada campo da classe representa uma coluna da tabela no banco. Além disso, foi adicionado manualmente o atributo *List<Foto> fotos* na classe, mesmo que esse campo não exista diretamente na tabela de 'Publicacao'. Essa lista foi incluída apenas como uma referência para facilitar o acesso às fotos ligadas à publicação, visto que toda publicação possuí fotos associadas a ela. Com esse mapeamento, é possível acessar facilmente todas as fotos de uma publicação dentro do sistema, sem precisar fazer consultas separadas.

4.2.1.1.2 DTO

O código abaixo mostra a estrutura do *DTO (Data Transfer Object)* da classe 'Publicacao'. Esse tipo de classe é utilizado no *back-end* para controlar quais informações da classe original (*Model*) serão compartilhadas com outras partes do sistema, como o *front-end* ou APIs externas.

PublicacaoDTO.java

```
@Getter
@Setter
public class PublicacaoDTO {
    private Long id;
    private String titulo;
    private String descricao;
    private String nomeResponsavel;
    private String localizacao;
    private LocalDate dataMemoria;

    public PublicacaoDTO(Publicacao publicacao) {
        this.id = publicacao.getId();
        this.titulo = publicacao.getTitulo();
        this.descricao = publicacao.getDescricao();
        this.nomeResponsavel = publicacao.getNomeResponsavel();
        this.localizacao = publicacao.getLocalizacao();
        this.dataMemoria = publicacao.getDataMemoria();
    }
}
```

A ideia principal do *DTO* é evitar o envio de todos os dados da entidade, expondo apenas os campos realmente necessários para a operação desejada. No exemplo acima, o 'PublicacaoDTO' contém apenas os atributos mais importantes da publicação, como id, titulo, descricao, nomeResponsavel, localizacao e dataMemoria.

No construtor da classe, os valores são copiados diretamente de uma instância da classe Publicacao, permitindo que o sistema trabalhe com uma versão que contém as informações mais relevantes.

4.2.1.1.3 Controller

O código abaixo apresenta a classe *PublicacaoController*, que é um controlador REST responsável por gerenciar requisições HTTP relacionadas as publicações do sistema.

PublicacaoController.java

```
@RestController
@RequestMapping("/api/publicacoes")
public class PublicacaoController {

    private final PublicacaoService publicacaoService;

    public PublicacaoController(PublicacaoService publicacaoService) {
        this.publicacaoService = publicacaoService;
    }
```

É possível visualizar uma notação *@RestController* e também o mapeamento para o caminho base */api/publicacoes*, essa estrutura funciona como uma ponte entre o cliente e a camada de serviço *PublicacaoService*, que contém a lógica de negócio. Logo no ínicio da classe, é possível observar a instância desta camada de serviço e sua inicialização através do construtor.

Além dos métodos responsáveis pelas operações CRUD do fluxo, a classe apresenta o método *getFotosDaPublicacao*, que pode ser visualizado no código abaixo, este método que é responsável por buscar e retornar todas as fotos associadas a uma publicação específica, identificada pelo seu id. Utilizando o serviço *publicacaoService*, ele obtém uma lista fotos associadas a publicação e realiza o retorno.

PublicacaoController.java

```
    @GetMapping("/{id}/fotos")
    public ResponseEntity<List<FotoDTO>> getFotosDaPublicacao(@PathVariable
        Long id) {
        List<Foto> fotos = publicacaoService.getFotosPorPublicacao(id);
        List<FotoDTO> response = fotos.stream()
                .map(FotoDTO::new)
                .toList();
        return ResponseEntity.ok(response);
    }
```

4.2.1.1.4 Service

A classe de serviço é responsável por implementar as regras de negócio relacionadas à entidade. A camada de serviço serve como ponte entre a camada de controle (*Controller*) e a camada de acesso aos dados (*Repository*). Ela é responsável por validar, tratar e processar as informações antes que sejam salvas, atualizadas ou retornadas ao usuário.

PublicacaoService.java

```
@Service
public class PublicacaoService {

    private final PublicacaoRepository publicacaoRepository;
    private final SubProjetoRepository subprojetoRepository;

    public PublicacaoService(PublicacaoRepository publicacaoRepository,
            SubProjetoRepository subprojetoRepository) {
        this.publicacaoRepository = publicacaoRepository;
        this.subprojetoRepository = subprojetoRepository;
    }
```

No código é apresentado um trecho do código que pertence à classe *PublicacaoService*, onde inicialmente é feito a definição das dependências dos repositórios *PublicacaoRepository* e *SubProjetoRepository*, que são responsáveis por acessar os dados persistidos no banco referentes às entidades Publicacao e SubProjeto. Ambos os repositórios são declarados como atributos final.

Além dos métodos envolvendo CRUD, foram implementados métodos auxiliares de validação dos dados, esses métodos podem ser visualizados no código abaixo e descritos a seguir.

PublicacaoService.java

```
    public List<String> validarPublicacao(Publicacao publicacao) {
        List<String> errors = new ArrayList<>();

        if (publicacao.getSubprojeto() == null) {
            errors.add("A identificação do subProjeto não pode ser vazio");
        } else if
            (!subprojetoRepository.existsById(publicacao.getSubprojeto().getId()))
            {
            errors.add("SubProjeto não encontrado.");
        }
        if (publicacao.getTitulo() == null ||
            publicacao.getTitulo().isBlank()) {
            errors.add("O campo 'título' éobrigatório.");
        } else if (publicacao.getTitulo().length() > 45) {
            errors.add("O campo 'título' deve ter no máximo 45 caracteres.");
        }
        if (publicacao.getDescricao() == null ||
```

```
            publicacao.getDescricao().isBlank()) {
            errors.add("O campo 'descrição' éobrigatório.");
        } else if (publicacao.getDescricao().length() > 500) {
            errors.add("O campo 'descrição' deve ter no máximo 500
                caracteres.");
        }
        if (publicacao.getDataMemoria() == null) {
            errors.add("O campo 'data memória' éobrigatório.");
        } else if (!isValidDate(publicacao.getDataMemoria())) {
            errors.add("A 'data memória' deve ser igual ou anterior àdata
                atual.");
        }
        if (publicacao.getNomeResponsavel() == null ||
            publicacao.getNomeResponsavel().isBlank()) {
            errors.add("O campo 'nome do responsável' éobrigatório.");
        } else if (publicacao.getNomeResponsavel().length() > 100) {
            errors.add("O campo 'nome do responsável' deve ter no máximo 100
                caracteres.");
        }
        if (publicacao.getLocalizacao() == null ||
            publicacao.getLocalizacao().isBlank()) {
            errors.add("O campo 'localização' éobrigatório.");
        } else if (publicacao.getLocalizacao().length() > 50) {
            errors.add("O campo 'localização' deve ter no máximo 50
                caracteres.");
        }
        return errors;
    }

    private void validarDuplicidade(Publicacao publicacao, List<String>
        errors) {
        if (publicacaoRepository.existsByTitulo(publicacao.getTitulo())) {
            errors.add("Já existe uma publicação com este título!");
        }
    }

    private boolean isValidDate(LocalDate data) {
        return !data.isAfter(LocalDate.now());
    }
}
```

- *validarPublicacao()*: verifica se todos os campos obrigatórios foram preenchidos corretamente, respeitando limites de tamanho e regras de negócio, como a data não poder ser futura e a existência do SubProjeto relacionado.
- *validarDuplicidade()*: garante que não exista outra publicação com o mesmo título já cadastrada no sistema.
- *isValidDate()*: valida se a data informada é igual ou anterior à data atual, evitando registros com datas futuras inválidas.

4.2.1.1.5 Repository

No código abaixo é apresentado a interface 'PublicacaoRepository', que representa a camada de acesso aos dados da classe 'Publicacao'. Essa interface estende a interface JpaRepository, fornecida pelo Spring Data JPA, que já contém diversos métodos prontos para consultas, inserções, atualizações e exclusões no banco de dados, como *findAll(), findById(), save() e deleteById()*.

PublicacaoRepository.java

```
@Repository
public interface PublicacaoRepository extends JpaRepository<Publicacao, Long> {
    boolean existsByTitulo(String titulo);
}
```

Além dos métodos herdados do JpaRepository, foi criado um método personalizado:

- *existsByTitulo(String titulo)*: verifica se já existe uma publicação cadastrada com o título informado. Esse método é utilizado na camada de serviço para evitar duplicidade de títulos, garantindo integridade e consistência das informações.

4.2.1.2 Conexão com o banco de dados

Para a conexão com o banco de dados MySQL, o projeto utilizou o Spring Data JPA em conjunto com o Hibernate. O Spring Data JPA simplifica a implementação de repositórios proporcionando uma abordagem mais ágil e clara para o acesso a dados. Essa abstração reduz a complexidade da interação com o banco de dados, promovendo o desenvolvimento de aplicações Spring de forma mais padronizada, eficiente e alinhada às boas práticas de persistência (SPRING, 2025).

A configuração da conexão foi gerenciada no arquivo *application.properties* , onde foi definidos os parâmetros essenciais para o acesso ao MySQL.

O código abaixo apresenta as configurações descritas no arquivo.

application.properties

```
spring.datasource.url=jdbc:mysql://localhost:3306/bd_paredememoria
spring.datasource.username=root
spring.datasource.password=Root@123
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver

spring.jpa.database-platform=org.hibernate.dialect.MySQLDialect
spring.jpa.hibernate.ddl-auto=none
spring.jpa.show-sql=true
spring.jpa.properties.hibernate.format_sql=true
spring.jpa.properties.hibernate.cache.use_second_level_cache=false

server.error.include-stacktrace=never

spring.servlet.multipart.enabled=true
spring.servlet.multipart.max-file-size=10MB
spring.servlet.multipart.max-request-size=10MB
logging.level.org.springframework.security=DEBUG
api.security.token.secret=${JWT_SECRET:my-secret-key}

logging.level.org.hibernate.SQL=DEBUG
logging.level.org.hibernate.type.descriptor.sql.BasicBinder=TRACE
```

A seguir, são descritos os principais parâmetros utilizados:

- *spring.datasource.url*: define a URL de conexão com o banco de dados, especificando o protocolo JDBC, o host (*localhost*), a porta (*3306*) e o nome do banco (*bd_paredememoria*).
- *spring.datasource.username* e *spring.datasource.password*: definem, respectivamente, o nome de usuário e a senha utilizados para autenticação no banco.
- *spring.datasource.driver-class-name*: especifica a classe do driver JDBC responsável pela comunicação com o banco de dados MySQL (no caso, o driver oficial atualizado do MySQL).
- *spring.jpa.database-platform*: especifica que o sistema usará o dialeto SQL do MySQL, que adapta os comandos para o formato que o MySQL entende.
- *spring.jpa.hibernate.ddl-auto*: configuração importante que define que o sistema não deve criar ou alterar automaticamente a estrutura do banco de dados (tabelas). Isso é controlado manualmente para evitar perda de dados.

Essas configurações garantem que a aplicação interaja com o banco de dados de forma eficiente e segura. Elas estabelecem uma conexão bem definida entre o sistema e o banco de dados MySQL, utilizando credenciais específicas para evitar comportamentos indesejados, como a modificação acidental da estrutura do banco de dados.

4.2.1.3 Segurança do sistema

A segurança é um dos pilares essenciais em qualquer sistema, e, para proteger os dados e controlar o acesso à aplicação, foi adotada a autenticação baseada em JSON Web Tokens (JWT), um padrão aberto, definido pelo RFC 7519, que estabelece um método compacto e autossuficiente para a transmissão segura de informações entre partes na forma de um objeto JSON (JWT.IO, 2025).

Na aplicação desenvolvida com Spring Boot, a geração e validação dos JWTs é feita com o uso da biblioteca *java-jwt*, fornecida pela *Auth0*. Quando um usuário efetua login, suas credenciais são verificadas e, caso sejam válidas, um token JWT é gerado e enviado ao cliente. Esse token carrega informações relevantes sobre o usuário, como seu ID e nome de usuário, além de uma data de expiração, assegurando que o acesso seja concedido apenas dentro de um período determinado.

A Figura 28 ilustra o processo de requisição de login, no qual são enviadas as credenciais de usuário e senha ao sistema. Em resposta, o sistema retorna um objeto no formato JSON contendo um token de autenticação. Esse token armazena informações relativas ao usuário que realizou o login, bem como o tempo de expiração definido para a sessão, garantindo maior segurança no acesso ao sistema.

Figura 13 – Exemplo de login realizado pelo *back-end*

Fonte: Do Autor

4.2.2 Front-end

No desenvolvimento web, o *front-end* é a parte da aplicação responsável por tudo o que o usuário visualiza e com o que interage diretamente ao acessar uma página ou sistema. Isso inclui elementos como botões, menus, formulários, cores, imagens e a organização do layout, sendo essencial para proporcionar uma experiência agradável e funcional. Em resumo, o *front-end* conecta a lógica do sistema à experiência do usuário, tornando possível

a interação intuitiva com o conteúdo apresentado (TERA, 2025).

A Figura 14 ilustra a estrutura do *front-end* do projeto. Em complemento, o Quadro 4 detalha cada um dos pacotes e arquivos que compõem este projeto.

Figura 14 – Estrutura do projeto *front-end*

Fonte: Do Autor

Pacotes/Arquivos	Função
Pages	Pasta que contém os componentes Angular de todas as telas da aplicação
Services	Responsável pela comunicação com o *back-end* pela autenticação
Shared	Pasta que contém os componentes Angular que são reutilizados em todo resto da aplicação.
app.routes.ts	Responsável pelo gerenciamento das rotas da aplicação.
assets	Pasta que contém os arquivos que não são código-fonte, mas necessário na aplicação, como: imagens e icons.

Quadro 4 – Descrição da estrutura do projeto *front-end*.

Fonte: Do Autor

4.2.2.1 Exemplificação do Front-end

Nesta seção serão apresentados os códigos de exemplificação dos arquivos que compõem o *front-end*.

4.2.2.1.1 Component

Para o desenvolvimento do projeto, foi utilizada a estrutura padrão de um componente no Angular, gerada por meio do comando *ng generate component*. A partir desse comando, são criados automaticamente os arquivos de código. Para exemplificar esses arquivos, será apresentado abaixo a explicação das partes mais importantes dentro dos arquivos.

Os códigos abaixo se tratam da página *HTML* que está integrada com o Angular, dessa forma tendo diretrizes condicionais *(@if, *ngFor)* e eventos de interação para exibir uma lista de projetos. Na estrutura da página é possível ver a seguinte organização:

- Cabeçalho da seção: No código abaixo é apresentado a estrutura de cabeçalho da página, onde é exibido o título "Nossos Projetos"com um ícone. Caso o usuário autenticado seja um administrador (*isAdmin()*), é exibido também o botão "Adicionar Projeto".

list-projetos.component.html - Cabeçalho

```
<div class="projeto-main">
  <div class="section-header">
    <h1 class="section-title">
      <i class="fa fa-newspaper" aria-hidden="true"></i> Nossos Projetos
    </h1>
    @if (isAdmin()) {
      <button
        class="header-button"
        (click)="adicionarProjeto()"
        aria-label="Adicionar Projeto">
        Adicionar Projeto
      </button>
    }
  </div>
```

- Navegação rápida: No código abaixo é apresentado uma lista de links para cada projeto, permitindo ao usuário rolar rapidamente até a seção correspondente. Se não houver projetos cadastrados, é exibida uma mensagem informativa.

list-projetos.component.html - Navegação

```
    <p>Listagem de todos os projetos que possumos.</p>
    <div id="navigation-help" class="sr-only" tabindex="-1">
```

```
    Use as setas do teclado para navegar entre os perfis
  </div>
  <nav aria-labelledby="profile-nav" class="profile-index">
    <h2 class="sr-only" id="profile-nav">
      Navegue diretamente para um perfil
    </h2>
    <ul>
      @if (projetos && projetos.length > 0) {
        <li *ngFor="let projetos of projetos">
          <a href="javascript:void(0)" (click)="scrollToProfile(projetos.id)">
            {{ projetos.nome }} - Ver projeto
          </a>
        </li>
      } @else {
        <li>
          <span>Nenhum projeto cadastrado</span>
        </li>
      }
    </ul>
  </nav>
```

- Listagem completa dos projetos: No código é apresentado, para cada projeto, as seguintes informações:
  - O nome, com link para a página de detalhes do projeto.
  - Botões "Editar"e "Excluir", visíveis apenas para administradores autenticados (*isAuthenticated()* e *isAdmin()*).
  - A imagem do projeto, com *fallback* para uma imagem padrão caso a original falhe ao carregar.
  - Um pequeno resumo sobre o projeto.
- Acessibilidade: Elementos com *aria-label* são utilizados para melhorar a navegação por leitores de tela.

list-projetos.component.html - Lista

```
  @if (projetos && projetos.length > 0) {
    <div class="projeto-full" *ngFor="let projeto of projetos">
      <div class="projeto-header">
        <h2 class="projeto-titulo">
          <a [routerLink]="['/acervo', projeto.id]" [state]="{ projeto:
```

```
            projeto }">
            {{ projeto?.nome }}
          </a>
        </h2>

        @if (isAuthenticated() && isAdmin()) {
          <div class="projeto-actions-inline">
            <button class="header-button" (click)="editarProjeto(projeto.id)"
                [attr.aria-label]="'Editar' +projeto?.nome" >
              <i class="fa fa-edit"></i>
            </button>
            <button class="header-button" (
                click)="excluirProjeto(projeto.id)"
                [attr.aria-label]="'Excluir '+projeto?.nome">
              <i class="fa fa-trash"></i>
            </button>
          </div>
        }
      </div>

      <div class="projeto-content" id="project-{{ projeto.id }}">
        <div class="projeto-image-container">
          <img
            [src]="projeto.imagemUrl || 'assets/imagens/default-project.jpg'"
            [alt]="' cone do projeto '+projeto.nome"
            (error)="handleImageError($event)"
            class="projeto-img"
          />
        </div>

        <div class="projeto-text">
          <p>{{ projeto.resumo }}</p>
        </div>
      </div>
    </div>
  } @else {
    <div class="nenhum-projeto">
      <p>Nenhum projeto cadastrado</p>
    </div>
  }
```

Para a estilização da página foi gerado o arquivo utilizando *CSS*, com foco em

responsividade, identidade visual e acessibilidade. Abaixo, estão destacadas as principais características e organização do estilo aplicado:

- Plano de fundo e layout principal: A classe *.projeto-main* define o contêiner geral da página, com um fundo composto por um gradiente semi-transparente sobreposto a uma imagem externa. A estrutura utiliza *flexbox* em coluna, garantindo que os elementos se organizem verticalmente de forma responsiva.

Abaixo é possível visualizar o trecho do CSS responsável pela estilização do plano de fundo e layout principal.

list-projetos.component.css

```
  .projeto-main{
    background: linear-gradient(rgba(245,232,173, 0.5), rgba(245,232,173,
        0.5)),
    url("../../../../assets/imagens/joe-woods-4Zaq5xY5M_c-unsplash.jpg");
    width: 100%;
    min-height: 100vh;
    padding: 0 2rem 0 2rem;
    min-width: 280px;
    display: flex;
    flex-direction: column;
  }
.projeto-container {
  margin-bottom: 2rem;
  padding: 2rem 1em 2rem 1em;
  background: #F5E8AD;
  border-radius: 1.2rem;
  box-shadow: 0 10px 20px rgba(0, 0, 0, 0.19), 0 6px 6px rgba(0, 0, 0, 0.23);
}
.projeto-container p{
  margin-bottom: 0.5rem;
}
  .nenhum-projeto {
    text-align: center;
    padding: 40px;
    font-size: 1.2rem;
    color: #666;
  }
.section-header {
  text-align:center;
  display: flex;
```

```
  justify-content: space-between;
  align-items: center;
}
```

Enquanto o trecho abaixo se refere ao código responsável pela responsividade da página.

list-projetos.component.css - Responsividade

```
@media (max-width: 768px) {
  .projeto-main{
    padding: 0 1.5rem;
  }
  .projeto-header{
    display: flex;
    flex-direction: column-reverse;
    gap:0.5rem;
  }
  .projeto-full{
    margin-bottom: 2rem;
    padding: 1rem;
    max-height: 86rem;
  }
  .section-title{
    font-size: 1.2rem;
  }
  .section-title i.fa-newspaper{
    font-size: 2rem;
  }
  .header-button{
    font-size: 0.7rem;
    margin: 0.5rem;
  }
  .projeto-container h2{
    font-size: 1.5rem;
  }
  .projeto-img{
    max-width: 100%;
  }
  .projeto-actions-inline button{
    font-size: 1rem;
    margin: 0;
  }
```

```
  }
```

O código abaixo refere-se a um teste unitário desenvolvido com *Jasmine* e configurado utilizando o módulo de testes do *Angular* (*@angular/core/testing*). Esse teste tem como objetivo verificar se o componente *ProjetosComponent* é criado corretamente, sendo um arquivo gerado automaticamente ao criar um componente.

list-projetos.component.spec.ts

```
import { ComponentFixture, TestBed } from '@angular/core/testing';

import { ProjetosComponent } from './list-projetos.component';

describe('ProjetosComponent', () => {
  let component: ProjetosComponent;
  let fixture: ComponentFixture<ProjetosComponent>;

  beforeEach(async () => {
    await TestBed.configureTestingModule({
      imports: [ProjetosComponent]
    })
    .compileComponents();

    fixture = TestBed.createComponent(ProjetosComponent);
    component = fixture.componentInstance;
    fixture.detectChanges();
  });

  it('should create', () => {
    expect(component).toBeTruthy();
  });
});
```

O código *list-projetos.component.ts* representa um trecho da estrutura lógica do componente responsável por gerenciar e exibir os projetos. Desenvolvido com a linguagem *TypeScript*, ele é integrado ao Angular e segue o padrão de componentes independentes (*standalone*), permitindo reutilização mais modular. Sua organização apresenta as seguintes funcionalidades:

- Configuração inicial: Por meio do decorador `@Component`, são definidos o seletor do componente, os arquivos de estilo e o conteúdo HTML utilizado para renderização da interface. Também são importados serviços e módulos necessários, como o *Router*,

HttpClient e serviços personalizados.

- Definição do título da página: Utiliza-se o serviço *Title* para definir dinamicamente o título da aba do navegador com o texto "Nossos Projetos | Parede da Memória".
- Carregamento de informações: No método *ngOnInit()*, é chamada a função *carregarProjetos()*, responsável por buscar todos os projetos cadastrados no *backend* por meio do serviço *ProjetoService*, organizando-os em ordem alfabética.
- Controle de autenticação e autorização: Os métodos *isAuthenticated()* e *isAdmin()* verificam, respectivamente, se o usuário está logado e se possui permissões administrativas, permitindo controlar o acesso às funções de edição e exclusão.

list-projetos.component.ts

```
@Component({
  selector: 'app-projetos',
  standalone: true,
  imports: [ CommonModule, RouterModule ],
  templateUrl: './list-projetos.component.html',
  template: `
    <div *ngFor="let projeto of projetos" class="projeto-card">
      <img
        [src]="projeto.imagemUrl || 'assets/imagens/default.jpg'"
        [alt]="projeto.nome"
        (error)="handleImageError($event)"
      />
      <h3>{{ projeto.nome }}</h3>
    </div>`,
  styleUrls: ['./list-projetos.component.css']
})
export class ProjetosComponent {
  private titleService = inject(Title);
  private http = inject(HttpClient);
  private router = inject(Router);
  projetos: any[] = [];
  activeCard= 1;

  constructor(private projetoService: ProjetoService, private authService:
      AuthService, private modalService: ModalService) {
    this.titleService.setTitle('Nossos Projetos | Parede da Memória');
  }
```

```
ngOnInit() {
  this.carregarProjetos();
}

isAuthenticated(): boolean {
  return this.authService.isLoggedIn;
}

isAdmin(): boolean{
  return this.authService.isAdmin();
```

Enquanto no código abaixo é apresentado as implementações das funções de gerenciamento como *adicionarProjeto()*, *editarProjeto()* e *excluirProjeto()*, todas com navegação ou comunicação com o*backend*. A exclusão é protegida por um modal de confirmação e exige token de autenticação.

list-projetos.component.ts

```
adicionarProjeto() {
  this.router.navigate(['/form-projeto']);
}

editarProjeto(projetoId: number) {
  this.router.navigate(['/form-projeto', projetoId]);
}

excluirProjeto(projetoId: number) {
  const token = this.authService.getToken();
  if (!token) {
    this.handleUnauthorized();
    return;
  }
  this.modalService.show({
    title: 'Confirmação',
    content: 'Tem certeza que deseja excluir este projeto?',
    type: 'warning',
    confirmText: 'Sim',
    cancelText: 'Não',
    onConfirm: () => {
      const headers = new HttpHeaders({
        'Authorization': `Bearer ${token}`
      });
```

```
      this.http.delete(`http://localhost:8080/api/subprojetos/${projetoId}`,
          { headers })
        .subscribe({
          next: () => {
            this.modalService.show({
              title: 'Sucesso!',
              content: 'Projeto excluído com sucesso.',
              type: 'success',
              showCancel: false,
              onConfirm: () => window.location.reload()
            });
          },
          error: (error) => {
            this.modalService.show({
              title: 'Erro',
              content: error.error?.message || 'Não foi possível excluir o
                  projeto.',
              type: 'error',
              showConfirm: false,
              cancelText: 'Entendi'
            });
          }
        });
    },
    onCancel: () => {
    }
  });
}
```

4.2.2.1.2 Service

Para o desenvolvimento do projeto, foi utilizado também alguns serviços Angular, como *ProjetoService* que é responsável por gerenciar as operações relacionadas aos projetos. Este serviço segue o padrão de injeção de dependências do Angular e implementa as seguintes funcionalidades:

- Comunicação com a API: O serviço utiliza o HttpClient para realizar requisições HTTP ao *backend*, concentrando toda a lógica de comunicação com o *endpoint /api/subprojetos*.
- Conversão de Imagens: Implementa o método *converterBytesParaImagem()* que trans-

forma *bytes* recebidos da API em URLs de imagens no formato base64, garantindo a exibição correta dos logotipos dos projetos.

- Gestão de Autenticação: Inclui automaticamente o token *JWT* no cabeçalho das requisições que exigem autenticação, como a exclusão de acervos.
- Operações CRUD: Possuí os métodos:
  - Listagem completa de projetos *(getProjetos())*
  - Busca por ID *(getProjetoById())*
  - Obtenção de publicações relacionadas *(getAcervosByProjetoId())*
  - Exclusão de acervos *(excluirAcervo())*

Segue abaixo o código referente ao *ProjetoService*:

projeto.service.ts

```
@Injectable({
  providedIn: 'root'
})
export class ProjetoService {
  private apiUrl = 'http://localhost:8080/api/subprojetos';

  constructor(private http: HttpClient, private authService: AuthService) { }

  getProjetos(): Observable<any[]> {
    return this.http.get<any[]>(this.apiUrl).pipe(
      map(projetos => projetos.map(projeto => ({
        ...projeto,
        imagemUrl: this.converterBytesParaImagem(projeto.logotipo)
      }))
    ));
  }

  private converterBytesParaImagem(imageBytes: any): string | null {
    if (!imageBytes || imageBytes.length === 0) return null;
    try {
      if (typeof imageBytes === 'string') {
        return `${imageBytes}`;
      }
      const base64String = btoa(
        new Uint8Array(imageBytes).reduce(
          (data, byte) => data + String.fromCharCode(byte),''
```

```
      )
    );
    return `${base64String}`;
  } catch (e) {
    console.error('Erro ao converter imagem:', e);
    return null;
  }
}

getProjetoById(id: number) {
  return this.http.get<any[]>(`${this.apiUrl}/${id}`);
}

getAcervosByProjetoId(projetoId: number): Observable<any[]> {
  return this.http.get<any[]>(`${this.apiUrl}/${projetoId}/publicacoes`);
}
excluirAcervo(acervoId: number): Observable<any> {
  const token = this.authService.getToken();
  const headers = new HttpHeaders({
    'Authorization': `Bearer ${token}`
  });
  return this.http.delete(`${this.apiUrl}/acervos/${acervoId}`, { headers });
}
}
```

4.2.2.2 Rotas de navegação

Para o gerenciamento e controle das rotas de navegação no sistema *front-end*, foi utilizado o módulo de roteamento do Angular. Essa ferramenta nativa do framework Angular permite a implementação de roteamento interno na aplicação, possibilitando que o usuário transite entre diferentes páginas do sistema (ANGULAR, 2025).

As rotas do sistema são centralizadas e definidas no arquivo *app.routes.ts*. Neste arquivo, cada rota é associada a um *path* (o segmento da URL) e a um *component* (o componente Angular que será carregado e exibido quando essa URL for acessada). A seguir, é apresentada a configuração de rotas do projeto:

list-projetos.component.ts

```
export const routes: Routes = [
  { path: '', redirectTo: 'home', pathMatch: 'full' },
  { path: 'home', component: HomeComponent },
  { path: 'faleConosco', component: FaleConoscoComponent },
```

```
  { path: 'projetos', component: ProjetosComponent },
  { path: 'sobre', component: SobreComponent },
  { path: 'list-quemSomos', component: ListQuemSomosComponent},
  { path: 'form-quemSomos', component: FormQuemSomosComponent, canActivate:
      [AuthGuard] },
  { path: 'form-quemSomos/:id', component: FormQuemSomosComponent,
      canActivate: [AuthGuard] },
  { path: 'login', component: LoginComponent },
  { path: 'form-projeto', component: FormProjetoComponent, canActivate:
      [AuthGuard] },
  { path: 'form-projeto/:id', component: FormProjetoComponent, canActivate:
      [AuthGuard] },
  { path: 'form-usuario', component: FormUsuarioComponent, canActivate:
      [AuthGuard]},
  { path: 'list-usuario', component: ListUsuarioComponent, canActivate:
      [AuthGuard]},
  { path: 'acervo/:id', component: ListAcervoComponent },
  { path: 'form-acervo', component: FormAcervoComponent, canActivate:
      [AuthGuard] },
  { path: 'form-acervo/:id', component: FormAcervoComponent },
  { path: 'list-formulario', component: ListFormularioComponent, canActivate:
      [AuthGuard] },
];
```

Para exemplificar a navegação, ao clicar em um item de menu que leva, por exemplo, à página de edição de um projeto, o Angular utiliza o método de navegação (*router.navigate()*) com a rota e, se necessário, parâmetros (*/form-projeto/:id*). Isso faz com que o componente *FormProjetoComponent* seja carregado e exibido dinamicamente na tela, proporcionando uma experiência de usuário fluida e sem recarregamento completo da página.

4.2.3 Serviço de envio de e-mail

Durante o desenvolvimento do sistema, foi implementada uma funcionalidade essencial: o envio automático de e-mails para confirmar os pedidos de agendamento de visitas ao espaço físico do projeto "Parede da Memória". Para isso, foi utilizado o serviço EmailJS, que permite o envio de e-mails diretamente a partir do código JavaScript. Com essa ferramenta, não foi necessário configurar um servidor mais robusto pois o uso do EmailJS[2] simplificou esse processo, tornando mais rápida e prática a implementação dessa funcionalidade.

[2]Disponível em: <https://www.emailjs.com/>.

4.2.3.1 Configuração do EmailJS

Para que fosse possível utilizar o serviço foram necessárias as seguintes configurações:

Cadastro no EmailJS: Criação de uma conta no serviço EmailJS para obter as credenciais necessárias para a integração.

Configuração do Serviço: No painel administrativo do EmailJS, foi adicionado um novo "Email Service", configurando-o com o provedor de e-mail que seria utilizado para enviar as mensagens (no caso, o Gmail), que pode ser visualizado na Figura 15.

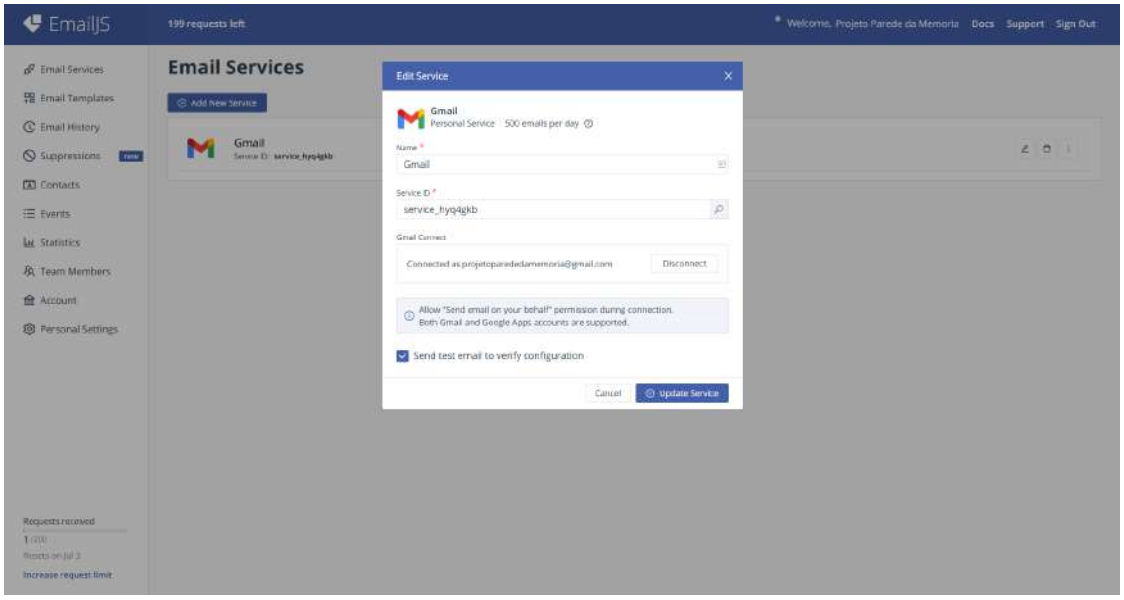

Figura 15 – Configuração do serviço de envio de e-mail

Fonte: Do Autor

Criação do Template de E-mail: Foi desenvolvido um template personalizado para o e-mail de confirmação, incluindo variáveis dinâmicas que seriam preenchidas pelos dados do formulário. O template, Figura 16, foi configurado com o ID e inclui placeholders para:

- Nome do visitante (from_name)
- E-mail do visitante (from_email)
- Data da visita (data_visita)
- Descrição da solicitação (descricao)
- Telefone para contato (telefone)

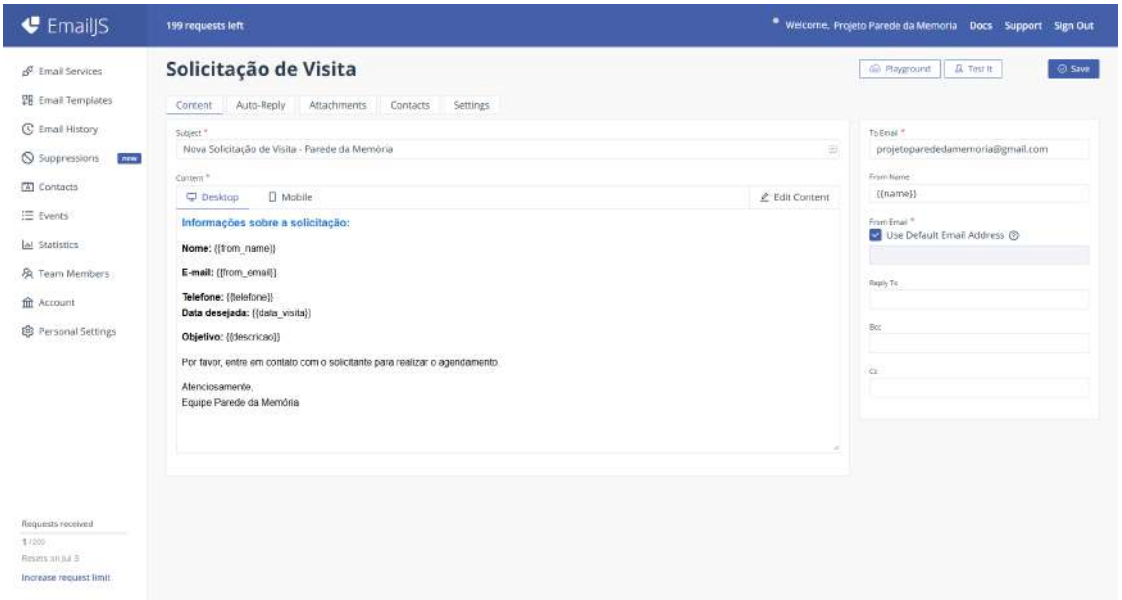

Figura 16 – Configuração do template de envio de e-mail

Fonte: Do Autor

4.2.3.2 Integração no código Angular

Para utilizar o EmailJS em um projeto Angular, foi necessário instalar o pacote *emailjs-com* via npm (Node Package Manager). O comando utilizado foi:

```
npm install emailjs-com
```

Isso adicionou a biblioteca ao projeto, permitindo a importação e uso dos métodos necessários para enviar e-mails. Após essa etapa inicial, foi realizado uma importação da biblioteca EmailJS no arquivo do componente (*fale-conosco.component.ts*).

```
import emailjs from 'emailjs-com';
```

Dessa forma, sendo possível acessar as funções principais do EmailJS, como *init()* (para inicialização) e *send()* (para disparar e-mails). No construtor do componente (*fale-conosco.component.ts*) foi inicializado o EmailJS com a chave pública da conta, obtida no painel de controle do EmailJS, conforme código:

```
  constructor(
    private fb: FormBuilder,
    private modalService: ModalService,
    private datePipe: DatePipe,
    private http: HttpClient,
    private router: Router,
    private authService: AuthService
```

```
) {
  this.contatoForm = this.fb.group({
    nome: ['', [Validators.required,
        Validators.maxLength(this.inputLimits.nome)]],
    email: ['', [Validators.required, Validators.email,
        Validators.maxLength(this.inputLimits.email)]],
    data: ['', [Validators.required, this.futureDateValidator]],
    descricao: ['', [Validators.required,
        Validators.maxLength(this.inputLimits.descricao)]],
    telefone: ['', [Validators.required,
        Validators.maxLength(this.inputLimits.telefone)]]
  });
  this.titleService.setTitle('Fale Conosco | Parede da Memória');
  emailjs.init('69dLhtgMr-Oagk5_f'); //Chave publica do EmailJS
  this.loadContactInfo();
}
```

Essa chave vincula o *frontend* ao serviço do EmailJS, permitindo o envio de e-mails sem expor credenciais sensíveis. Ainda dentro deste componente, o envio do e-mail ocorre após o formulário ser validado e os dados serem enviados ao *backend*. O método *enviarEmailConfirmacao()*, apresentado no código abaixo, é responsável por:

1. Formatar os dados do formulário.
2. Preencher o template do e-mail com as variáveis dinâmicas.
3. Disparar o e-mail usando o EmailJS.

```
private async enviarEmailConfirmacao() {
  const dataFormatada =
      this.datePipe.transform(this.contatoForm.value.data,'dd/MM/yyyy') ||
      this.contatoForm.value.data;

  //Parâmetros dinâmicos que preenhem o template
  const templateParams = {
    from_name: this.contatoForm.value.nome, //Nome do remetente
    from_email: this.contatoForm.value.email, //E-mail do remetente
    data_visita: dataFormatada, // Data fdrmatada
    descricao: this.contatoForm.value.descricao || 'Não informado',
        //Descrição
    telefone: this.contatoForm.value.telefone //Telefone para contato
  };
```

```
  //Envia o e-mail usando o EmailJS
  await emailjs.send(
    'service_hyq4gkb', //ID do serviço configurado no EmailJS
    'visita_template', // ID do tempalte de e-mail
    templateParams // Dados dinâmicos para preenchimento
  );
}
```

Parâmetro	Descrição
service_hyq4gkb	ID do serviço de e-mail configurado no painel do EmailJS
visita_template	ID do template de e-mail criado no EmailJS, contendo o HTML e placeholders
templateParams	Objeto com os dados dinâmicos que substituem os placeholders no template.

Quadro 5 – Descrição da estrutura da função emailjs.send().

Fonte: Do Autor

4.2.4 Interfaces

Nessa seção serão apresentadas as telas do sistema.

4.2.4.1 Telas do sistema

Ao acessar o sistema, o usuário é recebido pela página inicial, que oferece um texto de boas-vindas sobre o projeto. Para facilitar a navegação, uma barra lateral exibe o menu com todas as funcionalidades disponíveis. A Figura 17 representa a tela inicial padrão, enquanto a Figura 18 apresenta a mesma tela, porém com o menu expandido, mostrando as opções de interação. Como é uma página pública, o usuário não está logado.

Figura 17 – Imagem da tela inicial do sistema.

Fonte: Do Autor

Figura 18 – Imagem da tela inicial do sistema com menu expandido

Fonte: Do Autor

Ao selecionar a opção "Fale Conosco"no menu principal, o usuário é redirecionado para uma tela que centraliza as informações de contato do projeto. Esta interface disponibiliza o endereço físico, o telefone de contato, o e-mail e as redes sociais do projeto. A Figura 19 ilustra a apresentação dessas informações.

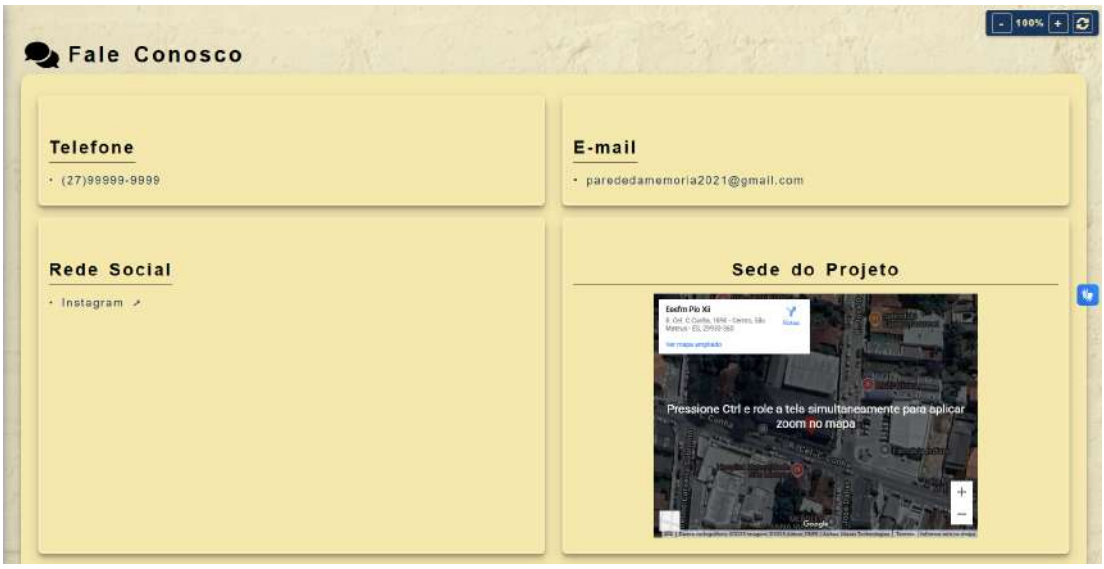

Figura 19 – Imagem da tela "Fale Conosco"com informações de contato

Fonte: Do Autor

Além disso, a tela permite que o usuário preencha e envie um formulário aos gestores do sistema, com o objetivo de solicitar uma visita à localização física do projeto "Parede da Memória", conforme detalhado na Figura 20.

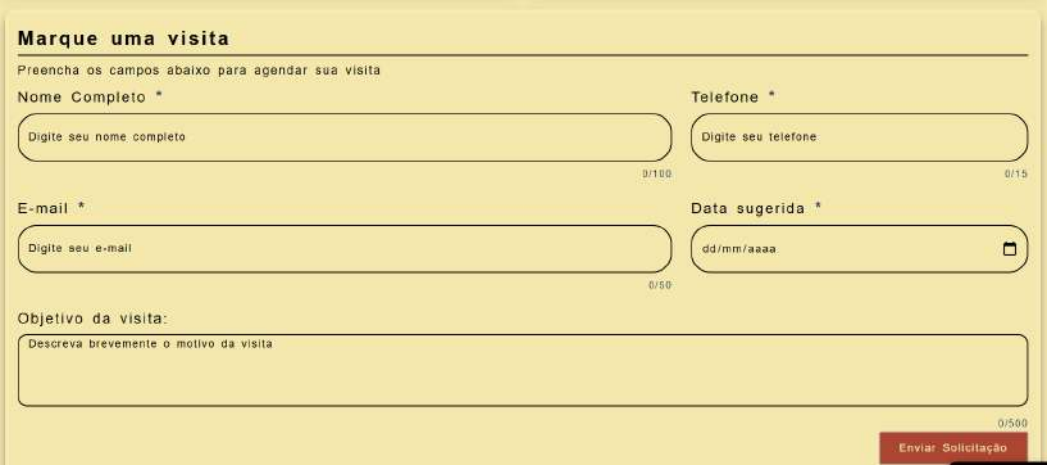

Figura 20 – Imagem da tela "Fale Conosco"com formulário de visita

Fonte: Do Autor

Após o envio da solicitação, o sistema confirma a operação com uma mensagem de sucesso, ilustrada na Figura 21.

Figura 21 – Imagem da tela da modal apresentando a mensagem de sucesso ao enviar solicitação de agendamento

Fonte: Do Autor

Simultaneamente, um e-mail contendo os dados preenchidos é automaticamente enviado para o endereço de e-mail padrão do projeto, conforme detalhado na Figura 22.

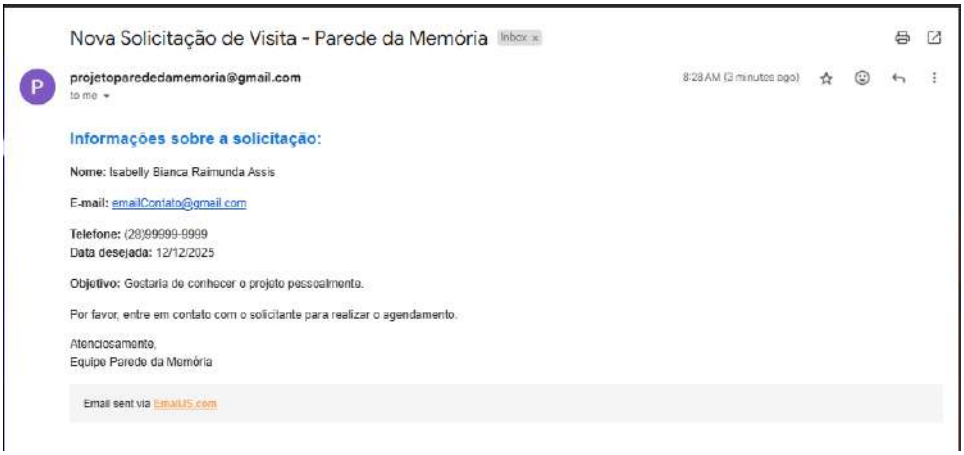

Figura 22 – Imagem do e-mail enviado pelo sistema ao realizar solicitação de agendamento.

Fonte: Do Autor

Ao acessar a seção "Nossos Projetos"no menu principal, o usuário é direcionado a uma tela que exibe uma listagem completa de todos os projetos cadastrados no sistema. Cada projeto é apresentado com seu respectivo nome, logomarca e resumo. A navegação por essa listagem é feita através da barra de rolagem ou pela utilização de links de atalho localizados na parte superior da página. A Figura 23 ilustra a interface dessa tela.

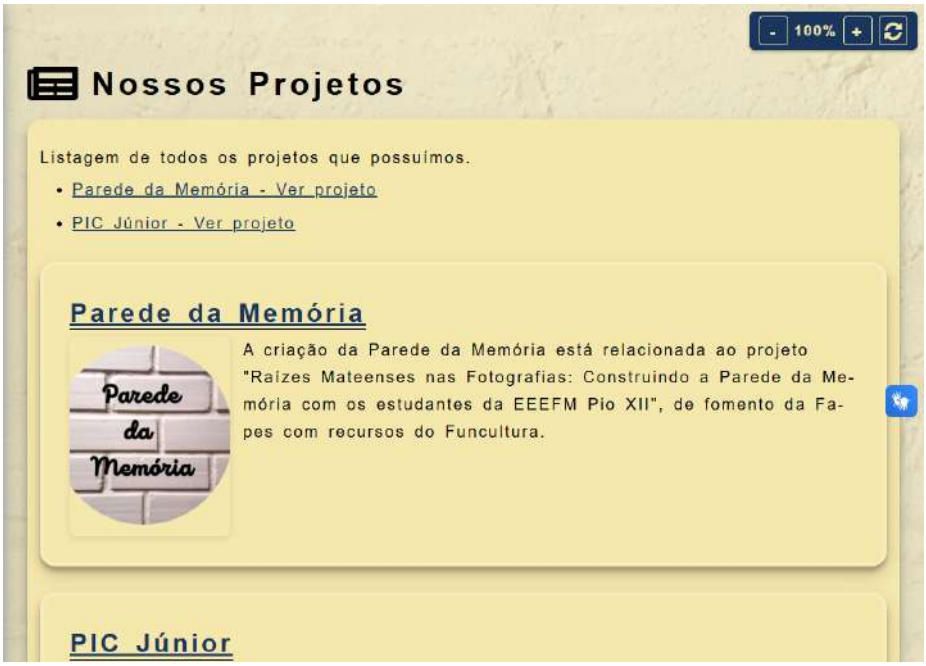

Figura 23 – Imagem da tela de listagens dos projetos

Fonte: Do Autor

Ao clicar no nome de um projeto, o usuário é direcionado para a tela que exibe as publicações vinculadas a ele. Nesta tela, é possível visualizar a descrição completa do

projeto, filtrar as publicações por nome ou localização e acessar uma listagem detalhada que inclui: nome, foto principal, autor da foto, data da memória e localização. A navegação nessa listagem pode ser realizada por meio da barra de rolagem ou pela aplicação dos filtros de pesquisa, conforme ilustrado na Figura 24.

Figura 24 – Imagem da tela de listagens de publicações

Fonte: Do Autor

Ao selecionar o nome de uma publicação, o usuário é direcionado à tela de detalhes da publicação. Esta tela permite a visualização do nome, autor, data, localização e descrição completa da imagem. Caso a publicação contenha múltiplas imagens, a navegação entre elas é facilitada por botões de “foto anterior” e “próxima foto”.

A Figura 25 ilustra a apresentação dessa tela.

Figura 25 – Imagem da tela de detalhes da publicação

Fonte: Do Autor

Ao selecionar a opção "Sobre o Projeto" no menu principal, o usuário é redirecionado para uma tela (apresentada na Figura 26) que contém um texto explicativo abrangente. Este texto aborda a criação e os objetivos do projeto, fornecendo informações essenciais sobre sua concepção e finalidade.

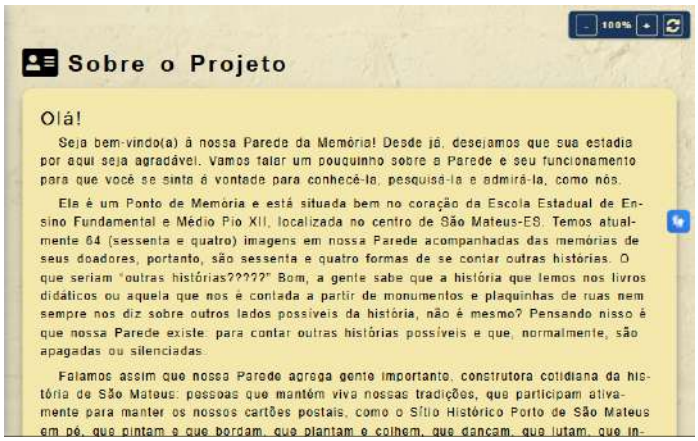

Figura 26 – Imagem da tela de informações sobre o projeto

Fonte: Do Autor

Ao selecionar a opção "Quem Somos" no menu principal, o sistema apresentará

uma listagem dos colaboradores associados aos projetos exibidos. Cada colaborador é detalhado com nome, foto de perfil, um breve relato e os projetos aos quais está vinculado. A navegação por essa listagem é realizada por meio de uma barra de rolagem ou pela utilização de links de atalho situados na parte superior da página. A Figura 27 ilustra a interface dessa tela.

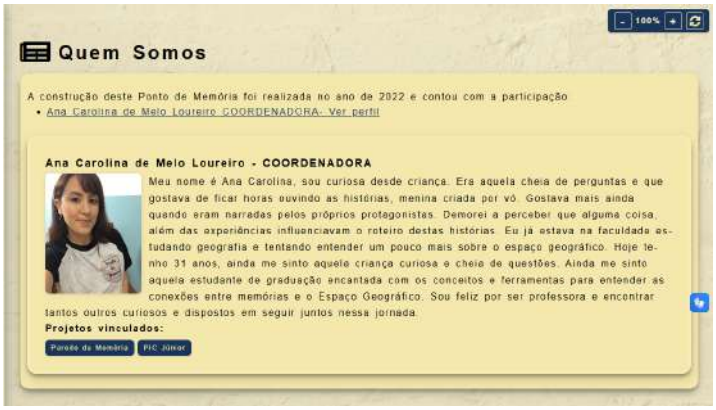

Figura 27 – Imagem da tela de listagem dos colaboradores

Fonte: Do Autor

Ao selecionar a opção "Login"no menu principal, o sistema redireciona o usuário para a tela de login. Nela, são disponibilizados os campos “Nome do usuário” e “Senha de acesso” para a autenticação. A Figura 28 apresenta a interface da tela de login do sistema.

Figura 28 – Imagem da tela de login do sistema

Fonte: Do Autor

É importante destacar que somente os usuários com privilégios administrativos terão um login, garantindo acesso a funcionalidades adicionais para a gestão do projeto.

Após realizar o login corretamente, o sistema exibe uma mensagem de sucesso, conforme apresentado na Figura 29 e o usuário é redirecionado para a tela inicial.

Figura 29 – Imagem da tela da modal apresentando a mensagem de sucesso ao logar.

Fonte: Do Autor

Após a autenticação bem-sucedida, o usuário que tem acesso como administrador master obtém acesso aos recursos de gerenciamento e administração do sistema, visualizando as funcionalidades correspondentes. A Figura 30 ilustra os botões dedicados à edição dos itens de contato, que incluem Telefone, E-mail e Rede Social.

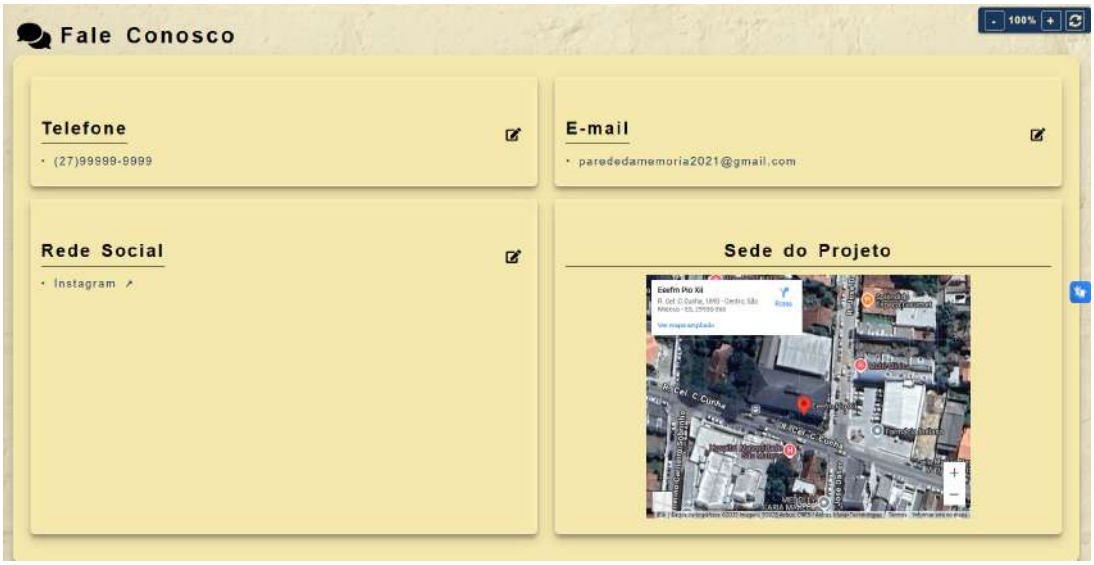

Figura 30 – Imagem da tela “Fale Conosco” com opção de edição.

Fonte: Do Autor

As Figuras 31 e 32 complementam essa visualização, apresentando os respectivos campos de preenchimento e edição para cada um desses itens.

Figura 31 – Imagem da tela "Fale Conosco" com os campos de Telefone e E-mail.

Fonte: Do Autor

Figura 32 – Imagem da tela "Fale Conosco" com os campos de Rede Social

Fonte: Do Autor

Após autenticação, o usuário com perfil de administrador master pode acessar a aba 'Nossos Projetos', onde lhe é concedida a funcionalidade de gerenciar os projetos do sistema. Isso inclui a capacidade de adicionar novos projetos por meio do botão 'Adicionar Projeto', e também realizar alterações ou excluir projetos existentes utilizando os botões de edição e exclusão, respectivamente.

A Figura 33 ilustra a interface dessa tela administrativa.

Figura 33 – Imagem da tela "Nossos Projetos" com acesso de administrador master.

Fonte: Do Autor

Ao selecionar a opção para adicionar um novo projeto, o usuário é direcionado a uma interface que apresenta um formulário. Nele, são apresentados os campos necessários para o cadastro e criação de um projeto no sistema, conforme ilustrado na Figura 34.

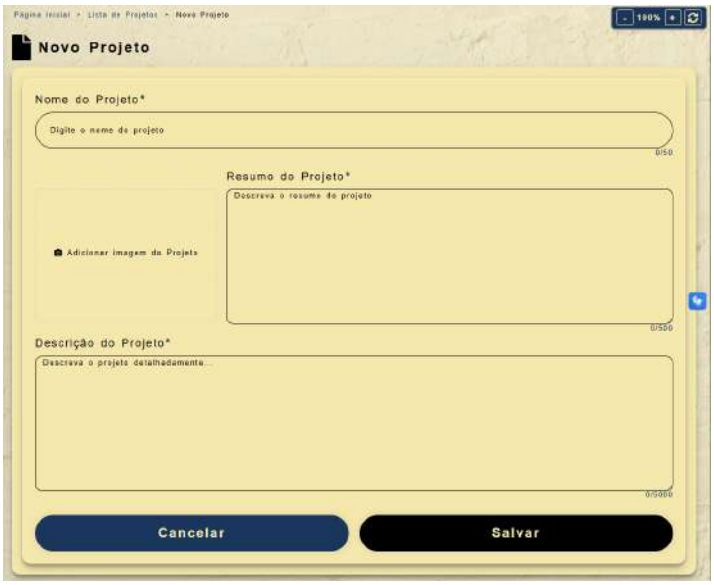

Figura 34 – Imagem da tela para adicionar novo projeto.

Fonte: Do Autor

Em complemento, a Figura 35 ilustra a tela de edição, onde os campos estão preenchidos com os dados atuais e são editáveis.

Figura 35 – Imagem da tela para editar projeto.

Fonte: Do Autor

A Figura 36 mostra uma modal de confirmação para o caso do usuário escolher excluir o projeto.

Figura 36 – Imagem da tela da modal de confirmação de exclusão.

Fonte: Do Autor

No caso de o login ter sido realizado por um coadministrador ou um administrador master, ambos os perfis terão a permissão de adicionar novas publicações à lista de projetos. Essa funcionalidade é acessível por meio do botão 'Adicionar Publicação', conforme ilustrado na Figura 37.

Figura 37 – Imagem da tela "Publicações" com acesso de administrador master e/ou coadministrador.

Fonte: Do Autor

Ao selecionar essa opção, o usuário é redirecionado para a tela apresentada na Figura 38, onde são exibidos os campos para preenchimento.

Figura 38 – Imagem da tela de adicionar publicação.

Fonte: Do Autor

Para o perfil do tipo administrador, é apresentado as opções de editar ou exluir as publicações, conforme apresentado na Figura 39.

Figura 39 – Imagem da tela de visualização de publicação com acesso de administrador

Fonte: Do Autor

Com o login de administrador master, torna-se possível acessar a aba “Quem Somos” com permissões gerenciais. Nessa seção, o administrador pode realizar ações como adicionar novos colaboradores, editar informações já existentes ou excluir registros, conforme necessário.

A Figura 40 ilustra a interface apresentada ao administrador master, destacando os campos e funcionalidades disponíveis para a gestão do conteúdo da página “Quem Somos”.

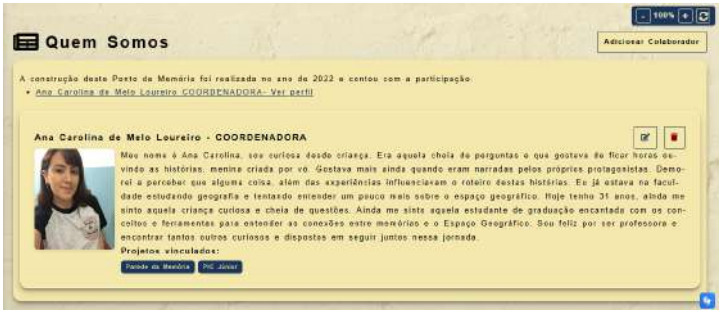

Figura 40 – Imagem da tela de visualização de "Quem somos"com acesso de administrador

Fonte: Do Autor

As Figuras 41 e 42 apresentam, respectivamente, as interfaces utilizadas para a adição de um novo colaborador e para a edição de dados de um colaborador já cadastrado

no sistema.

Figura 41 – Imagem da tela de adicionar colaborador.

Fonte: Do Autor

Figura 42 – Imagem da tela para editar colaborador.

Fonte: Do Autor

Como administrador master, o usuário possui acesso exclusivo à interface de "Agendamentos de Visitas" no menu principal, uma funcionalidade destinada à organização e ao controle das visitas solicitadas por meio do formulário "Fale Conosco". Essa seção foi

desenvolvida para centralizar, acompanhar e registrar as intenções de visita informadas pelos usuários da plataforma, permitindo uma gestão dessas demandas.

A Figura 43 ilustra a tela correspondente, na qual é possível visualizar todas as solicitações recebidas. Cada entrada apresenta os dados fornecidos pelos usuários, como nome, e-mail, telefone e mensagem, permitindo ao administrador master avaliar cada pedido. Essa triagem é essencial para assegurar um controle mais rigoroso sobre os agendamentos, além de contribuir para a organização interna da equipe responsável.

Ao acessar essa interface, o administrador master pode selecionar a data desejada para o atendimento no campo "Marcar um dia". Uma vez escolhida, a data é automaticamente registrada no sistema, funcionando como referência interna para a realização da visita.

Figura 43 – Imagem da tela de gerenciamento dos agendamentos de visitas

Fonte: Do Autor

É fundamental destacar que o sistema não realiza o envio automático de notificações ou confirmações ao solicitante. Toda e qualquer comunicação relacionada à visita deve ser feita manualmente pelos administradores, utilizando canais externos como e-mail, telefone ou outros meios disponíveis. O sistema serve, portanto, exclusivamente como ferramenta de controle interno, centralizando as informações e organizando os dados, mas sem executar

interações diretas com o usuário final.

As Figuras 44 e 45 ilustram, respectivamente, as interfaces das janelas modais de confirmação utilizadas nos processos de agendamento e rejeição de solicitações de visita. Essas modais são exibidas ao administrador master como uma etapa intermediária e essencial para garantir a confirmação da ação a ser realizada, evitando alterações acidentais ou decisões equivocadas.

Figura 44 – Imagem da tela da modal de confirmação de agendamento de visita.

Fonte: Do Autor

Figura 45 – Imagem da tela da modal de confirmação de exclusão.

Fonte: Do Autor

Na tela de "Usuários"no menu principal, o administrador master tem acesso à lista completa dos usuários cadastrados no sistema, que são, por definição, os coadministradores. A Figura 46 ilustra essa interface.

Figura 46 – Imagem da tela de gerenciamento dos usuários

Fonte: Do Autor

Na Figura 47 é apresentado a mensagem de confirmação que é solicitada ao administrador master ao selecionar a opção de excluir um usuário.

Figura 47 – Imagem da tela da modal de confirmação de exclusão

Fonte: Do Autor

Ao selecionar a opção “Adicionar usuário” no canto superior direito da página, o administrador master tem acesso à funcionalidade de cadastro de novos usuários no sistema. Essa ação permite incluir colaboradores, conforme Figura 48.

Figura 48 – Imagem da tela de adicionar usuário

Fonte: Do Autor

Ao clicar na opção “Sair” do sistema no menu principal, conforme Figura 48 , o usuário é desconectado e redirecionado para a página de login (Figura 28).

As demais Figuras de telas do sistema não exibidas nesse capítulo são apresentadas no apêndice D.

4.3 Acessibilidade

A construção de um sistema web inclusivo é fundamental para garantir que ele possa ser utilizado pelo maior número possível de pessoas, independentemente de suas habilidades ou das tecnologias assistivas que utilizam. No desenvolvimento do sistema para o projeto "Parede da Memória", a acessibilidade foi abordada como um requisito fundamental, integrada em diversas etapas do processo para remover barreiras e promover uma melhor experiência a todos os usuários. As seguintes práticas foram implementadas para garantir a conformidade com princípios de acessibilidade.

4.3.1 Declaração do Idioma da Página

Para garantir que o conteúdo do sistema seja corretamente interpretado por tecnologias assistivas e por mecanismos de busca, o idioma principal da página foi explicitamente declarado. Essa prática, embora muitas vezes subestimada, é um pilar fundamental da acessibilidade.

A declaração do idioma, feita geralmente através do atributo lang na tag *<html>*

(ex: *<html lang="pt-BR»*), permite que leitores de tela ajustem a pronúncia, o sotaque e as regras gramaticais de acordo com o idioma especificado. Sem essa informação, um leitor de tela poderia tentar ler o texto em português com as regras de outro idioma, tornando a experiência confusa ou incompreensível para o usuário com deficiência visual. No código abaixo é possível visualizar a implementação dessa tag no arquivo *index.html* do projeto.

```
<!doctype html>
<html lang="pt-br" class="main">
  <head>
```

4.3.2 Alinhamento de textos

Para assegurar clareza e acessibilidade, o texto do sistema foi configurado predominantemente alinhado à esquerda (ex: *left* ou *start*). Em algumas seções específicas, o alinhamento foi ajustado conforme solicitação do cliente (ex.: centralizado ou justificado). Conforme ilustrado no código abaixo:

```
.breadcrumbs {
  margin-top: 1rem;
  font-size: 1rem;
  color: #000000;
  text-align: start;
}

.subtitles-home {
    text-align: left;
    text-shadow: none;
    width: 100%;
    padding: 0 0.8rem 0 0.8rem;
}
```

4.3.3 Navegação por Teclado e Foco Visual

Um dos atributos da acessibilidade para usuários que dependem de teclado (seja por deficiência motora, uso de leitores de tela ou preferência pessoal) é a capacidade de navegar e interagir com todos os elementos da interface sem o uso de um mouse.

- Implementação: Todos os elementos interativos do sistema, como botões, links, campos de formulário e elementos de navegação (menus), foram desenvolvidos para serem acessíveis via teclado. Isso significa que o usuário pode percorrer a interface usando as teclas Tab e Shift + Tab para navegar entre os elementos.

- Indicação de Foco Visível: Para complementar a navegação por teclado, foi assegurado que todos os elementos interativos exibam um indicador de foco visual claro ao serem selecionados. Esse destaque, realizado por meio de alterações visuais como mudança de cor e aumento do tamanho do elemento, permite ao usuário identificar com precisão o componente atualmente em foco, facilitando a interação e reduzindo o risco de desorientação durante o uso do sistema.

Na Figura 49, apresenta-se um exemplo do comportamento do sistema durante a navegação por teclado.

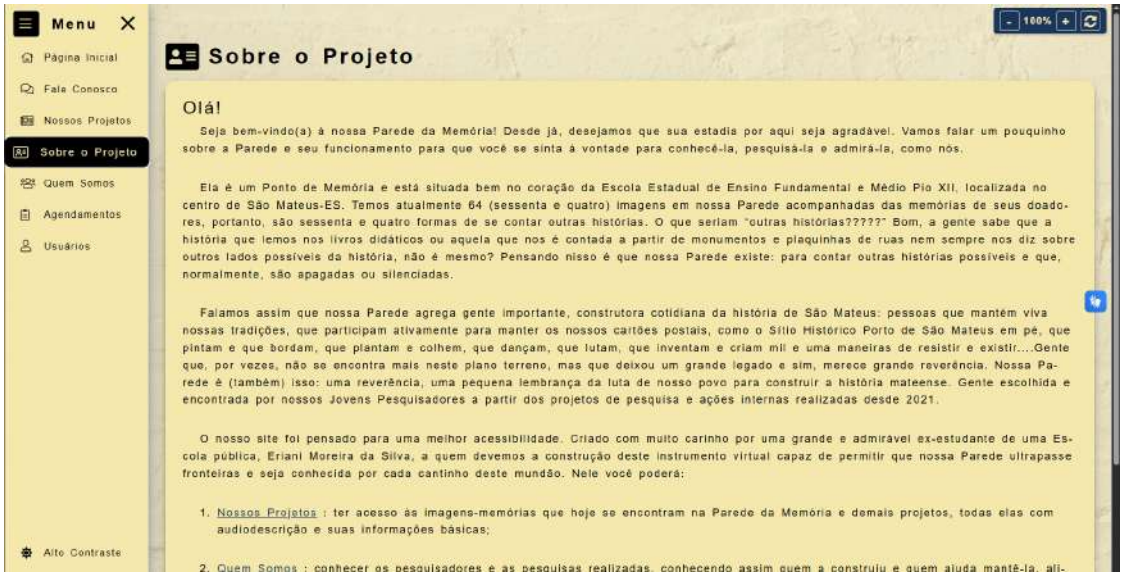

Figura 49 – Imagem da tela do menu principal apresentando foco ao navegar utilizando o teclado.

Fonte: Do Autor

Observa-se que, ao receber o foco, o item de menu principal “Sobre o Projeto” sofre alterações visuais, como a mudança de cor e o aumento do tamanho do elemento, indicando de forma clara que está selecionado. Esse comportamento também é reproduzido quando o usuário passa o cursor do mouse sobre o elemento, garantindo uma experiência visual consistente independentemente do modo de navegação utilizado.

4.3.4 Texto Alternativo para Imagens (Alt Text)

Imagens são componentes visuais importantes no projeto "Parede da Memória", por se tratar de um acervo de fotos (memórias), mas para usuários com deficiência visual que utilizam leitores de tela, elas não são compreendidas sem uma descrição textual.

- Implementação: Todas as imagens presentes no sistema (incluindo logotipos, ícones e imagens de conteúdo) foram dotadas de atributos *alt* descritivos. Este texto

alternativo permite que leitores de tela transmitam o conteúdo ou a função da imagem aos usuários, garantindo que a informação visual não seja perdida.

Na Figura 50, é possível visualizar o texto alternativo da imagem de logotipo de um projeto.

Figura 50 – Imagem da tela de projetos apresentando texto alternativo para logotipo.

Fonte: Do Autor

4.3.5 Estrutura Semântica do HTML

O uso correto das tags HTML não é apenas uma boa prática de desenvolvimento, mas um pilar fundamental para a acessibilidade, pois fornece uma estrutura lógica e hierárquica para o conteúdo.

- Implementação: Foram utilizadas tags HTML semânticas (*<nav>, <main>, <section>, <h1-h6>, <button>, <input>*, etc.) de forma apropriada. Isso permite que tecnologias assistivas, como leitores de tela, interpretem a função de cada seção da página e ajudem os usuários a navegar de forma mais eficiente e a compreender a hierarquia do conteúdo.

Na Figura 51, é possível visualizar um exemplo dessa hierarquia ao apresentar a tela da Página Principal, onde se tem a utilização de títulos *<h1>* a *<h6>* em ordem correta (sem pular níveis) ajudando na compreensão da estrutura da página.

Figura 51 – Imagem da tela de projetos apresentando hierarquia da página.

Fonte: Do Autor

4.3.6 Design Responsivo

Embora não seja estritamente uma diretriz de acessibilidade para deficiências, um *design responsivo* impacta diretamente a usabilidade e acessibilidade em diferentes dispositivos e tamanhos de tela. A Figura 52 ilustra a responsividade do sistema ao visualizar em telas mobiles.

Figura 52 – Imagem da tela principal na versão mobile.

Fonte: Do Autor

O sistema foi desenvolvido com um *design responsivo*, utilizando técnicas de CSS para garantir que a interface se adapte e seja totalmente funcional em diferentes tamanhos de tela, desde desktops até dispositivos móveis. Isso beneficia usuários que acessam o sistema em diversos contextos, incluindo aqueles que podem usar zooms de tela ou dispositivos com displays menores.

4.3.7 Rotulagem de Formulários (Labels)

Campos de formulário sem rótulos associados são inacessíveis para leitores de tela, que não conseguem identificar a finalidade do campo.

Todos os campos de formulário no sistema (*<input>*, *<textarea>*, *<select>*) foram associados a um elemento *<label>* correspondente, utilizando o atributo *for* vinculado ao *id* do campo. Isso garante que, quando um leitor de tela foca em um campo de formulário, ele possa anunciar a finalidade do campo ao usuário (ex: "Nome do usuário", "Senha de

acesso"), facilitando a interação e o preenchimento. No código abaixo é representando um exemplo dessa aplicação no campo “User de acesso” que é preenchido ao cadastrar um usuário no sistema.

```
<div class="input-box">
  <label for="username" id="label-username">
    User de acesso <span class="required-asterisk">*</span>
  </label>
  <input
    autocomplete="username"
    id="username"
    type="text"
    placeholder="Digite seu user"
    name="username"
    [(ngModel)]="createUser.user"
    required
    (input)="limitCharacterCount($event, 'user')"
    #userInput="ngModel"
    aria-labelledby="label-username"
    aria-required="true"
    aria-describedby="username-char-counter"
  />
  <small class="char-counter" id="username-char-counter"
    >{{ createUser.user.length || 0 }}/45</small
  >
  @if (userInput.touched && userInput.errors?.["required"]) {
    <div id="error-username" class="field-error" aria-live="assertive">
      Campo user de acesso éobrigatório
    </div>
  }
```

4.3.8 Mensagens de Status Acessíveis

A clareza e a acessibilidade das mensagens de status são cruciais para garantir que os usuários recebam feedback imediato e compreensível sobre suas ações, especialmente após interações que resultam em alteração da rota de navegação. Para isso, o sistema foi projetado para fornecer mensagens de status acessíveis, assegurando que todos os usuários, incluindo aqueles que utilizam tecnologias assistivas, sejam devidamente informados.

Por exemplo, após um login bem-sucedido, o sistema exibe uma mensagem de sucesso clara, que não apenas confirma a ação, mas também informa sobre o redirecionamento de tela que será realizado.

A Figura 53 ilustra essa mensagem. Essa abordagem garante que usuários com deficiência visual, que dependem de leitores de tela, compreendam a transição de contexto, contribuindo para uma experiência de navegação mais fluida e intuitiva.

Figura 53 – Imagem da tela da modal apresentando a mensagem clara de sucesso ao logar.

Fonte: Do Autor

4.3.9 Funcionalidade de Zoom: Acessibilidade e Flexibilidade Visual

Para aprimorar a acessibilidade e adaptar a experiência visual às necessidades individuais dos usuários, o sistema oferece a funcionalidade de zoom. Essa capacidade permite que os usuários aumentem ou diminuam o tamanho do conteúdo exibido, conforme sua preferência ou necessidade. A Figura 54 ilustra um exemplo para o sistema onde a tela de Página Inicial do menu principal está com o zoom aplicado, através do recurso disponibilizado no canto superior direito.

Figura 54 – Imagem da tela principal com o zoom aplicado.

Fonte: Do Autor

A possibilidade de dar zoom é crucial para diversos grupos de usuários. Para pessoas

com baixa visão, por exemplo, a ampliação do texto e das imagens torna o conteúdo legível sem a necessidade de ferramentas externas ou de configuração complexa do navegador. Da mesma forma, usuários que buscam maior conforto visual ou que utilizam dispositivos com diferentes resoluções de tela podem ajustar o nível de zoom para otimizar a visualização.

É importante ressaltar que, embora a maioria dos navegadores já ofereça a funcionalidade de zoom por meio de atalhos como Ctrl + (aumentar) e Ctrl - (diminuir), a inclusão de um botão de zoom dedicado no sistema fornece uma possibilidade adicional e mais explícita para o usuário realizar essa ação. Isso garante que a funcionalidade seja facilmente detectável e acessível, mesmo para aqueles que não estão familiarizados com os atalhos de teclado do navegador, promovendo uma navegação mais inclusiva e personalizada.

4.3.10 Confirmação para Alterações Críticas: Prevenção de Erros Irreversíveis

Para proteger as informações e evitar que algo seja excluído ou alterado por engano, o sistema pede uma confirmação antes de realizar ações importantes. Isso garante que o usuário tenha certeza do que está fazendo.

Ao tentar, por exemplo, excluir um usuário, o sistema não procede com a exclusão imediatamente. Em vez disso, uma janela de confirmação é apresentada, como ilustrado na Figura 55.

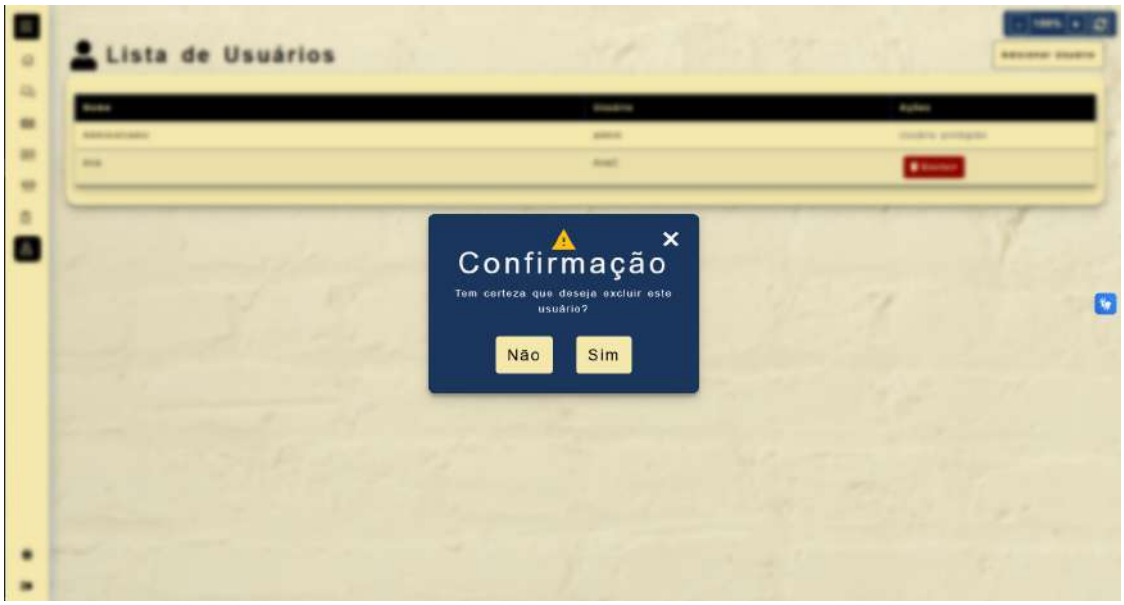

Figura 55 – Imagem da tela da modal solicitando a confirmação antes de excluir usuário.

Fonte: Do Autor

Essa etapa adicional serve como uma barreira de segurança e um ponto de validação que solicita ao usuário a confirmação de sua intenção antes que a ação seja efetivada. Essa funcionalidade é crucial para a acessibilidade, pois minimiza a chance de erros para

usuários que podem ter dificuldades motoras, visuais ou cognitivas, que poderiam, de outra forma, clicar acidentalmente. Ao exigir uma confirmação explícita, o sistema não só evita perdas de dados indesejadas, mas também proporciona aos usuários maior controle e confiança ao interagir com funcionalidades que envolvem alterações significativas.

4.3.11 Recuperação de dados após logout inesperado

Como forma de tornar a experiência mais acessível, confiável e inclusiva, foi implementada no sistema uma funcionalidade que permite recuperar automaticamente o último rascunho salvo ao realizar login. Esse recurso é especialmente útil em situações em que o usuário é desconectado inesperadamente, seja por inatividade, problemas de rede ou fechamento acidental do navegador.

A Figura 56 ilustra a interface da modal exibida ao realizar login, informando sobre a existência de um rascunho anterior e permitindo sua recuperação com apenas um clique.

Figura 56 – Imagem da modal de recuperação de dados.

Fonte: Do Autor

Essa funcionalidade contribui diretamente para a acessibilidade ao reduzir barreiras relacionadas à perda de informações, especialmente para usuários que possam ter limitações motoras, cognitivas ou dificuldades com procedimentos técnicos mais complexos. A recuperação automática evita frustrações, economiza tempo e garante que esforços anteriores não sejam perdidos, promovendo uma experiência mais estável e acolhedora.

4.3.12 Contraste de Cores

Para assegurar a legibilidade e a inclusão, o contraste entre o texto e o plano de fundo foi uma preocupação central no desenvolvimento, sendo vital para usuários com

baixa visão e daltonismo. Desde o levantamento inicial do sistema, a seleção da paleta de cores levou em conta a necessidade de um contraste adequado, alinhando-se às expectativas do cliente. A validação desse aspecto, por meio de testes específicos, será apresentada em detalhes na Seção 5.

Para exemplificar o nível de contraste adequado, são considerados os critérios de sucesso definidos pela WCAG relacionados à acessibilidade visual. A Figura 57 apresenta, de forma ilustrativa, como os diferentes níveis de contraste podem ser classificados, evidenciando aqueles que melhor atendem às recomendações para garantir uma leitura mais acessível.

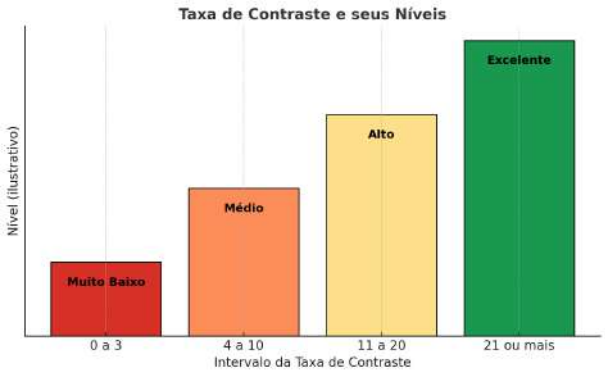

Figura 57 – Taxa de Contraste e seus níveis.

Fonte: Do Autor

Além da paleta de cores original apresentar nível de contraste alto, foi implementado um botão de alternância de contraste no sistema. Ao ser ativado, toda a aplicação passa a utilizar uma combinação de cores com contraste ainda mais elevado, utilizando predominantemente preto e branco e tendo nível excelente. Essa funcionalidade pode ser observada através do botão "Alto Contraste", na Figura 58.

Figura 58 – Imagem da tela principal com alterações de contraste.

Fonte: Do Autor

4.3.13 Integração com VLibras: Acessibilidade para a Comunidade Surda

Para ampliar ainda mais o alcance e a inclusão do sistema, foi implementada a integração com a ferramenta VLibras. Essa integração é um passo fundamental para garantir a acessibilidade da informação à comunidade surda brasileira e pessoas com deficiência auditiva, que se comunicam predominantemente por meio da Língua Brasileira de Sinais (Libras).[3]

A Figura 59 ilustra a utilização do VLibras no site desenvolvido.

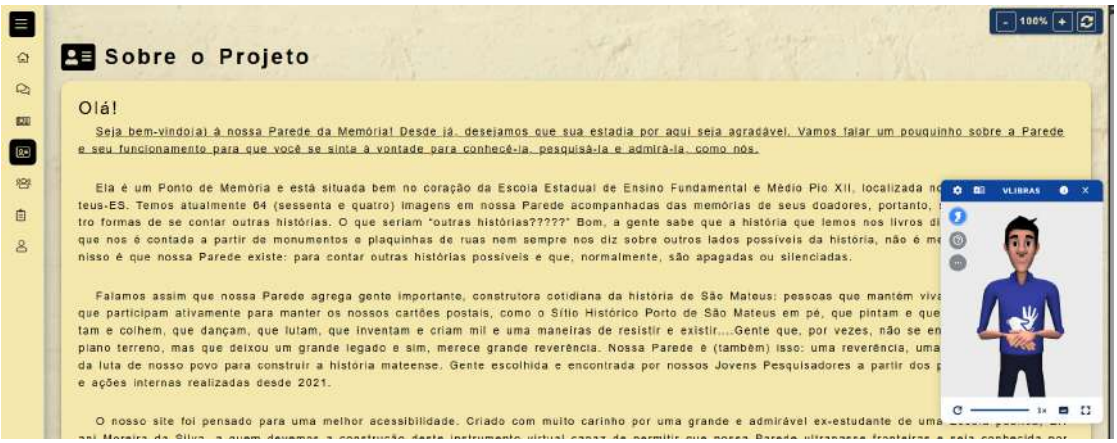

Figura 59 – Imagem da tela “Sobre o Projeto” com interação do VLibras.

Fonte: Do Autor

O VLibras é uma suíte de ferramentas gratuitas que traduz automaticamente conteúdos digitais (texto, áudio e vídeo) para Libras, apresentando um intérprete virtual em 3D (BRASIL, 2024). Ao adicionar essa funcionalidade ao sistema, os usuários com deficiência auditiva possuem acesso ao conteúdo do site de forma mais compreensível e autônoma. O intérprete de Libras aparece na tela, traduzindo em tempo real o texto da página, eliminando barreiras de comunicação e promovendo a inclusão digital para esse importante grupo de usuários.

4.3.14 Botão "Retornar ao Topo": Navegação Acessível e Facilitada

Para aprimorar a experiência de navegação em páginas com conteúdo extenso, foi adicionado, conforme apresentado na Figura 60, um botão "Retornar ao Topo". Essa funcionalidade oferece uma maneira rápida, eficiente e acessível para os usuários voltarem ao início da página sem a necessidade de rolagem manual prolongada.

[3]Mais informações em <https://www.gov.br/ines/pt-br>

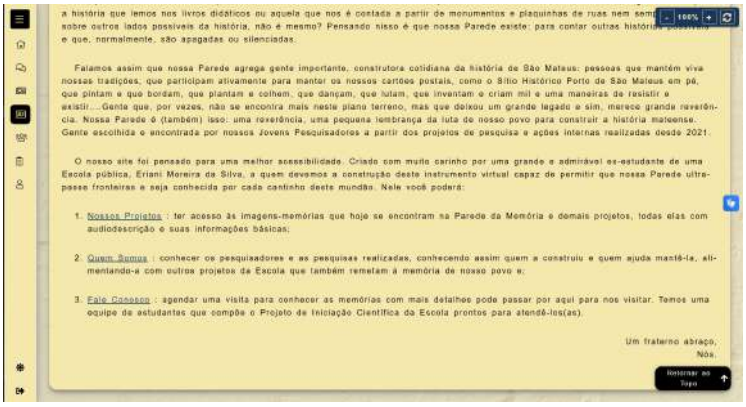

Figura 60 – Imagem da tela “Sobre o Projeto” com botão de "Retornar ao topo".

Fonte: Do Autor

A presença de um botão visível e facilmente acessível, que pode ser ativado tanto por mouse quanto por navegação via teclado, é um recurso importante de acessibilidade. Ele beneficia especialmente usuários com deficiência motora, que podem ter dificuldade em realizar movimentos de rolagem repetitivos, e usuários de leitores de tela, que podem perder a noção da posição na página ao navegar por grandes volumes de texto. Ao proporcionar um atalho claro e detectável para retornar ao topo, o sistema garante uma navegação mais intuitiva, confortável e inclusiva para todos.

4.3.15 Modal Acessível: Foco Preso e Fechamento por Teclado

A implementação de modais (*pop-ups*) exige atenção redobrada à acessibilidade, visto que podem interromper o fluxo de navegação e desorientar usuários de tecnologias assistivas. Para garantir uma experiência inclusiva, o sistema incorpora modais que seguem as melhores práticas de acessibilidade.

Primordialmente, ao abrir um modal, o foco do teclado é automaticamente direcionado para o seu conteúdo, e é preso dentro dele. Isso significa que, enquanto o modal estiver ativo, a navegação via teclado não permitirá que o usuário saia dos elementos interativos dentro do modal, evitando que o foco “escape” para o conteúdo de fundo. Essa funcionalidade é crucial para que usuários de leitores de tela e de navegação por teclado não se percam na interface, garantindo que interajam apenas com o modal até que ele seja fechado.

Além disso, a usabilidade e a acessibilidade são aprimoradas pela capacidade de fechar o modal tanto clicando no ícone de fechar quanto utilizando a tecla Esc. Essa dupla funcionalidade oferece uma maneira intuitiva e eficiente para os usuários dispensarem o modal sem a necessidade exclusiva de um mouse ou de localizar um botão “Fechar” na tela, reforçando a navegação por teclado e a flexibilidade de interação para todos.

4.3.16 Navegação por Atalhos: Acesso Rápido a Conteúdos Repetidos

Para otimizar a experiência de navegação em seções do site que apresentam listagens extensas com dados repetidos, como as telas “Quem Somos” e “Nossos Projetos”, foi implementada uma lista de links de atalho no topo da página. Essa funcionalidade permite que o usuário navegue diretamente para um item específico de interesse, sem a necessidade de percorrer toda a lista.

Ao invés de exigir rolagem contínua para localizar um determinado conteúdo, esses links de atalho oferecem um acesso rápido e direto. Isso melhora significativamente a usabilidade e a acessibilidade, especialmente para usuários que utilizam teclado ou tecnologias assistivas, pois reduz a quantidade de navegação necessária e torna a localização da informação desejada muito mais eficiente. É possível observar essa implementação nas Figuras 61 e 62.

Figura 61 – Imagem da tela “Quem Somos” com lista de itens disponível através de links.

Fonte: Do Autor

Figura 62 – Imagem da tela “Nossos projetos” com lista de itens disponível através de links.

Fonte: Do Autor

4.3.17 Uso de Breadcrumbs: Orientação e Navegação Hierárquica

Para fornecer aos usuários uma clara orientação sobre sua localização dentro da estrutura do site e facilitar a navegação, foi implementada a funcionalidade de *breadcrumbs* (ou "migalhas de pão") que pode ser visualizado na Figura 63.

Os *breadcrumbs* são um auxílio visual e de navegação que exibem o caminho hierárquico percorrido pelo usuário desde a página inicial até a página atual. Por exemplo, em uma página de publicação, o *breadcrumb* é "Página Inicial > Nossos Projetos > [Nome do Projeto] - Publicações > [Status]". Cada item do *breadcrumb* é um link clicável, permitindo que o usuário retorne facilmente a qualquer nível superior na hierarquia. Isso é especialmente útil em sites com muitas páginas e conteúdo aninhado, pois reduz a necessidade de memorizar o caminho e oferece uma maneira rápida e intuitiva de explorar outras seções relacionadas, melhorando a usabilidade e a experiência geral de navegação.

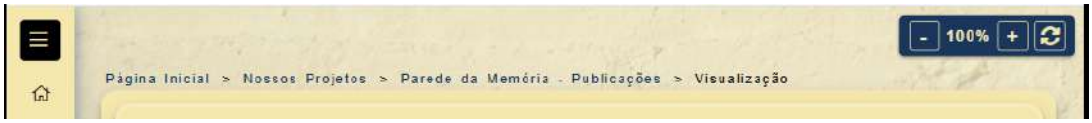

Figura 63 – Imagem da tela de visualização de publicação com foco no Breadcrumb.

Fonte: Do Autor

4.3.18 Títulos de Páginas Intuitivos: Clareza na Navegação

Cada página do sistema possui um título intuitivo e descritivo. O título da página, visível na barra de título do navegador e em abas, é a primeira informação que muitos usuários, especialmente aqueles que utilizam leitores de tela ou que possuem múltiplas abas abertas, consultam para entender o conteúdo atual e a finalidade da página. Dessa forma, é possível visualizar na Figura 64, através da página Nossos Projetos o título utilizado pelo navegador.

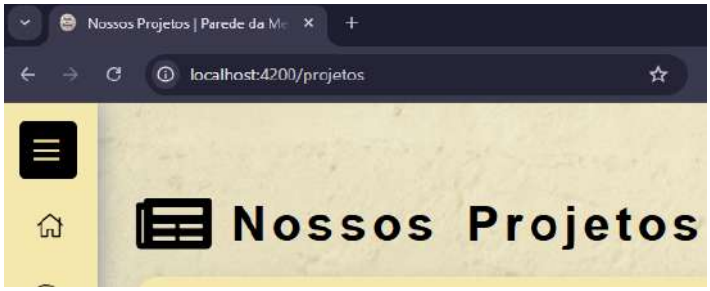

Figura 64 – Imagem da tela “Nossos Projetos” com aba do navegador descritiva.

Fonte: Do Autor

4.3.19 Uso de Atributos ARIA

Para garantir que o site fosse acessível a todos os usuários, especialmente aqueles que utilizam leitores de tela, foi feito o uso de atributos ARIA (Accessible Rich Internet Applications). Esses atributos funcionam como “etiquetas” adicionais, fornecendo informações extras sobre os elementos da interface, como sua função, estado atual e forma de

interação. Dessa forma, tecnologias assistivas podem interpretar o conteúdo de maneira mais precisa, permitindo uma navegação acessível e inclusiva.

No código apresentado, observa-se a aplicação prática desses atributos em elementos de navegação. O atributo *aria-label* foi utilizado nos links para fornecer um nome acessível que descreve claramente sua função, garantindo que leitores de tela transmitam a informação de forma compreensível aos usuários. Já o atributo *aria-current* foi empregado para indicar o estado do link em relação à página atual, auxiliando na orientação de navegação e permitindo que o usuário saiba, de forma inequívoca, qual página está sendo visualizada. A adoção desses atributos justifica-se pela necessidade de tornar a experiência de navegação mais clara e organizada para pessoas que dependem de tecnologias assistivas.

```
<nav class="breadcrumbs" aria-label="Navegacao estrutural">
  <a
    (click)="navigateWithConfirmation('/')"
    tabindex="0"
    role="link"
    aria-label="Retornar a Pagina Inicial"
    aria-current="false"
    >Pagina Inicial</a
  >
  &gt;
  <a
    (click)="navigateWithConfirmation('/projetos')"
    tabindex="0"
    role="link"
    aria-label="Retornar a Nossos Projetos"
    aria-current="false"
    >Nossos Projetos</a
  >
  &gt;
  <a
    (click)="voltar()"
    tabindex="0"
    role="link"
    aria-label="Retornar as publicacoes"
    aria-current="false"
    >{{ projetoNome }}</a
  >
  &gt;
  <span class="current-page" aria-current="location">
    {{
```

```
        modoVisualizacao
          ? "Visualizao  "
          : modoEdicao
            ? "Edicao"
            : "Nova Publicaco"
      }}
    </span>
  </nav>
```

5 Garantia de Qualidade e Processo de Testes

Para garantir que o sistema desenvolvido atenda plenamente aos requisitos de acessibilidade e funcionalidades propostas, foram elaborados cenários e metodologias de teste específicos, que serão detalhados nesta seção. Os testes realizados visam não apenas validar o cumprimento das especificações dos requisitos, mas também avaliar a inclusão, assegurando que os objetivos do projeto sejam alcançados de forma eficaz e acessível.

5.1 Testes com apoio de ferramentas

Nas seções a seguir, são apresentadas as validações realizadas no sistema com o auxílio de ferramentas automatizadas. As subseções incluem testes conduzidos com o uso de ferramentas como validadores, simuladores e extensões de acessibilidade.

5.1.1 Lighthouse

O Lighthouse[1] é uma ferramenta gratuita e open source que analisa páginas da web e sugere melhorias. Ele testa desempenho, acessibilidade, SEO e outros aspectos importantes, funcionando em diversas páginas e sites. Podendo ser usar de três formas: direto no Chrome DevTools, via linha de comando ou como módulo do Node.js (DEVELOPERS, 2025).

Por conter a funcionalidade de testar acessibilidade, todas as páginas do sistema passaram pelo teste usando o Lighthouse. Para calcular a pontuação de acessibilidade de um site através do Lighthouse é realizado uma média ponderada de todas as pontuações. Cada auditoria de acessibilidade é aprovada ou reprovada. Diferentemente das demais auditorias, a acessibilidade não é vista parcialmente. Na verificação 'Botões com identificação acessível', por exemplo, a presença de qualquer elemento em não conformidade resulta em falha completa da auditoria - mesmo que outros componentes estejam corretamente rotulados com atributos ARIA (DEVELOPERS, 2019).

Para realizar os testes, foram abertas todas as páginas no navegador Chrome e pressionado F12 no teclado, abrindo assim o Chrome DevTools, navegado até a aba 'Lighthouse', conforme apresentado na Figura 65. E após selecionar a opção "Analisar estado da página", a ferramenta realizou todo o processamento e gerou o relatório de testes, documentados no Quadro 6.

[1] Disponível através do navegador Chrome

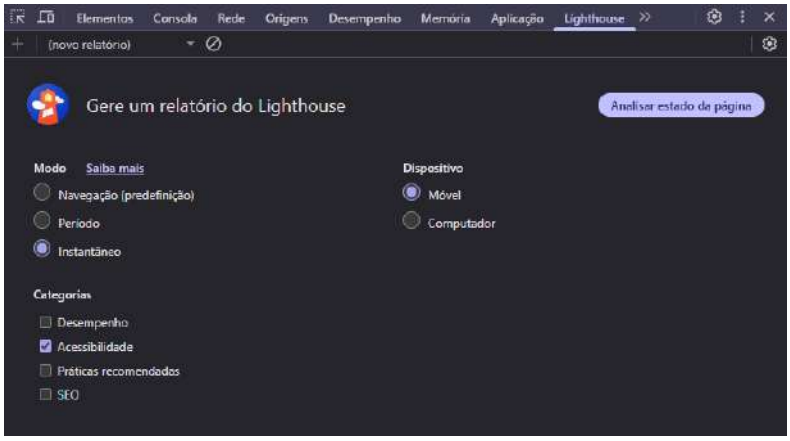

Figura 65 – Tela do Lighthouse

Fonte: Do Autor

Descrição do defeito	Solução	Estado
Botões sem identificação acessível	Implementado atributo *aria-label* em componentes de botão da interface	Corrigido
Links sem identificação acessível	Implementado atributo *aria-label* em componentes de link da interface	Corrigido
Configuração user-scalable="no"impede que os usuários deem zoom na página, e maximum-scale="1"limita o aumento do conteúdo.	Removido a restrição de zoom e aumentado o limite de ampliação de 1 para 5	Corrigido
A hierarquia de itens de lista (* *) estava incorreta.	Os itens de lista ((* *) não estavam contidos nos elementos superiores (* *), sendo assim foi feito essa adição	Corrigido
As áreas de toque não têm tamanho nem espaçamento suficientes.	Aumentado o tamanho de ícones	Corrigido

Quadro 6 – Descrição de resultados dos testes usando Lighthouse.

Fonte: Do Autor

Os relatórios de testes, apresentados no quadro, foram gerados utilizando os seguintes parâmetros:

- **Modo: Instantâneo** — refere-se a uma execução única de auditoria no estado atual da página, sem simulação de interações prolongadas ou navegação contínua.
- **Dispositivo: Móvel e Computador** — indica que os testes foram executados considerando tanto a visualização em dispositivos móveis, quanto em computadores.
- **Categorias: Acessibilidade, Práticas Recomendadas e SEO** —
  - *Acessibilidade*: avalia a conformidade da página com diretrizes de acessibilidade.

– *Práticas Recomendadas*: verifica se a página segue padrões modernos de desenvolvimento web, garantindo segurança, desempenho e compatibilidade.

– *SEO (Search Engine Optimization)*: analisa aspectos que influenciam na indexação da página por mecanismos de busca.

5.1.2 WAVE (Ferramenta de Avaliação de Acessibilidade da Web)

O WAVE[2] é uma ferramenta gratuita desenvolvida pela WebAIM para avaliar a acessibilidade de páginas web, recomendado pela WCAG. Ele permite identificar problemas que possam dificultar a navegação por pessoas com deficiência, destacando elementos como contrastes inadequados, ausência de descrições em imagens e estrutura de cabeçalhos. A análise pode ser feita diretamente pelo site ou por meio de uma extensão do navegador, fornecendo feedback visual que facilita a correção dos problemas detectados (WEBAIM, 2025).

Na Figura 66 é apresentado uma exemplificação da utilização da extensão WAVE, enquanto o quadro apresenta os resultado dos testes realizados.

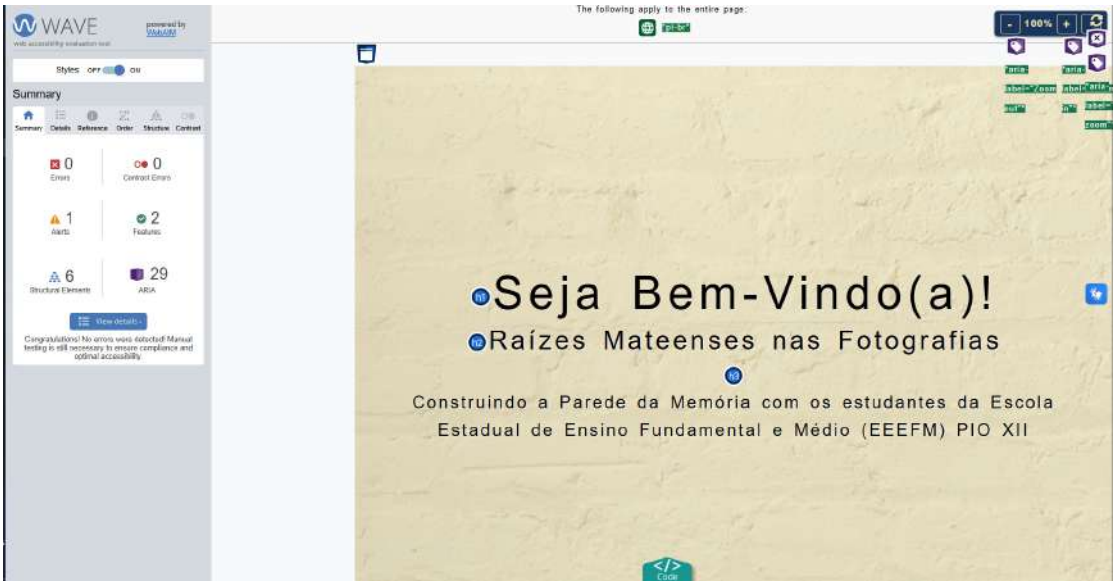

Figura 66 – Tela inicial do sistema com extensão WAVE

Fonte: Do Autor

Para a realização dos testes, foi realizada a navegação em todas as páginas do sistema, utilizando a aplicação WAVE. Como a ferramenta executa automaticamente a análise com base nos critérios de sucesso da WCAG, não foi necessária nenhuma configuração adicional, sendo suficiente apenas interpretar os resultados apresentados. Ao abrir o painel exibido no lado esquerdo da página, foram apresentados os achados

[2]https://wave.webaim.org/

da avaliação, como, por exemplo, a verificação da hierarquia dos textos, a adequação do contraste e o atendimento aos demais critérios de sucesso. Os resultados consolidados dessa análise podem ser consultados no quadro 7.

Descrição do defeito	Solução	Estado
Navegador com dificuldade de identificar classificação/hierarquia dos textos	Alterado a hierarquia dos textos apresentados na modal, substituindo a tag *<p>* por *<h2>*	Corrigido

Quadro 7 – Descrição de resultados dos testes usando extensão WAVE.

Fonte: Do Autor

5.1.3 Contraste

Um dos aspectos essenciais para garantir a acessibilidade de um website é o contraste entre seus elementos. Para validar esse requisito, utilizou-se a aplicação Color Contrast Analyzer[3], a extensão Dark Reader[4] e a extensão Colorblindly[5].

Utilizando a aplicação Color Contrast Analyzer foi realizado o procedimento de captura das cores dos elementos textuais e visuais do sistema, utilizando a funcionalidade de conta-gotas disponibilizada pela ferramenta. A partir dessa seleção, o Color Contrast Analyzer comparou automaticamente a cor do texto com a cor de fundo, gerando o valor de contraste correspondente. Com base nesse resultado, foi possível avaliar se cada item atingia os níveis de conformidade exigidos (AA ou AAA). A Figura 67 ilustra esse processo.

Figura 67 – Tela da aplicação Color Contrast Analyzer

Fonte: (TPGI, 2025)

[3]Disponível em: <https://www.tpgi.com/color-contrast-checker/>.
[4]Disponível em <https://darkreader.org/>
[5]Disponível em <https://wearecolorblind.com/resources/colorblindly-colorblindness-simulator/>

Também foram realizados testes complementares utilizando a extensão Dark Reader, que oferece a possibilidade de aplicar temas escuros ou de inversão de cores em todo o navegador. Essa funcionalidade permitiu simular cenários em que usuários optam por modos de exibição alternativos, seja por preferência pessoal, seja por necessidade relacionada a condições visuais específicas, como sensibilidade à luminosidade. Durante os testes, foram avaliados o comportamento dos elementos textuais e visuais do sistema "Parede da Memória" sob essas diferentes configurações, observando se a legibilidade era preservada, se havia perda de informações visuais relevantes ou sobreposição inadequada de cores. A Figura 68 apresenta um exemplo da interface submetida a esse processo de análise.

Figura 68 – Tela da extensão Dark Reader

Fonte: (LTD., 2025)

Por fim, foi utilizada a extensão Colorblindly, que permite simular a percepção visual de pessoas com diferentes tipos de daltonismo. O objetivo dessa análise foi verificar se os elementos do sistema "Parede da Memória" permaneciam legíveis e compreensíveis mesmo quando visualizados sob condições de deficiência de percepção de cores. A ferramenta possibilitou alternar entre diferentes simulações, como protanopia, deuteranopia e tritanopia, avaliando o contraste entre elementos textuais e visuais, bem como a distinção entre cores que possuíam funções informativas. Dessa forma, foi possível identificar potenciais barreiras que poderiam comprometer a usabilidade do sistema por usuários daltônicos. A Figura 69 apresenta um exemplo da interface analisada por meio dessa ferramenta.

Figura 69 – Tela da extensão Colorblindly

Fonte: (COLORBLIND, 2025)

O resultado para todos os testes realizados foram documentados no Quadro 8.

Descrição do defeito	Solução	Estado
Contraste da mensagem de texto campo obrigatório na tela de login não passou na visualização "modo dark"da extensão Dark Reader	Alterado a cor do campo obrigatório para uma com maior contraste (*1A365D*)	Corrigido
Contraste do conteúdo textual do sistema utilizando a visualização "modo claro"da extensão Dark Reader não passou no critério 1.4.6	Nenhuma alteração feita pela paleta de cores do cliente	Não alterado

Quadro 8 – Descrição de resultados dos testes usando Color Contrast Analyzer, Colorblindly e Dark Reader.

Fonte: Do Autor

É importante destacar que as cores originais do site atenderam plenamente a todos os critérios de contraste definidos pela WCAG. Contudo, ao utilizar a extensão Dark Reader, que altera automaticamente o tema do navegador e modifica a paleta de cores da página, foi identificado um caso específico em que um dos critérios não foi atendido. Esse resultado não está relacionado ao design original do sistema, mas sim à modificação externa aplicada pela extensão.

5.1.4 Validação de HTML

Além dos testes visuais realizados dentro do sistema, foram executadas validações técnicas aprofundadas diretamente no código-fonte HTML das páginas geradas. Este processo meticuloso foi essencial para garantir que a acessibilidade não fosse apenas aparente na interface, mas que estivesse intrinsicamente codificada, seguindo os padrões web estabelecidos.

Para isso, utilizou-se uma combinação de ferramentas de auditoria especializadas e reconhecidas internacionalmente: a plataforma portuguesa AcessMonitor[6] e o Validador de Markup do W3C (World Wide Web Consortium). A escolha dessas ferramentas deve-se à sua autoridade em seus respectivos campos; o AcessMonitor é especificamente desenvolvido para verificar conformidade com as Diretrizes de Acessibilidade para Conteúdo Web (WCAG), enquanto o Validador [7] assegura que o código *HTML* está sintaticamente correto e bem-formado, um pré-requisito fundamental para uma acessibilidade robusta.

Em ambos os casos, o procedimento consistiu na submissão direta do código-fonte das páginas do sistema "Parede da Memória" às plataformas de validação. O AcessMonitor realizou a análise automática com base nos critérios de sucesso da WCAG, gerando relatórios detalhados com as conformidades e as falhas encontradas em relação à acessibilidade. Já o Validador W3C verificou a estrutura do código HTML e CSS de acordo com os padrões internacionais da Web, permitindo identificar inconsistências de marcação e potenciais problemas de compatibilidade entre navegadores. As Figuras 70 e 71 que, respectivamente, apresentam as telas do AcessMonitor e Validador W3C. Enquanto o Quadro 9 apresenta o resultado dos testes realizados.

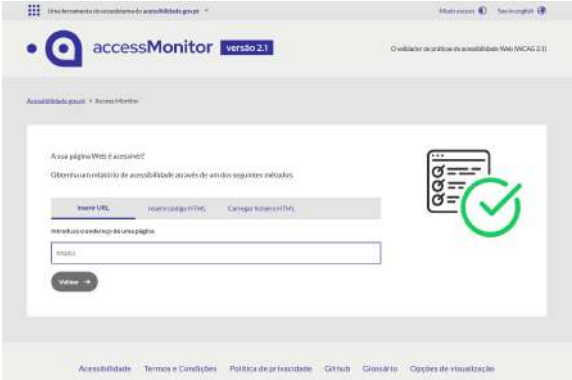

Figura 70 – Tela do site AcessMonitor

Fonte: ((AMA), 2025)

[6]Disponível em <https://accessmonitor.acessibilidade.gov.pt/>
[7]Disponível em <https://validator.w3.org/>

Figura 71 – Tela do Validador W3C

Fonte: ((AMA), 2025)

Descrição do defeito	Solução	Estado
Botões sem nomes acessíveis para identificação	Adicionado *aria-label* como propriedade	Corrigido
Elemento interativo com texto acessível que difere do texto vísivel	Alinhado a nomenclatura do nome do link com seu texto acessível	Corrigido
Imagem não contém texto alternativo	Adicionado a propriedade *alt* e sua respectiva descrição	Corrigido
Constatado a falta do atributo *lang*	No projeto Spring, somente o arquivo *index.html* tem essa tag que engloba todas as demais páginas	Corrigido
Constatado a falta do atributo *title* nas páginas	A manipulação do atributo *title* das páginas estão contidas no arquivo *.ts*	Corrigido

Quadro 9 – Descrição de resultados dos testes usando AcessMonitor e Validador W3C.

Fonte: Do Autor

5.1.5 Simulador para responsividade

Durante os testes de responsividade do sistema, foi utilizado a extensão de navegador Simulador móvel (FIRST, 2025) que simula diferentes tamanhos de tela, como os de smartphones e tablets. Foi verificado se os elementos da interface se ajustavam corretamente em diversos dispositivos, garantindo uma boa experiência de uso. Com ela, foi possível identificar e corrigir problemas de layout, sobreposição de componentes e quebra de conteúdo, assegurando que o sistema fosse funcional e visualmente adequado em resoluções variadas.

Os testes foram conduzidos de forma sistemática, navegando por todas as telas do sistema “Parede da Memória” em diferentes resoluções simuladas. Para cada tela, foram executadas as versões correspondentes a dispositivos móveis, tablets e notebooks, observando o comportamento dos elementos da interface em cada contexto. Esse processo

permitiu verificar se os componentes se mantinham organizados, legíveis e funcionais, independentemente do tamanho da tela, além de identificar pontos específicos que necessitavam de ajustes para assegurar a responsividade completa do sistema. O Quadro 10 apresenta os resultados dos testes.

Descrição do defeito	Solução	Estado
Muitas telas quebraram durante os testes em diferentes tamanhos de dispositivos, prejudicando a visualização dos conteúdos.	Foram adicionadas media queries específicas para adaptar os elementos conforme a largura da tela.	Corrigido

Quadro 10 – Descrição de resultados dos testes de responsividade.

Fonte: Do Autor

5.1.6 Axe DevTools

O Axe DevTools, suite de ferramentas de auditoria de acessibilidade desenvolvida pela Deque Systems(SYSTEMS, 2025b) – empresa líder e referência no mercado de conformidade digital –, foi empregado de forma estratégica na avaliação, atuando através de duas modalidades complementares de análise: (i) testes automatizados e (ii) testes guiados (guided tests).

Um aspecto distintivo do Axe DevTools em relação a outras extensões convencionais é a sua integração nativa ao painel de desenvolvedor (DevTools) do navegador. Para acessá-lo, é necessário abrir o painel (geralmente via tecla F12 ou comando de inspecionar elemento) e navegar até a aba específica intitulada "axe DevTools".

A versão gratuita da extensão para navegador executada de forma local e rápida, escaneando o DOM da página em busca de padrões de código conhecidos que violam os critérios de sucesso.

Já os testes guiados exigem a criação de uma conta e disponibiliza um período experimental de 14 dias, utiliza um sistema de Inteligência Artificial proprietário da Deque para orientar o avaliador humano através de um fluxo de verificações manuais estruturadas. A IA não realiza o teste sozinha, mas direciona o usuário, passo a passo, a interagir com a interface e, em seguida, registra o resultado.

Os testes no sistema foram realizados de forma sistemática, navegando por todas as telas do sistema enquanto a extensão Axe DevTools estava ativa no DevTools do navegador. Para os testes automatizados, cada página foi escaneada individualmente, permitindo que a ferramenta identificasse rapidamente os problemas de acessibilidade existentes, como elementos sem textos alternativos, contrastes inadequados e estrutura semântica incorreta. Em seguida, foram conduzidos os testes guiados, seguindo o fluxo orientado pela inteligência artificial da extensão, interagindo com componentes, menus e formulários, e

registrando manualmente os resultados das verificações recomendadas.

As Figuras 72 e 73 ilustram, respectivamente, as interfaces de análise automatizada e a interface de testes guiados. Enquanto o Quadro 11 apresenta os resultados dos testes realizados.

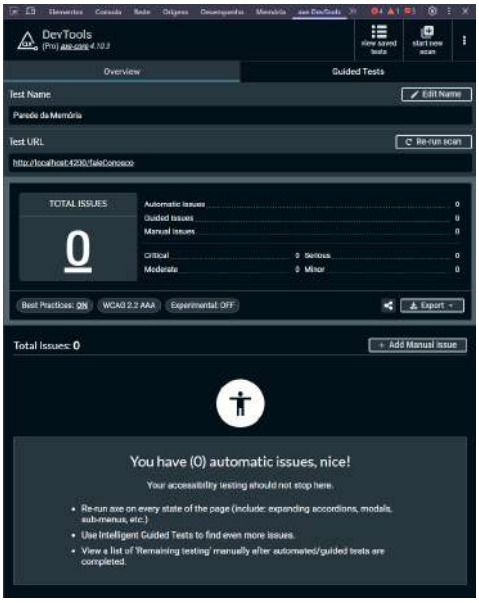

Figura 72 – Tela da extensão Axe DevTools para testes automatizados

Fonte: (SYSTEMS, 2025a)

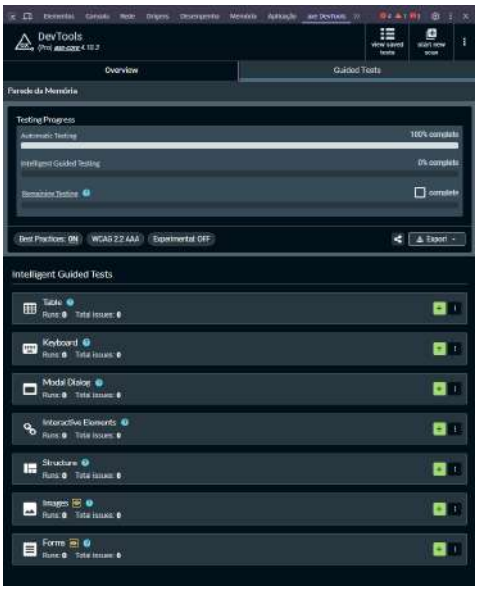

Figura 73 – Tela da extensão Axe DevTools para testes guiados

Fonte: (SYSTEMS, 2025a)

Problema identificado	**Implementação da correção**	**Status**
Ausência de landmark principal para navegação	Implementação de *<div role="main»* contendo o conteúdo principal no arquivo *home.html*	Corrigido
Nome acessível ausente/inadequado para elementos de formulário	Melhoria nos rótulos(*label*) de campos no formulário para maior clareza das instruções	Corrigido
Botão inacessível por teclado (propriedade *disabled* ativada)	Remoção do estado *disabled* padrão	Corrigido
A página principal apresentava a hierarquia dos títulos iniciando em *h2*	Alterado para que a hierarquia iniciasse a partir do *h1*	Corrigido
Múltiplos elementos estavam utilizando o mesmo valor no atributo *aria-label*	Alterado os valores do atributo para que fossem únicos	Corrigido
Funcionalidade inacessível via teclado	Adição de atributo *role*	Corrigido
Quando um modal é aberto, a página por trás ainda fica acessível causando um impacto na ordem de leitura	Implementado um foco preso aos elementos da modal, bloqueando conteúdos externos com *aria-hidden*	Corrigido
O papel (*role*) do elemento da modal está ausente podendo levar a leitores de tela terem dificuldade de entenderem sua finalidade	Adicionado ao item da modal os atributos *role="dialog"aria-modal="true"*	Corrigido
Campos obrigatórios devem incluir *ria-required="true"* a (além de *required*) para que leitores de tela anunciem corretamente.	Adicionado o atributo em todos os campos obrigatórios	Corrigido
Item genérico do código (** sem role associada	Adicionado um atributo *role="button"* para identificação como botão	Corrigido

Quadro 11 – Resultados dos testes usando Axe DevTools.

Fonte: Do Autor

5.1.7 HeadingsMAp

Foram realizados testes envolvendo a hierarquia e o mapa de cabeçalhos do sistema, garantindo que as páginas apresentassem uma estrutura lógica adequada para leitores de tela e que as tags HTML fossem utilizadas corretamente. Para isso, foi utilizada a extensão de navegador HeadingsMap[8] , que permite visualizar a estrutura de cabeçalhos de cada página de forma clara e organizada.

Os testes foram conduzidos navegando por todas as telas do sistema "Parede da Memória" com a extensão ativa, verificando a ordem e a hierarquia dos elementos *H1* a *H6* em cada página. Durante a análise, foram observados possíveis desalinhamentos ou uso inadequado de tags, assegurando que cada seção estivesse corretamente estruturada e identificável por tecnologias assistivas.

A Figura 74 apresenta a tela principal do sistema com a extensão HeadingsMap ativa, demonstrando a visualização da hierarquia de cabeçalhos.

[8]Disponível em https://chromewebstore.google.com/detail/headingsmap/flbjommegcjonpdmenkdiocclhjacmbi

Figura 74 – Tela da extensão HeadingsMAp

Fonte: Do Autor

Os resultados mostraram que a hierarquia das páginas estava correta, não sendo necessárias alterações adicionais, confirmando que a estrutura do conteúdo atendia aos critérios de acessibilidade relacionados à navegação e leitura por leitores de tela.

5.1.8 Funkify

Utilizando a extensão Funkify[9] , foi possível realizar testes no sistema simulando diferentes tipos de usuários e habilidades, incluindo baixa visão, limitações motoras, dislexia, entre outras. O procedimento consistiu em navegar por todas as telas da aplicação com a extensão ativa, alternando entre os modos de simulação para observar como cada funcionalidade e elemento da interface se comportava sob diferentes condições de uso.

A Figura 75 apresenta a tela da extensão durante a execução dos testes para dislexia, enquanto o Quadro 12 apresenta os resultados obtidos.

Figura 75 – Tela da extensão Funkity

Fonte: (AB, 2025)

[9]Disponível em <https://www.funkify.org/>

Descrição do defeito	Solução	Estado
Durante os testes utilizando o simulador de tremores e baixa visão - onde o cursor do mouse apresenta movimentos involuntários -, identificou-se que o ícone para fechar o menu lateral possuía uma área de clique insuficiente, dificultando a interação dos usuários.	Realizado o aumento do tamanho do ícone para ter uma área de clique mais abrangente	Corrigido
Durante os testes utilizando o simulador de dislexia, encontrou-se muito dificuldade para ler textos, sejam pequenos ou longos	Adicionado em todo conteúdo textual do sistema um espaçamento, entre as palavras e linhas, recomendado pela WCAG	Corrigido

Quadro 12 – Descrição de resultados dos testes usando Funkify.

Fonte: Do Autor

5.1.9 NVDA

Para a realização dos testes de acessibilidade com leitores de tela, utilizou-se o NVDA (NonVisual Desktop Access)[10] , um software gratuito e de código aberto que, após instalado no ambiente Windows, narra automaticamente todas as interações do usuário com a interface, incluindo navegação por elementos, leitura de conteúdos textuais e execução de ações. A Figura 76 apresenta a interface inicial do NVDA durante os testes e o Quadro 13 os resultados obtidos.

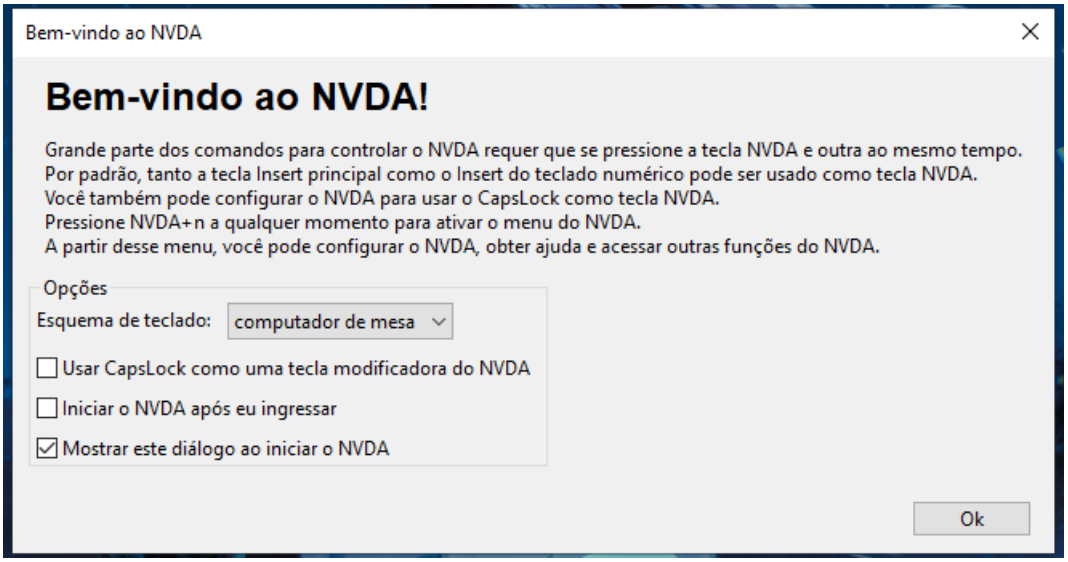

Figura 76 – Tela da aplicação NVDA

Fonte: (ACCESS, 2025)

[10]Disponível em <https://www.nvaccess.org/download/>

Descrição do defeito	Solução	Estado
"Link" sem texto descritivo, ficando invisível para o leitor de tela	Adição do atributo *aria-label*	Corrigido
Imagem sem texto propriedade *alt*, ficando invisível para o leitor de tela a descrição	Adição do atributo *alt*	Corrigido
Incompreensão do leitor de tela para a leitura de tabelas	Adição dos atributos *scope="col"* e *scope="row"*	Corrigido
Ao digitar a senha, como está "mascarado" o leitor de tela ficou lendo e interpretando como 'asteriscos', logo o usuário teria dificuldade para entender se está escrevendo correto ou não	Alterado a visualização inicial do campo para não ficar como *default* mascarado	Corrigido
Ao abrir uma modal, o título da mesma não está sendo lida	Adicionado um atributo *aria-labelledby* associada a tag *h1* da modal	Corrigido

Quadro 13 – Descrição de resultados dos testes usando NVDA.

Fonte: Do Autor

Os testes foram conduzidos navegando por todas as telas do sistema, utilizando o NVDA para simular a experiência de usuários com deficiência visual. Durante o procedimento, foi verificado se todos os elementos da interface eram corretamente identificados e descritos pelo leitor de tela, se a ordem de navegação seguia uma sequência lógica, e se mensagens, formulários e botões forneciam feedback adequado.

5.2 Testes Manuais

Nas seções a seguir, são apresentadas validações realizadas no sistema sem o uso de ferramentas externas, contemplando validações com base nos critérios da WCAG e registros de reuniões.

5.2.1 WCAG

Os critérios de sucesso da WCAG 2.2 foram adotados como referência principal para a avaliação dos parâmetros de acessibilidade do sistema. Para organizar e documentar a análise, foi elaborado um checklist baseado nesses critérios, no qual cada item foi verificado em todas as telas da aplicação. No Quadro 14, são apresentados os critérios de sucesso que não foram atendidos durante a avaliação.

Quadro 14 – Resultado das validações dos critérios de sucesso WCAG - Perceptível

Critério de Sucesso	**Justificativa**
1.2.1 Apenas áudio e apenas vídeo (pré-gravado) (A)	O critério não se aplica, pois o site não possui conteúdo somente em áudio ou vídeo pré-gravado.
1.2.2 Legendas para Mídia Baseada em Tempo (A)	O critério não se aplica, pois o site não possui mídias temporais, como vídeos ou áudios sincronizados.
1.2.3 Descrição de Áudio ou Alternativa para Mídia (A)	O critério não se aplica, pois não há conteúdo multimídia com áudio e vídeo simultâneos.
1.2.4 Legendas (Gravado) (AA)	O critério não se aplica, pois o site não possui vídeos gravados com áudio.
1.2.5 Descrição de Áudio (Gravado) (AA)	O critério não se aplica, pois não há vídeos com elementos visuais importantes e sem descrição textual.
1.2.6 Alternativa de Mídia (AAA)	O critério não se aplica, pois o site não contém mídias temporais que exijam alternativas equivalentes.
1.2.7 Descrição Estendida de Áudio (AAA)	O critério não se aplica, pois não há vídeos com elementos visuais complexos ou não narrados.
1.2.8 Mídia ao Vivo	O critério não se aplica, pois o site não transmite conteúdos de áudio ou vídeo ao vivo.
1.2.9 Apenas Áudio (Gravado) (AAA)	O critério não se aplica, pois o site não possui conteúdo apenas em áudio gravado.
1.4.2 Controle de Áudio (A)	O critério não se aplica, pois o site não possui nenhum conteúdo com reprodução automática de áudio.
1.4.7 Imagens de Texto (Não Essenciais) (AAA)	O critério não se aplica, pois o site não possui áudio com reprodução automática nem sons de fundo.
2.2.1 Tempo Ajustável (A)	O critério não se aplica, pois o site não possui nenhuma funcionalidade de tempo ajustável
2.2.2 Pausar, Parar, Ocultar (A)	O critério não se aplica, pois o site não possui elementos que precisem de controle de pausa/stop.
2.3.3 Animações ao Interagir (AAA)	O critério não se aplica, pois o site não possui nenhuma animação.

Quanto aos demais critérios, listados no apêndice A, o sistema está em conformidade.

5.2.2 Reuniões

Durante a criação do sistema, foi realizado reuniões com as partes interessadas (as professoras coordenadoras Ana Carolina de Melo Loureiro e Prisciliana Costa Ventura) e também com professores da UFES para validar questões técnicas. A seguir, é descrito todas as mudanças feitas com base nessas discussões.

No dia 30 de abril de 2025, foi realizado uma reunião presencial com o professor orientador Giuliano Prado, onde foram levantados os pontos citado no quadro e sendo importante ressaltar que todas as alterações foram realizadas mediante aprovação das partes interessadas.

Descrição	Solução
Alterar fonte do sistema, pois a visualização da atual não é recomendado pela acessibilidade	Trocado a fonte para 'Arial', sans-serif;
Adicionar alternativa para controlar zoom no sistema	Implementado funcionalidade de aumentar e diminuir o zoom
Renomear fluxo de listagem de formulários para ficar mais intuitivo	Realizado renomeação do fluxo para "Agendamentos"
Ao acessar a página "Fale Conosco", estando logado, deve ser ocultado o formulário de "Marque uma visita"por ser uma funcionalidade que deve ficar disponível apenas para usuários que não estejam logado	Realizado ocultação do formulário
Realizar padronização do tamanho de *cards* de todo sistema	Após alinhamento com as partes interessadas, foi realizado a padronização conforme solicitado por elas. Dessa forma, somente em alguns tamanhos de tela, os *cards* ficaram com um tamanho fixo
Adicionar um campo de "Resumo"para os subprojetos do sistema e utilizar essa informação nos *cards* dos subprojetos	Adicionado o campo
Alterar a visualização da descrição dos subprojetos para ser apresentado somente se o usuário expandir o *accordion* na tela de "Listagem de Publicações	Alterado a visualização
Alterar o layout das informações das publicações, onde deve ser retirado os inputs - ao visualizar - e aumentar o espaço de apresentação das fotos	Alterado a visualização das publicações

Quadro 15 – Descrição de resultados da reunião com orientador.

Fonte: Do Autor

No dia 02 de maio de 2025, foi realizado uma reunião com as professoras Ana Carolina de Melo Loureiro e Prisciliana Costa Ventura, sendo elas o cliente principal

do sistema "Parede da Memória", sendo apresentadas no quadro abaixo as solicitações levantadas durante a reunião:

Descrição	Solução
Adicionar campo de "Telefone"no formulário de "Marque uma visita"	Incluído o campo de "Telefone"no fluxo de agendamento de visita
Alterar o número de caracteres do campo "Objetivo da visita"no formulário de "Marque uma visita"	Alterado o número de caracteres do campo para 500 caracteres
Alterar o número de caracteres do campo "Descrição"das fotos das publicações	Alterado o número de caracteres do campo para 500 caracteres
Renomear campo de descrição das fotos das publicações	Alterado o nome do campo para "Descrição da imagem"
Adicionar campo "Data marcada"no fluxo de agendamento de visita	Adicionado o campo
Renomear campo de data no fluxo de agendamento de visita, o campo deve se chamar "Data prevista"	Alterado o nome do campo
Adicionar possibilidade de criar, editar, visualizar e excluir itens do fluxo de "Quem Somos"	Realizado o levantamento de requisitos e alterado o fluxo, conforme solicitação
Alterar nível de acesso dentro do sistema para administradores e coadministrador	Realizado o levantamento de requisitos e alterado o fluxo com seus, respectivos, níveis de acesso
Alterar o texto que é apresentado na tela "Sobre o projeto"	Após ser enviado o texto, foi realizado a alteração.
Alterar layout de apresentação dos *cards* da listagem das publicações	Realizado o levantamento de requisitos e alterado o layout conforme solicitado

Quadro 16 – Descrição de resultados da primeira reunião com partes interessadas.

Fonte: Do Autor

Também foi realizado uma reunião no dia 08 de maio de 2025 com Simone Dornelas Costa, docente da UFES na área de interface humano-computador, que orientou a realização de testes em diferentes navegadores (Firefox, Edge, Opera, Brave, Chrome), além de realizar a validação do código através do Validador W3C.

Após a implementação de todas as melhorias e correções solicitadas anteriormente, realizou-se uma nova reunião com as partes interessadas em 04 de junho de 2025, na qual todas as alterações foram aprovadas.

No dia 25 de julho de 2025, foi realizado uma nova reunião com as parter interessadas, onde foi solicitado a seguinte alteração:

Descrição	Solução
Remover a CRUD de "Endereço"na tela de "Fale Conosco, substituindo pela incorporação do mapa do Google, deixando fixo o endereço.	Realizado alteração conforme solicitado

Quadro 17 – Descrição de resultados da terceira reunião com partes interessadas.

Fonte: Do Autor

5.2.2.1 Implantação

No dia 28 de julho de 2025, foi realizada uma visita presencial à Escola PIO XII para a disponibilização local do sistema em uma máquina destinada ao projeto "Parede da Memória". Visando proporcionar uma experiência mais fluida e acessível para os usuários, optou-se pela disponibilização do sistema por meio da ferramenta Docker.

Ao disponibilizar o sistema através de um container no Docker foi possível empacotar todas as dependências, bibliotecas e configurações do sistema em um único ambiente padronizado, reduzindo a complexidade de instalação e minimizando erros de configuração. Essa escolha teve como principal objetivo simplificar a execução do sistema, proporcionando uma interface mais clara, organizada e visualmente agradável, especialmente para os usuários da escola que não possuem familiaridade técnica.

Junto à instalação, foi entregue uma documentação completa e cuidadosamente elaborada, com o objetivo de promover autonomia à equipe escolar. Esse material contém instruções claras e detalhadas, elaboradas de forma acessível mesmo para usuários com pouca familiaridade com tecnologias. Entre os conteúdos disponíveis, destacam-se o passo a passo para instalação do Docker, o procedimento para download da pasta do projeto — contendo todo o código-fonte —, os comandos necessários para executar o sistema por meio de containers e orientações detalhadas para configuração e execução do ambiente completo.

Além de servir como guia técnico, essa documentação também foi pensada como um recurso de empoderamento. Como contempla todas as configurações do ambiente, qualquer pessoa com acesso ao material pode, de forma independente, reinstalar ou configurar o sistema em outro dispositivo, o que garante maior sustentabilidade e continuidade ao projeto, mesmo em caso de trocas de equipe ou equipamentos.

Outro aspecto fundamental abordado na documentação foi o processo de backup. Durante a visita, foi apresentado à equipe da escola como realizar corretamente o salvamento dos dados do projeto, de forma a preservar o conteúdo já inserido no sistema — como postagens, fotografias e descrições. O backup foi ressaltado como uma etapa essencial para garantir a segurança das informações, prevenindo perdas em casos de falhas técnicas, exclusões acidentais ou troca de equipamentos.

Além da parte técnica, a visita também teve um momento de troca e aprendizado com a equipe da escola. Foi feita uma apresentação simples e interativa com as professoras e os alunos que estão diretamente envolvidos com o projeto. Durante esse momento, foi apresentado como o sistema funciona na prática — desde a abertura até o cadastro de novas postagens com fotos e textos. Também houve um tempo reservado para tirar dúvidas e dar dicas de uso, com orientações.

A implantação do sistema foi bem recebida pela equipe da escola PIO XII, que destacou a interface simples, a organização do acervo e o suporte prestado.

6 Discussão dos resultados e considerações finais

O projeto exigiu organização, planejamento, capacidade de adaptação e busca por soluções, sendo habilidades que serão levadas para o mercado de trabalho. Mas, principalmente, permitiu o aprofundamento em temas como usabilidade, acessibilidade e desenvolvimento centrado no usuário.

A implantação inicial do sistema foi realizada em ambiente local, permitindo testes de navegação, validação de conteúdo e sua organização e análises de acessibilidade. Durante essa fase, a recepção dos usuários finais foi bastante positiva. As professoras responsáveis pelo projeto, Ana Carolina de Melo Loureiro e Prisciliana Ventura, demonstraram satisfação com o sistema, destacando especialmente a organização das informações, o layout, a navegação intuitiva e a adequação do conteúdo do site para à proposta do projeto "Parede da Memória". Além disso, o sistema ampliou as possibilidades de trabalho dos envolvidos, pois permitiu o acesso facilitado ao acervo fotográfico, a atualização de conteúdos de maneira autônoma e o uso do material como recurso didático em sala de aula.

A estrutura do sistema proporcionou uma nova forma de interação com as imagens históricas da comunidade do município de São Mateus(ES), local onde o projeto “Parede da Memória” atende. Anteriormente, as fotos ficavam limitadas ao espaço físico da escola, sendo impressas o reveladas e expostas em murais nas paredes e, com a digitalização e organização no ambiente digital (web), o conteúdo pode ser acessado de forma dinâmica, com filtros por categorias e possibilidade de navegação. E com a futura disponibilização do sistema localmente, isso facilitou com que o conteúdo possa ser acessado por qualquer pessoa, ampliando seu uso para além da escola. As imagens podem ser aproveitadas como material didático em diferentes contextos educacionais e também servir como fonte de consulta para interessados na história local. Além disso, o site abre espaço para a participação da comunidade em geral que podem entrar em contato para agendar visitas ou até contribuir com novas informações e fotografias.

A aplicação das técnicas de acessibilidade foi planejada desde as etapas iniciais do projeto, guiando decisões de layout, organização de conteúdo e funcionalidades. Embora não tenha sido possível realizar testes com usuários com deficiência durante o desenvolvimento, foram utilizadas diversas diretrizes da WCAG 2.2 para garantir o máximo de conformidade. Ferramentas automáticas, como o axe DevTools e o Lighthouse, foram empregadas para validação técnica, além de inspeções manuais para verificar contrastes, foco, semântica do HTML e estrutura lógica de navegação.

Foram aplicadas técnicas como o uso de cores com alto contraste, estruturação adequada por meio de títulos e subtítulos (hierarquia semântica), utilização de elementos nativos do HTML, inclusão de textos alternativos em imagens e aplicação do atributo *lang* para garantir a correta leitura por leitores de tela. A validação, realizada por meio de ferramentas automáticas e também por inspeção manual, indicou que o site atendeu a 72 critérios de sucesso da WCAG 2.2, correspondendo a aproximadamente 84% do total. Esses resultados demonstram um bom nível de acessibilidade, ainda assim, reconhece-se que a realização de testes com usuários com deficiência é essencial, ficando como recomendação para futuras fases do projeto.

Uma análise crítica da aplicação evidencia alguns pontos positivos importantes: o projeto promoveu a integração entre teoria e prática, contribuiu para a formação profissional, e gerou um produto com real potencial de impacto social. Além disso, o desenvolvimento guiado por princípios de acessibilidade fortaleceu a visão de inclusão digital e responsabilidade social no uso da tecnologia.

Entre os pontos de melhoria, destaca-se a limitação no tempo disponível para testes mais amplos com diferentes perfis de usuários, especialmente aqueles com deficiência. Além disso, a comunicação entre a equipe técnica e os usuários não técnicos da escola exigiu atenção constante, com adaptações na linguagem e nas orientações para garantir uma boa compreensão do sistema. Quanto à disponibilidade, embora o sistema tenha sido implantado localmente em um computador do projeto utilizando Docker — o que possibilitou o uso e a alimentação do sistema pela equipe escolar —, a ausência de uma hospedagem online limitou seu alcance público e o acesso remoto contínuo. Ainda assim, essa estratégia garantiu o funcionamento pleno da aplicação dentro da escola.

A etapa de hospedagem definitiva do sistema ainda está prevista para a continuidade do projeto, uma vez que depende da elaboração de um orçamento e da liberação de verba institucional. A proposta é que o site seja publicado em um servidor público, com um domínio próprio, garantindo acesso direto e contínuo tanto para a equipe da escola quanto para a comunidade externa. Com isso, espera-se ampliar significativamente o alcance do sistema, promovendo maior visibilidade ao projeto e facilitando a participação de novos colaboradores.

Em relação aos objetivos propostos inicialmente, a avaliação é a seguinte:

- Análise e planejamento do sistema: Objetivo plenamente alcançado. Todas as etapas de levantamento de requisitos, definição de arquitetura e planejamento das funcionalidades foram concluídas, resultando em um escopo bem definido que guiou o desenvolvimento.

- Estruturação do banco de dados do sistema: Objetivo alcançado. O banco de dados foi modelado e implementado com sucesso, garantindo a integridade, organização e

eficiente armazenamento de todas as informações do projeto “Parede da Memória”.

- Implementação das funcionalidades do sistema: Objetivo alcançado. O sistema foi integralmente desenvolvido, atendendo a todos os requisitos funcionais previstos, incluindo o gerenciamento do acervo, a interface de navegação para o usuário final e os recursos de acessibilidade.

- Estudo e aplicação de técnicas de acessibilidade, contribuindo para que a web seja mais acessível: Objetivo alcançado. Além do estudo aprofundado das WCAG 2.2 e de tecnologias assistivas, as práticas recomendadas foram aplicadas de forma efetiva no desenvolvimento. Foram implementadas estruturas semânticas, navegação completa por teclado, textos alternativos descritivos para imagens, adequado contraste de cores e outros recursos que garantem uma experiência inclusiva e acessível a todos os usuários, contribuindo diretamente para uma web mais aberta.

- Disponibilização das informações coletadas no Projeto “Parede da Memória” para todos os usuários: Objetivo parcialmente atingido. O sistema foi implantado e está em pleno funcionamento em um ambiente local na escola, utilizando Docker para garantir estabilidade e padronização. Ele organiza e apresenta o conteúdo de forma clara, responsiva e acessível, sendo utilizado por professores e alunos. No entanto, como ainda não foi publicado na internet, seu acesso permanece restrito à comunidade escolar. A infraestrutura técnica e visual da aplicação está completamente preparada para uma futura hospedagem online, que permitirá o acesso universal ao acervo.

Com o desenvolvimento do presente trabalho, foi possível colocar em prática e consolidar diversos conhecimentos aprendidos no curso de Sistemas de Informação, como os principais exemplos: análise e gerenciamento de requisitos, modelagem de sistemas, modelagem de banco de dados, conhecimento de programação de aplicações web. Dessa forma, após estudos e análise, foi possível atingir o objetivo geral que se tratava do desenvolvimento de um sistema web para a Escola PIO XII, vinculado ao projeto de extensão UFES PRO TI, consolidando-se como uma importante ferramenta de disseminação do conteúdo do projeto "Parede da Memória".

Além de utilizar os conhecimentos adquiridos na graduação, também foi extremamente importante o conhecimento acerca de acessibilidade digital e como alcançá-la, essa etapa não só auxiliou na construção de um site acessível, como também agregou conhecimento técnico relevante à formação profissional. Contribuir para uma web mais acessível para os usuários também foi um dos pilares do projeto. Diversas técnicas de acessibilidade foram aplicadas, validações com ferramentas automatizadas e também validações manuais. O conteúdo do projeto foi estruturado em páginas informativas com foco em acessibilidade e usabilidade do usuário, com recursos que tornam as informações claras e acessíveis.

A aplicação prática de técnicas recomendadas foi de extrema importância para que o site alcançasse o grau aceitável de acessibilidade e mantivesse todos os critérios de requisitos e layout solicitados pelo cliente, mostrando que é possível aliar usabilidade e inclusão sem comprometer o design.

A realização deste trabalho proporcionou um aprendizado prático extremamente relevante. O contato com tecnologias atuais, a necessidade de seguir padrões de acessibilidade e o quanto pequenas técnicas (como o uso do atributo *lang* no HTML) podem fazer uma grande diferença para pessoas que utilizam alguma tecnologia assistiva, o que agregou ao conhecimento pessoal e profissional, sendo experiências enriquecedoras.

Integrar este projeto ao PRO TI reforçou o papel da universidade na transformação social. O sistema desenvolvido atende a uma demanda concreta da escola PIO XII, contribuindo diretamente para a valorização da memória da cidade onde estão as fotografias e o acesso ao conhecimento. Com o foco em acessibilidade, o projeto reforçou ainda mais a transformação social e, com a futura hospedagem do site o impacto social ultrapassará os muros da escola, alcançando alunos, familiares, moradores da região e qualquer pessoa que tenha acesso à internet.

6.1 Trabalhos Futuros

Dentre os próximos passos previstos para a continuidade do projeto, destaca-se a necessidade de realizar a hospedagem definitiva do sistema em um servidor público. Essa etapa permitirá o acesso remoto por parte da comunidade externa, incluindo ex-alunos, familiares e demais interessados, ampliando significativamente o impacto do sistema e reforçando o papel da tecnologia como meio de inclusão e valorização da memória local.

Além disso, considerando que o site foi desenvolvido com foco em acessibilidade, está prevista a realização de testes com pessoas com deficiência, a fim de validar, na prática, a efetividade das soluções adotadas. Essa etapa será essencial para identificar possíveis ajustes e promover melhorias contínuas. Como o sistema será utilizado por esse público, conforme solicitado pelas professoras responsáveis, pretende-se realizar um monitoramento contínuo da aplicação, com a coleta de feedbacks das educadoras à medida que os alunos forem utilizando o sistema. Isso permitirá que as adaptações futuras sejam orientadas por demandas reais de uso, garantindo que a acessibilidade evolua de forma concreta e alinhada às necessidades dos usuários.

Atualmente, todas as funcionalidades solicitadas pela equipe da Escola PIO XII foram atendidas, sendo ressaltado que o projeto foi desenvolvido com base em uma estrutura flexível, que possibilita futuras melhorias e adaptações, caso venham a ser desejadas.

Entre as possíveis evoluções que poderão ser implementadas, destacam-se:

- Criação de uma área de comentários ou depoimentos, fortalecendo o vínculo entre a comunidade e a memória registrada no site;
- Possibilidade de inserção de vídeos nas publicações, enriquecendo a forma de apresentação dos conteúdos;
- Inclusão da funcionalidade de adicionar áudios gravados como descrição das fotos do acervo, ampliando a acessibilidade para usuários com deficiência visual ou dificuldades de leitura, por meio de recursos auditivos.

Um aspecto importante a ser considerado no futuro é a continuidade da manutenção do sistema ao longo do tempo. Apesar de sua interface intuitiva, é compreensível que, com o passar dos anos, possam surgir dúvidas relacionadas à atualização de conteúdos, ajustes técnicos ou alterações da estrutura da aplicação. Pensando nisso, considera-se a possibilidade de oferecer suporte contínuo à equipe da escola, seja por meio de oficinas ou da disponibilização de materiais de apoio claros e acessíveis.

Dessa forma, o sistema permanece preparado para evoluir conforme novas necessidades surjam, levando em consideração os critérios de acessibilidade, usabilidade e requisitos do projeto.

Referências

AB, F. *Funkify – Disability Simulator*. 2025. <https://www.funkify.org/>. Disponível em: <https://www.funkify.org/>.

ACCESS, N. *NVDA – Leitor de tela gratuito para Windows*. 2025. <https://www.nvaccess.org/download/>. Disponível em: <https://www.nvaccess.org/download/>.

ALGAWORKS. *O que é Spring Boot?* 2017. Acessado em 25 de novembro de 2024. Disponível em: <https://blog.algaworks.com/spring-boot/>.

ALGAWORKS. *O que é Angular?* 2018. Acessado em 25 de novembro de 2024. Disponível em: <https://blog.algaworks.com/o-que-e-angular/>.

ALURA. *Bootstrap - O que é, como e quando usar?* 2021. Disponível em: <https://www.alura.com.br/artigos/bootstrap>.

(AMA), A. para a M. A. *Access Monitor: Validador de Acessibilidade Web*. 2025. <https://accessmonitor.acessibilidade.gov.pt/>. Disponível em: <https://accessmonitor.acessibilidade.gov.pt/>.

ANGULAR. *RouterLink - Relative link paths*. 2025. <https://v17.angular.io/api/router/RouterLink#relative-link-paths>.

ARAUJO, R. *Arquitetura MVC (Model-View-Controller)*. 2024. Disponível em: <https://blog.grancursosonline.com.br/arquitetura-mvc/>.

BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software architecture in practice*. [S.l.]: Addison-Wesley Professional, 2003.

BENAKOUCHE, T. Tecnologia é sociedade: contra a noção de impacto tecnológico. *Cadernos de pesquisa*, v. 17, p. 1–28, 1999.

BEZERRA, E. *Princípios de Análise e Projeto de Sistemas com UML*. [S.l.]: Elsevier Rio de Janeiro, 2007. v. 2.

BIGDATACORP; WPT. *[Estudo] Acessibilidade na web brasileira | 3a edição*. 2021.

BOOTSTRAP. *BootStrap*. 2022. Disponível em: <https://getbootstrap.com.br/docs/4.1/about/overview/>.

BOSCH, J. *Design and use of software architectures: adopting and evolving a product-line approach*. [S.l.]: Pearson Education, 2000.

BRASIL. Lei nº 13.146, de 6 de julho de 2015. *Diário Oficial [da] República Federativa do Brasil*, Brasília, DF, 2015. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm>.

BRASIL, G. F. do. *VLibras — Governo Digital*. 2024. Disponível em: <https://www.gov.br/governodigital/pt-br/acessibilidade-e-usuario/vlibras>.

BRASÍLIA, U. C. D. *Engenharia de Requisitos de Software*. 2018. Disponível em: <https://conteudo.catolica.edu.br/conteudos/nbt_cursos/engenharia_requisitos/tema_02/index.html>.

CALDWELL, B. et al. *Understanding WCAG 2.0*. [S.l.]: EU: W3C. Descargado de http://www. w3. org/TR/UNDERSTANDINGWCAG20 . . . , 2008.

CANALTECH. *O que é o Notion?* 2023. Disponível em: <https://canaltech.com.br/apps/o-que-e-o-notion/>.

CAT. Coordenadoria nacional para integração da pessoa portadora de deficiência, corde – secretaria especial dos direitos humanos, presidência da república. *Cadernos de pesquisa*, 2007.

COCKBURN, A. *Escrevendo Casos de Usos Eficazes: Um guia prático para desenvolvedores de software*. [S.l.]: Bookman Editora, 2005.

COLORBLIND, W. are. *Colorblindly Color Blindness Simulator*. 2025. <https://wearecolorblind.com/resources/colorblindly-colorblindness-simulator/>. Disponível em: <https://wearecolorblind.com/resources/colorblindly-colorblindness-simulator/>.

DATAPSICO. *Dicionário de Dados: Um guia para organizá-los de forma eficiente*. 2021. Disponível em: <https://medium.com/datapsico/dicionario-de-dados-ac3ce726c34b>.

DENNIS, A.; WIXOM, B. H.; ROTH, R. M. *Análise e projeto de sistemas* . [S.l.]: Grupo Gen-LTC, 2000.

DEVELOPERS, C. *Introduction to Lighthouse*. 2025. <https://developer.chrome.com/docs/lighthouse/overview>.

DEVELOPERS, G. C. *Lighthouse accessibility score*. 2019. Chrome DevTools documentation. Disponível em: <https://developer.chrome.com/docs/lighthouse/accessibility/scoring>.

DEVMEDIA. *Serviços RESTful: verbos HTTP*. 2016. Disponível em: <https://www.devmedia.com.br/servicos-restful-verbos-http/37103>.

DIAS, A. L. *Um processo para sistemas web com foco em acessibilidade e usabilidade*. Tese (Doutorado) — Universidade de São Paulo, 2014.

DPW. *Introdução à ARIA para acessibilidade na Web*. 2020. Disponível em: <https://desenvolvimentoparaweb.com/miscelanea/aria-acessibilidade-web-a11y/>.

DUARTE, D. M. *OrientaIFPB: um sistema web para gestão das atividades de orientação acadêmica*. Dissertação (B.S. thesis), 2022.

EMAG. *eMAG - Modelo de Acessibilidade em Governo Eletrônico*. 2014. Disponível em: <https://emag.governoeletronico.gov.br>.

ESTATÍSTICA, I. I. B. D. G. E. *Censo Brasileiro de 2010. Brasil: IBGE,*. 2010.

FERNANDES, H. M. *O que é um desenvolvedor frontend e o que ele faz?* 2020. Disponível em: <https://marquesfernandes.com/tecnologia/o-que-e-um-desenvolvedor-frontend-e-o-que-ele-faz/>.

FERNANDES, J.; GODINHO, F. *Acessibilidade aos sítios Web da AP para Cidadãos com Necessidades Especiais – Requisitos de Visitabilidade*. [S.l.]: Programa Acesso da UMIC/PCM e CERTIC / Univ. de Trás os Montes e Alto Douro, 2003.

FERRAZ, R. *Acessibilidade na Web: Boas práticas para construir sites e aplicações acessíveis*. [S.l.]: Casa do Código, 2020. v. 1.

FIELDING, R. T. *Architectural styles and the design of network-based software architectures*. [S.l.]: University of California, Irvine, 2000.

FILHO, D. C. de O. Um passo a passo para a elaboração do diagrama de caso de uso da uml. 2011.

FIRST, M. *Simulador móvel – ferramenta de teste responsiva*. 2025. Chrome Web Store. Versão 4.5.8; 1000000+ usuários; 3.7K avaliações com nota média 4.9/5. Disponível em: <https://chromewebstore.google.com/detail/simulador-m%C3%B3vel-ferrament/ckejmhbmlajgoklhgbapkiccekfoccmk?hl=pt-BR>.

FREIRE, A. P. *Acessibilidade no desenvolvimento de sistemas web: um estudo sobre o cenário brasileiro*. Tese (Doutorado) — Universidade de São Paulo, 2008.

GRILLO, F. D. N.; FORTES, R. P. d. M. Aprendendo javascript. *São Carlos: USP*, 2008.

HAT, R. *O que é uma API?* 2017. Disponível em: <https://www.redhat.com/pt-br/topics/api/what-are-application-programming-interfaces>.

HAT, R. *API REST*. 2020. Disponível em: <https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api>.

HATANAKA, F. M. *Ciclo de vida do Scrum*. 2013. Disponível em: <https://www.hatanaka.com.br/info/artigo/gerenciamento-de-projetos/ciclo-de-vida-scrum.php>.

HEUSER, C. A. *Projeto de banco de dados: Volume 4 da Série Livros didáticos informática UFRGS*. [S.l.]: Bookman Editora, 2009.

HOSTINGER. *Acessibilidade Web – O Que É e Como Construir um Site Acessível?* 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/acessibilidade-web>.

HOSTINGER. *O que é CSS? Guia Básico para Iniciantes*. 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-css-guia-basico-de-css>.

HOSTINGER. *O Que é HTML? Guia Básico Para Iniciantes*. 2022. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-html-conceitos-basicos>.

HOSTINGER. *O que é Bootstrap*. 2024. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-bootstrap#O_Que_e_Bootstrap>.

IBGE. *Pesquisa Nacional por Amostra de Domicílios - Síntese de Indicadores 2014.Brasil: IBGE,*. 2014.

IBM. *O que é Java Spring Boot?* 2022. Acessado em 25 de novembro de 2024. Disponível em: <https://www.ibm.com/br-pt/topics/java-spring-boot>.

INFOESCOLA. *HTML*. 2022. Disponível em: <https://www.infoescola.com/informatica/html/>.

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION. *Ergonomics — Accessibility of human-system interaction*. 2021. Disponível em: <https://www.iso.org/standard/74071.html>.

JWT.IO. *Introduction to JSON Web Tokens*. 2025. <https://jwt.io/introduction>.

LAMPKOWSKI, M.; SILVA, J. H. Integração de sistemas de informação: A importância das integrações e como realizá-las com interoperabilidade de forma a produzirem e consumirem informações reutilizáveis. In: *III JORNACITEC*. [S.l.: s.n.], 2014.

LARA, S. M. A. de; OLIVEIRA, D.; FORTES, R. P. de M. Acessibilidade sensível ao contexto para adultos mais velhos em sistemas web. In: *IHC+ CLIHC*. [S.l.: s.n.], 2011. p. 353–357.

LAZAR, J. et al. Web accessibility in the mid-atlantic united states: a study of 50 homepages. *Universal Access in the Information Society*, Springer, v. 2, n. 4, p. 331–341, 2003.

LESINHOVSKI, A. C. *Desenvolvimento de nova proposta de site acessível para o portal terceira idade*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2014.

LIE, H. W.; BOS, B. *Cascading style sheets: designing for the Web*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 1997.

LTD., D. R. *Dark Reader: Dark Mode Everywhere*. 2025. <https://darkreader.org/>. Disponível em: <https://darkreader.org/>.

MELO, I. V. de A. *Um estudo qualitativo sobre arquitetura de software no desenvolvimento de sistemas reais*. 141 p. Monografia (Dissertação) — Universidade Federal de Campina Grande, Paraíba, 2015.

MONITORA. *As principais metodologias de desenvolvimento de software que você precisa saber*. 2020. Disponível em: <https://www.monitoratec.com.br/blog/metodologias-de-desenvolvimento-de-software/>.

MORROW, S. Getting to know WCAG 2.2. *Sharyn Morrow's Blog*, October 2023. Disponível em: <https://www.sharynmorrow.com/2023/10/getting-to-know-wcag-2-2/>.

MUSCIANO, C.; KENNEDY, B.; NOVAK, G. M. *HTML: the definitive guide*. [S.l.]: American Institute of Physics, 1998.

NIELSEN, J. Usability inspection methods. In: *Conference companion on Human factors in computing systems*. [S.l.: s.n.], 1994. p. 413–414.

PEREIRA, C.; TEIXEIRA, J. *Padrões de acessibilidade web utilizado em aplicações ricas para inclusão de deficientes visuais. Monografia*. 2019. Disponível em: <https://monografias.brasilescola.uol.com.br/computacao/padroes-de-acessibilidade-web-utilizado-em-aplicacoes-ricas-para-inclusao-de-deficientes-visuais.htm>.

PRESSMAN, R. *Engenharia de Software, McGrawHill, 6a*. São Paulo: Edição, 2006.

PRESTON, K. Assistive technologies: Principles and practice. *Rehabilitation Nursing*,

Lippincott Williams & Wilkins Ovid Technologies, v. 28, n. 2, p. 64, 2003.

PUPO, D. T.; MELO, A. M.; FERRÉS, S. P. Acessibilidade: discurso e prática no cotidiano das bibliotecas. *Campinas: Biblioteca Central César Lattes*, 2006.

ROCHA, R. G. O. Uso de tablets como ferramenta facilitadora em projetos de inclusão digital de idosos. Universidade Estadual Paulista (UNESP), 2013.

SERVICES, A. W. *The Difference Between Logical and Physical Data Models*. 2025. Disponível em: <https://aws.amazon.com/pt/compare/the-difference-between-logical-and-physical-data-model/#:~:text=O%20modelo%20l%C3%B3gico%20de%20dados,de%20forma%20independente%20da%20plataforma.>

SILVA, D. E. dos S.; SOUZA, I. T. de; CAMARGO, T. Metodologias ágeis para o desenvolvimento de software: Aplicação e o uso da metodologia scrum em contraste ao modelo tradicional de gerenciamento de projetos. *Revista Computação Aplicada-UNG-Ser*, v. 2, n. 1, p. 39–46, 2013.

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

SOUZA, I. de. *Bootstrap: saiba neste guia para iniciantes o que é, por que e como usá-lo*. 2019. Disponível em: <https://rockcontent.com/br/blog/bootstrap/>.

SPRING. *Spring Data JPA*. 2025. Disponível em: <https://spring.io/projects/spring-data-jpa>.

SYSTEMS, D. *Get Started with axe DevTools Browser Extension*. 2025. <https://www.deque.com/get-started-axe-devtools-browser-extension/>. Disponível em: <https://www.deque.com/get-started-axe-devtools-browser-extension/>.

SYSTEMS, I. D. *Get Started with axe DevTools Browser Extension*. 2025. Online documentation. Acessado em 09 de julho de 2025. Disponível em: <https://www.deque.com/get-started-axe-devtools-browser-extension/>.

TECHTUDO. *O que é API e para que serve? Cinco perguntas e respostas*. 2020. Disponível em: <https://www.techtudo.com.br/listas/2020/06/o-que-e-api-e-para-que-serve-cinco-perguntas-e-respostas.ghtml>.

TERA. *Front-end e Back-end: entenda essas duas áreas do desenvolvimento*. 2025. Acesso em: 26 maio 2025. Disponível em: <https://blog.somostera.com/carreiras-digitais/front-end-e-back-end>.

TPGI. *Colour Contrast Analyser (CCA)*. 2025. <https://www.tpgi.com/color-contrast-checker/>. Disponível em: <https://www.tpgi.com/color-contrast-checker/>.

VIEIRA, H.; BARANAUSKAS, M. C. C. Design e avaliação de interfaces humano-computador. *Campinas: Unicamp*, 2003.

W3C. *Web Accessibility Initiative- WAI*. 1997. Disponível em: <https://www.w3.org/WAI/>.

W3C. *WAI-ARIA Overview*. 2006. Disponível em: <https://www.w3.org/WAI/standards-guidelines/aria/>.

W3C. Cartilha de acessibilidade na web. *W3C Brasil, Nic.br, Cgi.br*, v. 1, 2014.

W3C. *Diretrizes de Acessibilidade para Conteúdo Web (WCAG) 2.1*. 2018.

W3C. *Evaluating Web Accessibility Overview*. 2022. Disponível em: <https://www.w3.org/WAI/test-evaluate/>.

W3C. *Introduction to Web Accessibility.Developed by the Education and Outreach Working Group (EOWG)*. 2022. Disponível em: <https://www.w3.org/WAI/fundamentals/accessibility-intro/#important.>

W3C. *Our mission*. 2025. Acesso em: 21 de agosto de 2025. Disponível em: <https://www.w3.org/mission/>.

WADDELL, C. et al. *Constructing accessible web sites*. [S.l.]: Apress, 2003.

WEBAIM. *WAVE Web Accessibility Evaluation Tool*. 2025. Disponível em: <https://wave.webaim.org/>.

World Wide Web Consortium (W3C). *Introduction to Web Accessibility*. 2022. Disponível em: <https://www.w3.org/WAI/fundamentals/accessibility-intro/>.

APÊNDICE A - Critérios de Sucesso - WCAG 2.2

Princípio 1: Perceptível

- **1.1.1 Conteúdo Não Textual (Nível A)**: Todo conteúdo não textual apresentado ao usuário possui uma alternativa em texto que serve ao mesmo propósito.
- **1.2.1 Apenas Áudio e Apenas Vídeo (Pré-gravado) (Nível A)**: Para conteúdo apenas áudio pré-gravado e conteúdo apenas vídeo pré-gravado são fornecidas alternativas.
- **1.2.2 Legendas (Pré-gravadas) (Nível A)**: Legendas são fornecidas para todo conteúdo de áudio pré-gravado em mídia sincronizada.
- **1.2.3 Descrição de Áudio ou Mídia Alternativa (Pré-gravada) (Nível A)**: É fornecida uma descrição de áudio ou mídia alternativa para conteúdo de vídeo pré-gravado.
- **1.2.4 Legendas (Em tempo real) (Nível AA)**: Legendas são fornecidas para todo conteúdo de áudio em tempo real em mídia sincronizada.
- **1.2.5 Descrição de Áudio (Pré-gravada) (Nível AA)**: Descrição de áudio é fornecida para todo conteúdo de vídeo pré-gravado.
- **1.2.6 Língua de Sinais (Pré-gravada) (Nível AAA)**: Interpretação em língua de sinais é fornecida para todo conteúdo de áudio pré-gravado.
- **1.2.7 Descrição de Áudio Estendida (Pré-gravada) (Nível AAA)**: Onde as descrições de áudio não podem ser fornecidas devido a pausas inadequadas no áudio principal, é fornecida uma descrição de áudio estendida.
- **1.2.8 Alternativa em Mídia (Pré-gravada) (Nível AAA)**: É fornecida uma alternativa em mídia para toda mídia sincronizada pré-gravada e para toda mídia apenas vídeo pré-gravada.
- **1.2.9 Apenas Áudio (Em tempo real) (Nível AAA)**: É fornecida uma alternativa para mídia apenas áudio em tempo real que transmita informações equivalentes.
- **1.3.1 Informação e Relações (Nível A)**: A informação, estrutura e relações veiculadas através da apresentação podem ser determinadas por software ou estão disponíveis em texto.

- **1.3.2 Sequência com Significado (Nível A)**: Quando a sequência em que o conteúdo é apresentado afeta seu significado, uma sequência de leitura correta pode ser determinada por software.
- **1.3.3 Características Sensoriais (Nível A)**: As instruções fornecidas para entender e operar o conteúdo não dependem exclusivamente de características sensoriais dos componentes.
- **1.3.4 Orientação (Nível AA)**: O conteúdo não restringe sua visualização e operação a uma única orientação de exibição.
- **1.3.5 Identificar o Propósito da Entrada (Nível AA)**: O propósito de cada campo de entrada coletando informações do usuário pode ser determinado programaticamente.
- **1.3.6 Identificar o Propósito (Nível AAA)**: O propósito dos componentes da interface do usuário, ícones e regiões pode ser determinado programaticamente.
- **1.4.1 Uso de Cor (Nível A)**: A cor não é usada como o único meio visual de transmitir informações.
- **1.4.2 Controle de Áudio (Nível A)**: Se algum áudio em uma página web toca automaticamente por mais de 3 segundos, existe um mecanismo para pausar ou parar o áudio.
- **1.4.3 Contraste (Mínimo) (Nível AA)**: A apresentação visual de texto e imagens de texto tem uma relação de contraste de pelo menos 4.5:1.
- **1.4.4 Redimensionamento de Texto (Nível AA)**: O texto pode ser redimensionado até 200% sem perda de conteúdo ou funcionalidade.
- **1.4.5 Imagens de Texto (Nível AA)**: Se as mesmas tecnologias podem ser usadas para alcançar o efeito visual desejado, o texto é usado para transmitir a informação em vez de imagens de texto.
- **1.4.6 Contraste (Melhorado) (Nível AAA)**: A apresentação visual de texto e imagens de texto tem uma relação de contraste de pelo menos 7:1.
- **1.4.7 Áudio de Segundo Plano Baixo ou Nenhum (Nível AAA)**: Para conteúdo pré-gravado apenas áudio, o áudio de primeiro plano é claramente distinguível do áudio de fundo.
- **1.4.8 Apresentação Visual (Nível AAA)**: Para a apresentação visual de blocos de texto, existe um mecanismo para alcançar o seguinte.

- **1.4.9 Imagens de Texto (Sem Exceção) (Nível AAA)**: Imagens de texto são usadas apenas para decoração pura ou quando uma apresentação de texto específica é essencial.
- **1.4.10 Redimensionamento de Texto (Nível AA)**: Texto pode ser redimensionado sem assistência de tecnologia até 200% sem perda de conteúdo ou funcionalidade.
- **1.4.11 Contraste de Componentes Não Textuais (Nível AA)**: Componentes visuais necessários para identificar componentes da interface do usuário e estados têm contraste de pelo menos 3:1.
- **1.4.12 Espaçamento de Texto (Nível AA)**: Em blocos de texto, são fornecidas definições para espaçamento de linha, parágrafo, letra e palavra.
- **1.4.13 Conteúdo em Destaque ou Foco (Nível AA)**: Quando recebe foco ou é destacado, conteúdo adicional se torna visível e depois escondido.

Princípio 2: Operável

- **2.1.1 Teclado (Nível A)**: Toda a funcionalidade do conteúdo é operável através de uma interface de teclado.
- **2.1.2 Sem Armadilha de Teclado (Nível A)**: Se o foco do teclado pode ser movido para um componente da página usando uma interface de teclado, então o foco pode ser movido para fora desse componente usando apenas a interface de teclado.
- **2.1.3 Teclado (Sem Exceção) (Nível AAA)**: Toda a funcionalidade do conteúdo é operável através de uma interface de teclado sem exigir timings específicos para as teclas individuais.
- **2.1.4 Atalhos de Caracteres (Nível A)**: Se um atalho de teclado que usa uma única letra, pontuação, número ou caractere de símbolo é implementado no conteúdo, então pelo menos uma das seguintes opções é verdadeira.
- **2.2.1 Ajuste de Temporização (Nível A)**: Para cada limite de tempo definido pelo conteúdo, pelo menos uma das seguintes opções é verdadeira.
- **2.2.2 Pausar, Parar, Ocultar (Nível A)**: Para conteúdo em movimento, piscando, rolando ou atualizando automaticamente, todas as seguintes opções são verdadeiras.
- **2.2.3 Sem Temporização (Nível AAA)**: O tempo não é uma parte essencial do evento ou atividade apresentada pelo conteúdo.

- **2.2.4 Interrupções (Nível AAA)**: Interrupções podem ser adiadas ou suprimidas pelo usuário.
- **2.2.5 Reautenticação (Nível AAA)**: Quando uma sessão autenticada expira, o usuário pode continuar a atividade sem perda de dados após reautenticação.
- **2.2.6 Timeouts (Nível AA)**: Os usuários são avisados sobre a duração de qualquer inatividade do usuário que possa causar perda de dados.
- **2.3.1 Três flashes ou Abaixo do Limiar (Nível A)**: As páginas web não contêm nada que pisque mais de 3 vezes em qualquer período de 1 segundo.
- **2.3.2 Três Flashes (Nível AAA)**: As páginas web não contêm nada que pisque mais de 3 vezes em qualquer período de 1 segundo.
- **2.3.3 Animação de Interações (Nível AAA)**: A animação de interações pode ser desativada.
- **2.4.1 Ignorar Blocos (Nível A)**: É fornecido um mecanismo para pular blocos de conteúdo repetidos em páginas web.
- **2.4.2 Título da Página (Nível A)**: As páginas web têm títulos que descrevem o tópico ou propósito.
- **2.4.3 Ordem do Foco (Nível A)**: Se uma página web pode ser navegada sequencialmente e a ordem de navegação afeta o significado ou operação, os componentes recebem foco em uma ordem que preserva o significado e a operabilidade.
- **2.4.4 Propósito do Link (No Contexto) (Nível A)**: O propósito de cada link pode ser determinado a partir apenas do texto do link ou do texto do link junto com seu contexto.
- **2.4.5 Múltiplas Formas (Nível AA)**: É fornecida mais de uma forma para localizar uma página web dentro de um conjunto de páginas web.
- **2.4.6 Cabeçalhos e Rótulos (Nível AA)**: Cabeçalhos e rótulos descrevem tópico ou propósito.
- **2.4.7 Foco Visível (Nível AA)**: Qualquer interface operável por teclado tem um modo de operação onde o indicador de foco é visível.
- **2.4.8 Localização (Nível AAA)**: Informações sobre a localização do usuário dentro de um conjunto de páginas web estão disponíveis.
- **2.4.9 Propósito do Link (Apenas Link) (Nível AAA)**: O propósito de cada link pode ser determinado a partir apenas do texto do link.

- **2.4.10 Cabeçalhos de Seção (Nível AAA)**: As seções do conteúdo são organizadas usando cabeçalhos.
- **2.4.11 Foco Não Obscurecido (Mínimo) (Nível AA)**: Quando um componente recebe foco, o componente não é completamente obscurecido por outro conteúdo.
- **2.4.12 Foco Não Obscurecido (Melhorado) (Nível AAA)**: Quando um componente recebe foco, nenhuma parte do componente é obscurecida por outro conteúdo.
- **2.4.13 Interrupção do Foco (Nível AAA)**: As interrupções do foco podem ser adiadas ou suprimidas pelo usuário.
- **2.5.1 Gestos de Ponteiro (Nível A)**: Toda funcionalidade que usa gestos multiponto ou baseados em trajetória pode ser operada com um único ponteiro.
- **2.5.2 Cancelamento de Ponteiro (Nível A)**: Para funcionalidade que pode ser operada usando um único ponteiro, pelo menos uma das seguintes opções é verdadeira.
- **2.5.3 Rótulo no Nome (Nível A)**: Para componentes da interface do usuário com rótulos que incluem texto ou imagens de texto, o nome contém o texto que é apresentado visualmente.
- **2.5.4 Ativação por Movimento (Nível A)**: A funcionalidade que pode ser operada por movimento do dispositivo ou movimento do usuário também pode ser operada por componentes da interface do usuário.
- **2.5.5 Tamanho do Alvo (Nível AAA)**: Os alvos de ponteiro têm pelo menos 44 por 44 pixels CSS.
- **2.5.6 Entrada Simultânea de Mecanismos de Entrada (Nível AAA)**: O conteúdo não restringe o uso de modalidades de entrada disponíveis em uma plataforma.
- **2.5.7 Movimento de Agitação (Nível A)**: A funcionalidade que pode ser operada por agitação do dispositivo também pode ser operada por componentes da interface do usuário.
- **2.5.8 Alvo de Ponteiro (Melhorado) (Nível AA)**: Para conteúdo da web que tenha alvos de ponteiro, pelo menos um dos seguintes é verdadeiro.

Princípio 3: Compreensível

- **3.1.1 Idioma da Página (Nível A)**: O idioma humano padrão de cada página web pode ser determinado programaticamente.

- **3.1.2 Idioma de Partes (Nível AA)**: O idioma humano de cada passagem ou frase no conteúdo pode ser determinado programaticamente.
- **3.1.3 Palavras Incomuns (Nível AAA)**: É fornecida uma definição para palavras incomuns.
- **3.1.4 Abreviações (Nível AAA)**: É fornecida uma definição para abreviações.
- **3.1.5 Nível de Leitura (Nível AAA)**: Quando o texto exige habilidade de leitura mais avançada que o nível de educação secundária inferior, é fornecida versão suplementar.
- **3.1.6 Pronúncia (Nível AAA)**: É fornecida uma pronúncia para palavras onde o significado é ambíguo sem saber a pronúncia.
- **3.2.1 Foco (Nível A)**: Quando qualquer componente recebe foco, não inicia uma mudança de contexto.
- **3.2.2 Entrada (Nível A)**: Alterar a configuração de qualquer componente da interface do usuário não causa automaticamente uma mudança de contexto.
- **3.2.3 Navegação Consistente (Nível AA)**: Os componentes de navegação que são repetidos em múltiplas páginas web ocorrem na mesma ordem relativa.
- **3.2.4 Identificação Consistente (Nível AA)**: Componentes que têm a mesma funcionalidade dentro de um conjunto de páginas web são identificados consistentemente.
- **3.2.5 Mudança a Pedido (Nível AAA)**: Mudanças de contexto são iniciadas apenas por solicitação do usuário.
- **3.3.1 Identificação de Erro (Nível A)**: Se um erro de entrada é detectado automaticamente, o item que está em erro é identificado e o erro é descrito ao usuário em texto.
- **3.3.2 Rótulos ou Instruções (Nível A)**: Rótulos ou instruções são fornecidos quando o conteúdo requer entrada do usuário.
- **3.3.3 Sugestão de Erro (Nível AA)**: Se um erro de entrada é detectado automaticamente e sugestões de correção são conhecidas, então as sugestões são fornecidas ao usuário.
- **3.3.4 Prevenção de Erro (Legal, Financeiro, Dados) (Nível AA)**: Para páginas web que causam compromissos legais ou transações financeiras, que modificam ou apagam dados controláveis pelo usuário, ou que enviam respostas de teste do usuário, pelo menos uma das seguintes opções é verdadeira.

- **3.3.5 Ajuda (Nível AAA)**: A ajuda contextual está disponível.
- **3.3.6 Prevenção de Erro (Todos) (Nível AAA)**: Para páginas web que requerem que o usuário envie informações, pelo menos uma das seguintes opções é verdadeira.
- **3.3.7 Status de Envio (Nível AA)**: Para cada envio que resulte em uma mudança legal ou transação financeira, ou que modifique ou exclua dados controláveis pelo usuário, ou que envie respostas de teste do usuário, o status da operação é indicado.
- **3.3.8 Acesso à Reversão (Nível AAA)**: É fornecida uma maneira de reverter envios.
- **3.3.9 Acesso à Confirmação (Nível AAA)**: É fornecida uma maneira de confirmar envios.

Princípio 4: Robusto

- **4.1.1 Processamento (Nível A)**: (Removido da WCAG 2.2) Em conteúdo implementado usando linguagens de marcação, os elementos têm tags de início e fim completas, os elementos são aninhados de acordo com suas especificações, os elementos não contêm atributos duplicados e qualquer ID é único.
- **4.1.2 Nome, Função, Valor (Nível A)**: Para todos os componentes da interface do usuário, o nome e a função podem ser determinados programaticamente.
- **4.1.3 Mensagens de Status (Nível AA)**: Em conteúdo implementado usando linguagens de marcação, as mensagens de status podem ser determinadas programaticamente através de funções ou propriedades.
- **4.1.4 Identificação de Propósito (Nível AA)**: Em conteúdo implementado usando linguagens de marcação, o propósito de muitos campos de entrada de usuário pode ser determinado programaticamente.

APÊNDICE B - Narrativas de casos de uso

UC002 - Gerenciar Coadministradores

Descrição: Este caso de uso descreve como o administrador master interage com o sistema para registrar novos coadministradores e excluir.

Atores: Administrador Master.

Pré-condições:O administrador master deve estar logado no sistema.

Pós-condições: Um coadministrador é criado, excluído ou exibido conforme a ação do ator.

Fluxo Normal (Administrador)

Registrar Coadministrador

1. O sistema exibe a área de gerenciamento de coadministradores.
2. O administrador master seleciona a ação desejada: registrar novo coadministrador ou excluir.
3. O sistema solicita o nome do usuário.
4. O sistema solicita o user do usuário.
5. O sistema solicita a senha do usuário.
6. O sistema solicita novamente a senha do usuário..
7. O administrador clica no botão de cadastrar novo coadministrador.
8. O sistema salva o novo usuário no banco de dados.
9. O sistema confirma a operação com uma mensagem de sucesso.

Excluir Coadministrador

1. O sistema exibe a área de gerenciamento de coadministradores.
2. O administrador master seleciona a ação desejada: registrar novo coadministrador ou excluir.
3. O sistema exibe a lista de coadministradores cadastrados.
4. O administrador master seleciona um coadministrador para realizar a exclusão.

5. O sistema solicita confirmação da exclusão ao administrador master.
6. O administrador master confirma a exclusão.
7. O sistema salva a alteração no banco de dados.
8. O sistema confirma a operação com uma mensagem de sucesso.

Fluxos Alternativos

FA01 – Campo obrigatório não preenchido ao registrar novo coadministrador.

1. O sistema exibe a mensagem MSG001.
2. O administrador master preenche o campo obrigatório.
3. Retorna ao fluxo normal no passo 07.

FA02 – Campo de senha não está conforme o padrão de segurança.

1. O sistema exibe a mensagem MSG004.
2. O administrador master preenche o campo conforme padrão.
3. Retorna ao fluxo normal no passo 07.

FA03 – Campo de senha e senha novamente não coincidem.

1. O sistema exibe a mensagem MSG006.
2. O administrador master preenche o campo conforme padrão.
3. Retorna ao fluxo normal no passo 07.

Fluxos de Exceção

FE01 – Falha ao salvar novo coadministrador no banco de dados.

1. O sistema exibe a mensagem MSG002.
2. O sistema encerra a operação.

FE02 – Falha ao excluir coadministrador no banco de dados.

1. O sistema exibe a mensagem MSG003.
2. O sistema encerra a operação.

FE03 – Falha ao carregar a lista de coadministradores.

1. O sistema exibe a mensagem MSG005.
2. O sistema encerra a operação.

Mensagens do Sistema:

- MSG001: "Preencha o campo obrigatório."
- MSG002: "Não foi possível realizar o registro. Tente novamente!"
- MSG003: "Não foi possível excluir. Tente novamente!"
- MSG004: "A senha não atende aos requisitos mínimos de segurança."
- MSG005: "Não foi possível carregar os usuários. Tente novamente!"
- MSG006: "As senhas não coincidem. Por favor, verifique."

UC003 - Gerenciar Informações de Contato

Descrição: Este caso de uso descreve como o administrador master gerência e como o usuário público acessa as informações de contato de usuários registrados ou da organização responsável no sistema.

Atores: Administrador Master e Usuário Público.

Pré-condições:O administrador Master deve estar logado no sistema.

Pós-condições: As informações de contato são atualizadas no banco de dados (administrador master) e as informações de contato são exibidas corretamente (usuário público).

Fluxo Normal (Administrador Master)

1. O sistema exibe a área de gerenciamento de informações de contato.
2. O administrador seleciona um dos registros de contato para editar (e-mail, telefone, rede social).
3. O sistema exibe os campos, conforme a seleção do registro de contato.
4. O administrador master atualiza ou corrige os campos desejados.
5. O administrador master clica no botão de salvar alterações.
6. O sistema salva as informações atualizadas no banco de dados.

7. O sistema confirma a operação com uma mensagem de sucesso.

Fluxo Normal (Usuário Público)

1. O usuário público acessa a seção de contatos.
2. O sistema exibe as informações de contato do administrador ou da organização (e-mail, telefone, endereço, rede social).

Fluxos Alternativos

FA01 – Campo obrigatório não preenchido ao atualizar informações de contato.

1. O sistema exibe a mensagem MSG001.
2. O administrador master preenche o campo obrigatório.
3. Retorna ao fluxo normal no passo 05.

FA02 – Adição de mais de 3 registros de contato por categoria (e-mail, telefone, rede social).

1. O sistema exibe a mensagem MSG002.
2. O administrador master preenche o campo corretamente.
3. Retorna ao fluxo normal no passo 05.

FA03 – Número de telefone inserido inválido.

1. O sistema exibe a mensagem MSG003.
2. O administrador master preenche o campo corretamente.
3. Retorna ao fluxo normal no passo 05.

FA03 – E-mail inserido inválido.

1. O sistema exibe a mensagem MSG004.
2. O administrador master preenche o campo corretamente.
3. Retorna ao fluxo normal no passo 05.

Fluxos de Exceção

FE01 – Falha ao salvar informações de contato no banco de dados.

1. O sistema exibe a mensagem MSG005.
2. O sistema encerra a operação.

Mensagens do Sistema:

- MSG001: "Preencha o campo obrigatório."
- MSG002: "Você pode cadastrar no máximo 3"
- MSG003: "O campo 'telefone' está em um formato inválido.. Formato esperado: (XX) XXXX-XXXX ou (XX) 9XXXX-XXXX"
- MSG004: "O campo 'e-mail' está em um formato inválido."
- MSG005: "Erro ao salvar informações de contato. Tente novamente!"

UC004 - Gerenciar Agendamento de Visita

Descrição: Este caso de uso descreve como o administrador master gerência e como o usuário público pode solicitar um agendamento de visita através do sistema.

Atores: Administrador Master e Usuário Público.

Pré-condições:O administrador Master deve estar logado no sistema.

Pós-condições: A solicitação de visita será aprovada ou rejeitada (administrador master) e uma solicitação de agendamento de visita será cadastrada (usuário público).

Fluxo Normal (Administrador Master)

1. O sistema exibe a área de gerenciamento de agendamento de visitas.
2. O administrador seleciona um dos registros de contato para aprovar ou rejeitar.
3. O administrador master clica no botão de salvar alterações.
4. O sistema salva as informações atualizadas no banco de dados.
5. O sistema confirma a operação com uma mensagem de sucesso.

Fluxo Normal (Usuário Público)

1. O usuário público acessa o formulário de agendamento de visita.
2. O usuário público preenche os campos solicitados (Nome, telefone, E-mail, Data sugerida, Objetivo da visita)

3. O usuário público clica no botão de enviar solicitação.
4. O sistema salva as informações atualizadas no banco de dados e envia um e-mail para os responsáveis do sistema com as informações preenchidas no formulário.
5. O sistema confirma a operação com uma mensagem de sucesso.

Fluxos Alternativos

FA01 – Campo obrigatório não preenchido no formulário de solicitação

1. O sistema exibe a mensagem MSG001.
2. O usuário público preenche o campo obrigatório.
3. Retorna ao fluxo normal no passo 03.

FA02 – Número de telefone inserido inválido.

1. O sistema exibe a mensagem MSG002.
2. O usuário público preenche o campo corretamente.
3. Retorna ao fluxo normal no passo 03.

FA03 – E-mail inserido inválido.

1. O sistema exibe a mensagem MSG003.
2. O usuário público preenche o campo corretamente.
3. Retorna ao fluxo normal no passo 03.

FA04 – Data inserida é inferior ao dia atual.

1. O sistema exibe a mensagem MSG004.
2. O usuário público preenche o campo corretamente.
3. Retorna ao fluxo normal no passo 03.

Fluxos de Exceção

FE01 – Falha ao enviar solicitação de agendamento no banco de dados.

1. O sistema exibe a mensagem MSG005.
2. O sistema encerra a operação.

FE02 – Falha ao aprovar ou rejeitar agendamento.

1. O sistema exibe a mensagem MSG006.
2. O sistema encerra a operação.

FE03 – Falha ao carregar a lista de agendamentos.

1. O sistema exibe a mensagem MSG007.
2. O sistema encerra a operação.

Mensagens do Sistema:

- MSG001: "Preencha o campo obrigatório."
- MSG002: "O campo 'telefone' está em um formato inválido.. Formato esperado: (XX) XXXX-XXXX ou (XX) 9XXXX-XXXX"
- MSG003: "E-mail inválido."
- MSG004: "A data deve ser futura"
- MSG005: "Erro ao enviar sua solicitação. Tente novamente!"
- MSG006: "Não foi possível alterar o status. Tente novamente!"
- MSG007: "Não foi possível carregar os agendamentos. Tente novamente!"

UC005 - Gerenciar Publicações

Descrição: Este caso de uso descreve como o administrador master gerencia itens no acervo do sistema, incluindo a adição, modificação ou remoção de itens, o coadministrador adiciona itens e como o usuário público navega e visualiza o conteúdo das publicações.

Atores: Administrador Master, Coadministrador e Usuário Público.

Pré-condições:O administrador master e o coadministrador devem estar logado no sistema e o sistema deve conter um subprojeto previamente cadastrado para visualização.

Pós-condições: Um item é adicionado (administrador master e coadministrador), modificado ou removido no banco de dados (administrador master) e o conteúdo da publicação é exibido corretamente para o usuário público.

Fluxo Normal (Administrador Master e Coadministrador)

Criar publicação (Administrador Master e Coadministrador)

1. O sistema exibe a área de gerenciamento das publicações.

2. O administrador master seleciona a ação desejada: adicionar, modificar ou remover um item.
3. O sistema solicita fotos do item.
4. O sistema solicita o título do item.
5. O sistema solicita a data vinculada ao item.
6. O sistema solicita o nome do autor associado ao item.
7. O sistema solicita a localização do item.
8. O sistema solicita a descrição do item.
9. O administrador master clica no botão de salvar item.
10. O sistema salva o novo item no banco de dados.
11. O sistema confirma a operação com uma mensagem de sucesso.

Editar publicação (Administrador Master)

1. O sistema exibe a área de gerenciamento de publicação.
2. O administrador master seleciona a ação desejada: adicionar, modificar ou remover um item.
3. O sistema exibe a lista de itens disponíveis no acervo.
4. O administrador master seleciona um item para modificar.
5. O sistema exibe os campos atuais do item (título, foto, data, nome do autor, localização e descrição).
6. O administrador master altera os campos desejados.
7. O administrador master clica no botão de salvar alterações.
8. O sistema atualiza as informações do item no banco de dados.
9. O sistema confirma a operação com uma mensagem de sucesso.

Excluir Publicação (Administrador Master)

1. O sistema exibe a área de gerenciamento de publicação.
2. O administrador master seleciona a ação desejada: adicionar, modificar ou remover um item.

3. O sistema exibe a lista de itens disponíveis no acervo.
4. O administrador master seleciona um item para remover.
5. O sistema exibe uma mensagem de confirmação.
6. O administrador master confirma a remoção.
7. O sistema remove o item do banco de dados.
8. O sistema confirma a operação com uma mensagem de sucesso.

Fluxo Normal (Usuário Público)

1. O sistema exibe os subprojetos e o usuário seleciona o que deseja.
2. O usuário público acessa a lista de publicações vinculadas ao subprojeto.
3. O sistema exibe os itens com título, foto, nome do autor, data e localização.
4. O usuário público pode:
   - Pesquisar itens específicos utilizando nome e/ou localização.
5. O sistema exibe os itens correspondentes à pesquisa ou ao filtro.
6. O usuário público seleciona um item para visualizar mais detalhes.
7. O sistema exibe informações completas sobre o item selecionado (título, foto, data, nome do autor, localização e descrição).

Fluxos Alternativos

FA01 – Campo obrigatório não preenchido ao adicionar item.

1. O sistema exibe a mensagem MSG001.
2. O administrador master ou coadministrador preenche o campo obrigatório.
3. Retorna ao fluxo normal no passo 09.

FA02 – Campo obrigatório não preenchido ao modificar item.

1. O sistema exibe a mensagem MSG001.
2. O administrador master preenche o campo obrigatório.
3. Retorna ao fluxo normal no passo 09.

FA03 – Campo de data deve ser menor que a data atual.

1. O sistema exibe a mensagem MSG002.
2. O administrador master ou coadministrador preenche o campo obrigatório.
3. Retorna ao fluxo normal no passo 09.

Fluxos de Exceção

FE01 – Falha ao salvar novo item no banco de dados.

1. O sistema exibe a mensagem MSG003.
2. O sistema encerra a operação.

FE02 – Falha ao remover item do banco de dados.

1. O sistema exibe a mensagem MSG004.
2. O sistema encerra a operação.

Mensagens do Sistema:

- MSG001: "Preencha o campo obrigatório."
- MSG002: "A data deve ser no passado"
- MSG003: "Erro ao salvar publicação. Tente novamente!"
- MSG004: "Ocorreu um erro ao tentar excluir o item. Tente novamente!"

UC006 - Gerenciar Colaboradores

Descrição: Este caso de uso descreve como o administrador master e o usuário público interagem com o sistema para gerenciar e visualizar colaboradores.

Atores: Administrador Master e Usuário Público.

Pré-condições:O administrador master deve estar logado no sistema.

Pós-condições: Um colaborador é criado, editado, excluído ou exibido conforme a ação do ator.

Fluxo Normal (Administrador Master)

Criar Colaborador

1. O sistema exibe a tela de gerenciamento de colaboradores.

2. O administrador master seleciona a ação desejada: criar, editar ou excluir colaborador.
3. O sistema solicita o nome do colaborador.
4. O sistema solicita o tipo do colaborador (Coordenadora ou bolsista).
5. O sistema solicita a foto do colaborador.
6. O sistema solicita a descrição do colaborador.
7. O sistema solicita os projetos que o colaborador é associado.
8. O administrador master clica no botão de salvar colaborador.
9. O sistema salva as informações do novo colaborador no banco de dados.
10. O sistema confirma a operação com uma mensagem de sucesso.

Editar colaborador

1. O sistema exibe a tela de gerenciamento de colaborador.
2. O administrador master seleciona a ação desejada: criar, editar ou excluir colaborador
3. O sistema exibe a lista de colaboradores disponíveis.
4. O administrador master seleciona um colaborador para editar.
5. O sistema exibe as informações atuais do colaborador.
6. O administrador master altera os campos desejados (nome, tipo, foto, descrição ou projetos associados).
7. O administrador master clica no botão de salvar alterações.
8. O sistema atualiza as informações do colaborador no banco de dados.
9. O sistema confirma a operação com uma mensagem de sucesso.

Excluir colaborador

1. O sistema exibe a tela de gerenciamento de colaborador.
2. O administrador master seleciona a ação desejada: criar, editar ou excluir colaborador.
3. O sistema exibe a lista de colaboradores disponíveis.
4. O administrador master seleciona um colaborador para excluir.
5. O sistema exibe uma mensagem de confirmação.

6. O administrador master confirma a exclusão.
7. O sistema remove o colaborador do banco de dados.
8. O sistema confirma a operação com uma mensagem de sucesso.

Fluxo Normal (Usuário Público)

1. O sistema exibe a lista de colaboradores disponíveis e suas respectivas informações de: nome, tipo, foto, descrição e projetos associados

Fluxos Alternativos

FA01 – Campo obrigatório não preenchido ao criar colaborador.

1. O sistema exibe a mensagem MSG001.
2. O administrador master preenche o campo obrigatório.
3. Retorna ao fluxo normal no passo 08.

FA02 – Campo obrigatório não preenchido ao editar colaborador.

1. O sistema exibe a mensagem MSG001.
2. O administrador master preenche o campo obrigatório.
3. Retorna ao fluxo normal no passo 08.

Fluxos de Exceção

FE01 – Falha ao salvar colaborador no banco de dados.

1. O sistema exibe a mensagem MSG002.
2. O sistema encerra a operação.

FE02 – Falha ao salvar alterações no banco de dados.

1. O sistema exibe a mensagem MSG002.
2. O sistema encerra a operação.

FE03 – Falha ao excluir colaborador do banco de dados.

1. O sistema exibe a mensagem MSG002.

2. O sistema encerra a operação.

Mensagens do Sistema:

- MSG001: "Preencha o campo obrigatório."
- MSG002: "Erro ao processar solicitação"

UC007 - Realizar Login

Descrição: Este caso de uso descreve como o administrador master e o coadministrador podem realizar login no sistema.

Atores: Administrador Master e Coadministrador.

Pré-condições:Não há.

Pós-condições: O usuário é autenticado no sistema.

Fluxo Normal (Administrador Master e Coadministrador)

1. O sistema exibe a tela de login.
2. O sistema solicita o nome de usuário.
3. O sistema solicita a senha de acesso.
4. O administrador master ou coadministrador clica no botão de logar.
5. O sistema autentica o usuário.
6. O sistema confirma a operação com uma mensagem de sucesso.

Fluxos Alternativos

FA01 – Campo obrigatório não preenchido ao logar.

1. O sistema exibe a mensagem MSG001.
2. O administrador master ou coadministrador preenche o campo obrigatório.
3. Retorna ao fluxo normal no passo 04.

Fluxos de Exceção

FE01 – Falha ao realizar o login

1. O sistema exibe a mensagem MSG002.
2. O sistema encerra a operação.

Mensagens do Sistema:

- MSG001: “Preencha o campo obrigatório.”
- MSG002: “Credenciais incorretas, tente novamente.”

APÊNDICE C - Dicionário de dados

Modelo Lógico			Modelo Físico	
Atributo	Tipo de atributo	Descrição	Nome da variável	Tipo de variável
ID do Usuário	INTEIRO	Identificador único do usuário	id_usuario	INT
Nome do Usuário	TEXTO	Nome do usuário	nome	VARCHAR
User do Usuário	TEXTO	Identificador do usuário para login no sistema	user	VARCHAR
Senha do Usuário	TEXTO	Senha de acesso ao sistema	senha	VARCHAR
Administrador	BOOLEANO	Indica se o usuário é administrador	is_admin	BOOLEAN

Quadro 18 – Quadro do banco de dados - Usuário

Fonte: Do Autor

Modelo Lógico			Modelo Físico	
Atributo	Tipo de atributo	Descrição	Nome da variável	Tipo de variável
ID do Formulário	INTEIRO	Identificador único do formulário	id_formulario	INT
Nome do Formulário	TEXTO	Nome da pessoa que está enviando o formulário	nome	VARCHAR
E-mail do Formulário	TEXTO	E-mail da pessoa que está enviando o formulário	email	VARCHAR
Data do Formulário	DATA	Data que a pessoa deseja marcar a visita	data	DATE
Data Marcada	DATA	Data que foi confirmada para a visita	data_marcada	DATE
Descrição do Formulário	TEXTO	Descrição enviada no formulário	descricao	VARCHAR
Status do Formulário	ENUMERAÇÃO	Status para indicar se a visita foi agendada ou rejeitada	status	ENUM
Telefone do Formulário	TEXTO	Telefone da pessoa que está enviando o formulário	telefone	VARCHAR

Quadro 19 – Quadro do banco de dados - Formulário

Fonte: Do Autor

Modelo Lógico			Modelo Físico	
Atributo	Tipo de atributo	Descrição	Nome da variável	Tipo de variável
ID da Publicação	INTEIRO	Identificador único da publicação	id_publicacao	INT
Referência ao Subprojeto	INTEIRO	Chave estrangeira que referencia o subprojeto	fk_id_subprojeto	INT
Título da Publicação	TEXTO	Título da publicação	titulo	VARCHAR
Descrição da Publicação	TEXTO	Descrição da publicação	descricao	VARCHAR
Nome do Responsável	TEXTO	Nome do responsável por registrar a foto	nomeResponsavel	VARCHAR
Localização da Foto	TEXTO	Localização onde a foto foi tirada	localizacao	VARCHAR
Data da Memória	DATA	Data do evento registrado na publicação	data_memoria	DATE

Quadro 20 – Quadro do banco de dados - Publicação

Fonte: Do Autor

Modelo Lógico			Modelo Físico	
Atributo	Tipo de atributo	Descrição	Nome da variável	Tipo de variável
ID da Foto	INTEIRO	Identificador único da foto	id_foto	INT
Referência à Publicação	INTEIRO	Chave estrangeira que referencia a publicação	fk_id_publicacao	INT
Imagem	IMAGEM	Arquivo de imagem que fará parte da publicação	imagem	LONGBLOB

Quadro 21 – Quadro do banco de dados - Foto

Fonte: Do Autor

Modelo Lógico			Modelo Físico	
Atributo	Tipo de atributo	Descrição	Nome da variável	Tipo de variável
ID do Colaborador	INTEIRO	Identificador único do colaborador	id_colaborador	INT
Nome do Colaborador	TEXTO	Nome do colaborador	nome	VARCHAR
Foto do Colaborador	IMAGEM	Foto do colaborador	foto	LONGBLOB
Descrição do Colaborador	TEXTO	Descrição do perfil do colaborador	descricao	VARCHAR
Tipo de Colaborador	ENUMERAÇÃO	Tipo de colaborador (ex.: professor, aluno, voluntário)	tipo	ENUM

Quadro 22 – Quadro do banco de dados - Colaborador

Fonte: Do Autor

Modelo Lógico			Modelo Físico	
Atributo	Tipo de atributo	Descrição	Nome da variável	Tipo de variável
ID	INTEIRO	Identificador único do registro Colaborador-Subprojeto	id	INT
Referência ao Subprojeto	INTEIRO	Chave estrangeira que referencia o subprojeto	fk_id_subprojeto	INT
Referência ao Colaborador	INTEIRO	Chave estrangeira que referencia o colaborador	fk_id_colaborador	INT

Quadro 23 – Quadro do banco de dados - Colaborador-Subprojeto

Fonte: Do Autor

Modelo Lógico			Modelo Físico	
Atributo	Tipo de atributo	Descrição	Nome da variável	Tipo de variável
ID do Email	INTEIRO	Identificador único do e-mail	id_email	INT
Endereço de Email	TEXTO	Endereço de e-mail do colaborador ou entidade	email	VARCHAR

Quadro 24 – Quadro do banco de dados - Email

Fonte: Do Autor

Modelo Lógico			Modelo Físico	
Atributo	Tipo de atributo	Descrição	Nome da variável	Tipo de variável
ID da Rede Social	INTEIRO	Identificador único da rede social	id_rede_social	INT
Nome da Rede Social	TEXTO	Nome da rede social (ex.: Facebook, Instagram)	nome	VARCHAR
URL da Rede Social	TEXTO	URL ou link do perfil da rede social	url	VARCHAR

Quadro 25 – Quadro do banco de dados - Rede Social

Fonte: Do Autor

Modelo Lógico			Modelo Físico	
Atributo	Tipo de atributo	Descrição	Nome da variável	Tipo de variável
ID do Telefone	INTEIRO	Identificador único do telefone	id_telefone	INT
Número de Telefone	TEXTO	Número de telefone do colaborador ou entidade	telefone	VARCHAR

Quadro 26 – Quadro do banco de dados - Telefone

Fonte: Do Autor

APÊNDICE D - Telas do Sistema

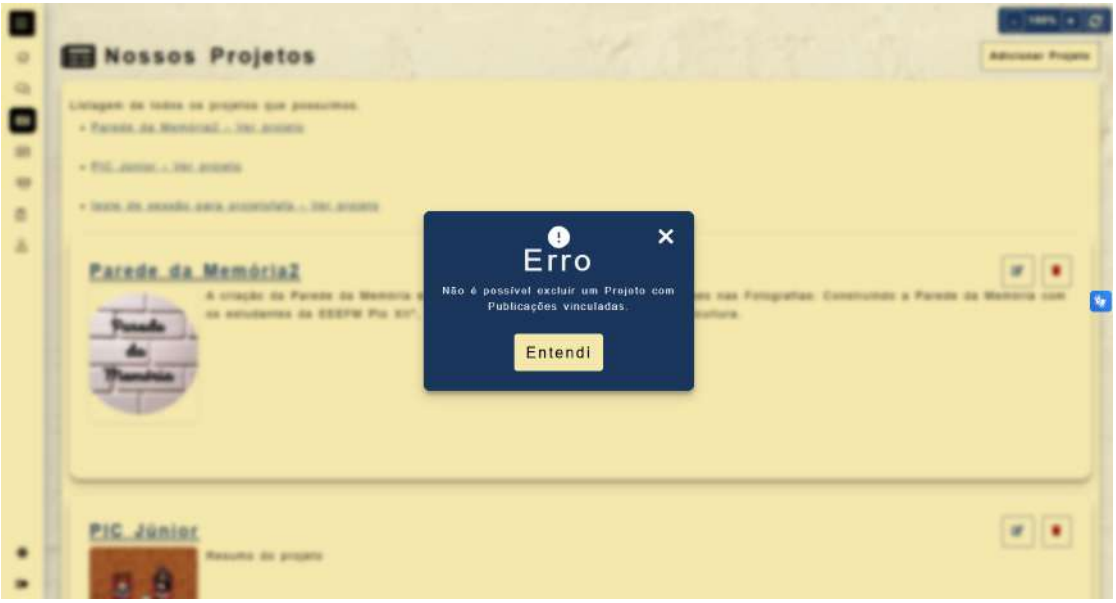

Figura 77 – Imagem da tela da modal de erro ao excluir projeto com publicações vinculadas.

Fonte: Do Autor

Figura 78 – Imagem da tela da modal de erro ao excluir projeto com colaboradores vinculados.

Fonte: Do Autor

Figura 79 – Imagem da tela de Publicações ao realizar uma filtragem sem retorno.

Fonte: Do Autor

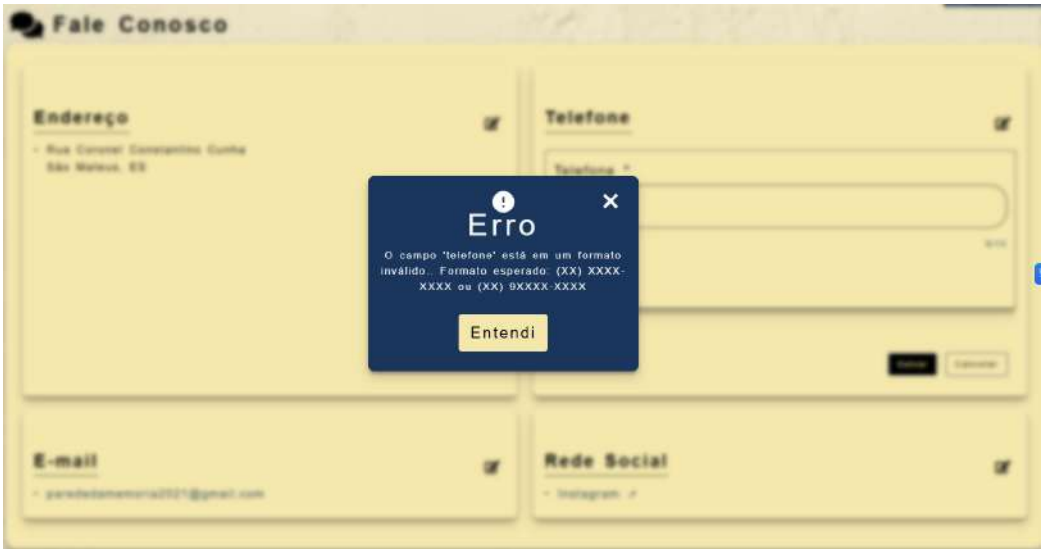

Figura 80 – Imagem da tela da modal de erro ao preencher campo de telefone incorretamente.

Fonte: Do Autor

Figura 81 – Imagem da tela da modal de confirmação ao sair da tela sem salvar as informações.

Fonte: Do Autor

Figura 82 – Imagem da tela de cadastro de usuários demonstrando erros de campo obrigatório no formulário.

Fonte: Do Autor

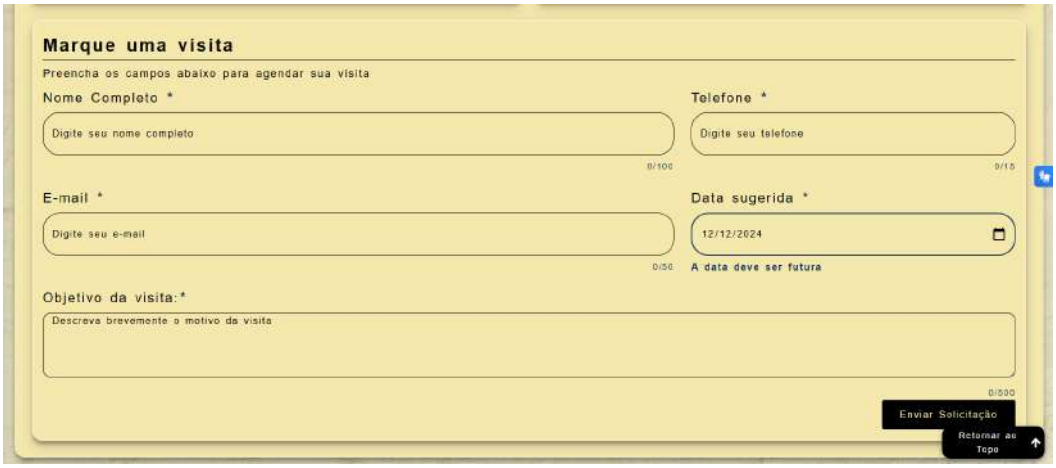

Figura 83 – Imagem da tela do formulário para solicitar uma visita demonstrando erro do campo data futura.

Fonte: Do Autor

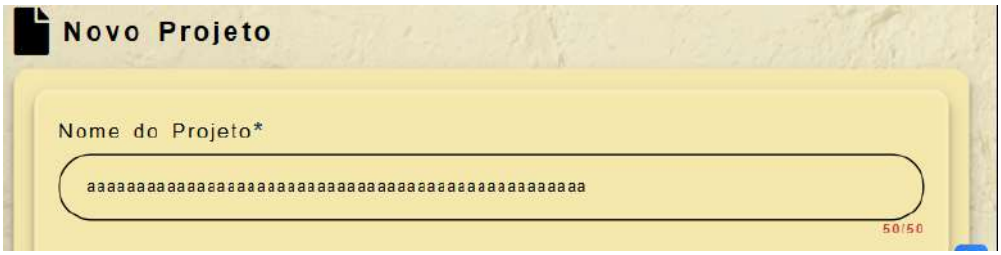

Figura 84 – Imagem da tela do formulário para adicionar um projeto demonstrando a alteração quando o limite de caracteres é alcançado.

Fonte: Do Autor