



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA CENTRAL
TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

HALYSON ITALLO CUNHA PIMENTEL

**DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA A GESTÃO DE
INFORMAÇÕES DA ACADEMIA DE CIÊNCIAS DO PIAUÍ (ACIPI)**

TERESINA, PIAUÍ
2025

HALYSON ITALLO CUNHA PIMENTEL

DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA A GESTÃO DE
INFORMAÇÕES DA ACADEMIA DE CIÊNCIAS DO PIAUÍ (ACIPI)

Trabalho de Conclusão de Curso (Relatório Técnico de Software) apresentado como exigência parcial para obtenção do diploma do Curso de Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central.

Orientador: Prof. M^º. Francisco Marcelino Almeida de Araújo

TERESINA, PIAUÍ
2025

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Pimentel, Halyson Itallo Cunha

P644d Desenvolvimento de uma aplicação web para a gestão de informações da academia de ciências do Piauí (CACIPI) / Halyson Itallo Cunha Pimentel. - 2025.

54 f.: il. color.

Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central, 2025.

Orientador : Prof Me. Francisco Marcelino Almeida de Araújo.

1. ACIPI. 2. Academia de ciências. 3. Sistema Web. 4. Público jovem.

I.Título.

CDD - 004.21

Elaborado por Tanize Maria Sales CRB 3/1024

HALYSON ITALLO CUNHA PIMENTEL

DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA A GESTÃO DE
INFORMAÇÕES DA ACADEMIA DE CIÊNCIAS DO PIAUÍ (ACIPI)

Trabalho de Conclusão de Curso (Relatório Técnico de Software) apresentado como exigência parcial para obtenção do diploma do Curso de Tecnólogo em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Teresina Central.

BANCA EXAMINADORA:

**Prof. M^e. Francisco Marcelino
Almeida de Araújo**

Instituto Federal de Educação,
Ciência e Tecnologia do Piauí -
IFPI

**Prof. M^e. Fernando Castelo
Branco Goncalves Santana**

Instituto Federal de Educação,
Ciência e Tecnologia do Piauí -
IFPI

**Antonio Kauê Cavalcante
Ferreira**

Instituto Federal de Educação,
Ciência e Tecnologia do Piauí -
IFPI

Matheus Araujo Dantas

Instituto Federal de Educação,
Ciência e Tecnologia do Piauí -
IFPI

Aos meus pais.

Aos meus educadores.

A todos que sempre me apoiaram.

AGRADECIMENTOS

Agradeço a todas as pessoas que estiveram ao meu lado durante esta jornada acadêmica, pois sem o apoio delas, este trabalho não seria possível.

Aos meus professores, pela orientação, conhecimento compartilhado e pela paciência ao responder às minhas dúvidas. Seu apoio foi essencial para o meu crescimento acadêmico e profissional.

Aos meus pais, pelo apoio constante e sacrifícios feitos para que eu pudesse alcançar meus objetivos. Seu apoio moral e material foram fundamentais em todos os momentos.

À minha namorada, por ser minha companheira de jornada, compreensiva e motivadora. Seu amor e encorajamento foram a luz que iluminou meu caminho, tornando cada desafio mais fácil de superar.

Aos amigos e companheiros do LABIRAS, expresso minha sincera gratidão por estarem sempre presentes, compartilhando não apenas os momentos de estudo, mas também de descontração e apoio mútuo. Em particular, gostaria de agradecer ao Kauê, Paulo e Diego por tornarem cada desafio enfrentado mais leve com sua união e risadas. Cada momento no laboratório foi enriquecido pela colaboração e amizade que compartilhamos, fundamentais para o sucesso deste trabalho e de vários outros. Sou profundamente grato pela oportunidade de contar com a presença de todos vocês ao meu lado.

Ao meu orientador, pela dedicação, orientação precisa e ensinamentos valiosos que levarei comigo pelo resto da vida. Sua orientação foi fundamental para a concretização deste projeto.

E, por último, mas não menos importante, agradeço a Deus por guiar meus passos, me fortalecer nos momentos de dificuldade e me abençoar com as oportunidades que surgiram ao longo dessa trajetória.

A todos vocês, meu mais profundo agradecimento.

*“A maioria dos bons programadores faz programação
não porque eles esperam ser pagos
ou obter adulação pelo público,
mas porque é divertido programar”.*
Linus Torvald

RESUMO

A ciência e o método científico transformaram a maneira como entendemos o mundo, marcando um divisor de águas no pensamento humano. No Brasil, a criação das primeiras academias de ciências e o crescimento do cenário científico no Piauí deram vida à Academia de Ciências do Piauí (ACIPI). Mais do que uma instituição, a ACIPI se tornou uma fonte de inspiração, conectando pessoas à busca por descobertas e inovações que impactam o nosso dia a dia. Porém, muitas vezes, o universo científico pode parecer distante ou intimidador, especialmente para os jovens, que podem não enxergar as oportunidades à sua volta. Para mudar essa realidade, a ACIPI se uniu ao *Laboratory of Intelligent Robotics, Automation, and Systems* (LABIRAS) para criar um sistema web. Seu objetivo era tornar as informações da academia mais acessíveis e atraentes, especialmente para o público jovem. O resultado foi uma plataforma interativa, construída com um visual agradável e totalmente funcional, especialmente voltado para atrair o público jovem. Sua implementação contou com a participação ativa dos membros da ACIPI, resultando em uma plataforma completa para gestão de conteúdo e promoção das atividades da academia.

Palavras-chaves: academia de ciências do piauí; sistema web; público jovem.

ABSTRACT

Science and the scientific method have transformed the way we understand the world, marking a turning point in human thought. In Brazil, the establishment of the first science academies and the growth of the scientific landscape in Piauí gave rise to the Academia de Ciências do Piauí (ACIPI). More than just an institution, ACIPI has become a source of inspiration, connecting people to the pursuit of discoveries and innovations that impact our daily lives. However, the scientific world can often seem distant or intimidating, especially for young people who may not see the surrounding opportunities. To change this reality, ACIPI partnered with the Laboratory of Intelligent Robotics, Automation, and Systems (LABIRAS) to create a web system. Its goal was clear: to make the academy's information more accessible and appealing, particularly to younger audiences. The result was an interactive platform, designed with an attractive and fully functional interface tailored to engage the youth. Its development involved the active participation of ACIPI members, culminating in a comprehensive platform for managing content and promoting the academy's activities.

Keywords: piauí academy of sciences; web system; young audience.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama UML de Classes Resumido	23
Figura 2 – Diagrama de Casos de Uso - Sistema Administrativo	25
Figura 3 – Diagrama de Casos de Uso - Sistema Visitante	26
Figura 4 – Diagrama da Arquitetura	28
Figura 5 – Página de início	29
Figura 6 – Página A Academia	31
Figura 7 – Página de Membros da Diretoria	32
Figura 8 – Página de Detalhe dos Membros da Diretoria	33
Figura 9 – Página de Membros	34
Figura 10 – Página de Detalhe dos Membros	35
Figura 11 – Página de Notícias	37
Figura 12 – Página de Notícias - Variação sem notícia	37
Figura 13 – Página de Detalhes da Notícia	38
Figura 14 – Página das Revistas	39
Figura 15 – Página dos Contatos	40
Figura 16 – Página de Login	40
Figura 17 – Página de Não Encontrado	41
Figura 18 – Painel de Membros	42
Figura 19 – Painel de Notícias	42
Figura 20 – Painel de Revistas	43
Figura 21 – Painel de Membros da Diretoria	43
Figura 22 – Página de Adicionar/Edição de Membro	44
Figura 23 – Página de Adicionar/Editar Diretoria	45
Figura 24 – Página de Adicionar/Editar Revistas	46
Figura 25 – Interface - Página de Adicionar/Editar Notícias	47

LISTA DE ABREVIATURAS E SIGLAS

ACID	Atomicidade, Consistência, Isolamento, Durabilidade
ACIPI	Academia de Ciências do Piauí
API	<i>Application Programming Interface</i>
AWS	<i>Amazon Web Services</i>
CRUD	<i>Create, Read, Update e Delete</i>
DOM	<i>Document Object Model</i>
JS	<i>JavaScript</i>
JSON	<i>JavaScript Object Notation</i>
HTML	<i>HyperText Markup Language</i>
LABIRAS	<i>Laboratory of Intelligent Robotics, Automation and Systems</i>
S3	<i>Simple Storage Service</i>
REST	<i>Representational State Transfer</i>
UML	<i>Unified Modeling Language</i>
W3C	<i>World Wide Web Consortium</i>

SUMÁRIO

1	INTRODUÇÃO	13
1.1	OBJETIVOS	14
1.1.1	Objetivo geral	14
1.1.2	Objetivos específicos	14
2	TECNOLOGIAS ENVOLVIDAS	15
2.1	TECNOLOGIAS DE <i>FRONTEND</i>	16
2.1.1	React JS	16
2.1.2	Formspree	16
2.1.3	Mapbox	17
2.2	TECNOLOGIAS DE <i>BACKEND</i>	17
2.2.1	Node JS	17
2.2.2	TypeScript	17
2.2.3	Express JS	17
2.2.4	PostgreSQL	18
2.2.5	Amazon S3	18
2.2.6	Redis	19
2.3	TECNOLOGIAS DE <i>DEPLOY</i>	19
2.3.1	Render	19
2.4	TECNOLOGIAS DE <i>PROTOTIPAÇÃO</i>	19
2.4.1	Figma	19
3	MODELAGEM DO PROJETO	20
3.1	LEVANTAMENTO DE REQUISITOS	21
3.1.1	Requisitos funcionais	21
3.1.2	Requisitos não funcionais	22
3.2	DIAGRAMA DE CLASSES	22
3.3	DIAGRAMA DE CASO DE USOS	23
3.4	ARQUITETURA DO SISTEMA	26
3.5	INTERFACES	28
4	SOFTWARE	48
4.1	IMPLANTAÇÃO	48
5	CONSIDERAÇÕES FINAIS	49

REFERÊNCIAS	50
ANEXO A – DECLARAÇÃO DE ENTREGA DO CÓDIGO FONTE DO SOFTWARE A COORDENAÇÃO DO CURSO . .	53

1 INTRODUÇÃO

O surgimento da ciência e do método científico é um marco fundamental na história do pensamento humano. Embora a curiosidade e a observação da natureza tenham existido desde tempos antigos, foi somente com o surgimento das civilizações que essas observações começaram a ser registradas e analisadas.

Esses antigos povos, que habitavam a Mesopotâmia, Egito, Grécia e China deram contribuições significativas para o desenvolvimento de conceitos científicos rudimentares, como astronomia, matemática e medicina (MARTINS; BUFFON; NEVES, 2019).

No entanto, foi durante o Renascimento europeu que o método científico começou a ser formalizado. Figuras como Galileu Galilei, com suas observações telescópicas revolucionárias, e Francis Bacon, com suas ideias sobre o empirismo e a indução, ajudaram a estabelecer os fundamentos do método científico moderno (CHIBENI, 2004). O método científico baseia-se na formulação de hipóteses testáveis, na realização de experimentos controlados e na análise sistemática dos resultados. Ele proporciona uma estrutura sólida para a investigação e a descoberta, permitindo que a ciência avance de maneira organizada e progressiva.

No Brasil, as primeiras tentativas de organização científica remontam ao século XVIII, com o surgimento da Academia Científica do Rio de Janeiro, criada em 1772 pelo marquês do Lavradio. Contudo, sua breve existência de apenas sete anos foi sucedida pela Sociedade Literária do Rio de Janeiro, encerrada por motivos políticos. No século XX, a multiplicação das sociedades científicas refletiu o crescimento da ciência no Brasil, impulsionado pela disseminação das universidades e o estabelecimento da pós-graduação. Com o advento desses acontecimentos, destaca-se a fundação da Sociedade Brasileira de Ciência em 1916, posteriormente renomeada Academia Brasileira de Ciências (ABC), e o estabelecimento das primeiras faculdades de filosofia, ciências e letras nas décadas de 1930, fundamentais para a formação de profissionais e pesquisadores em diversas áreas científicas (BUENO, 2022).

Diante desse cenário, a criação da Academia de Ciências do Piauí (ACIPI), em 2002, torna-se um marco histórico no panorama científico da região. Esse evento não apenas assinala um avanço significativo na promoção e desenvolvimento da pesquisa acadêmica local, mas também carrega consigo a herança da longa trajetória das instituições científicas no Brasil. Como um reflexo dessa história, a academia surge como um propagador de conhecimento, inspirando a comunidade a se engajar na busca por descobertas e inovações.

Mesmo com a importância de seu surgimento, a instituição científica enfrenta uma certa dificuldade para atrair novos membros, tornando-se um desafio persistente no

cenário acadêmico atual. Apesar do imenso potencial de crescimento e aprendizado que essas instituições oferecem, muitos jovens podem se sentir intimidados pela complexidade do ambiente científico ou desconhecem as oportunidades disponíveis. Além disso, a competição com outras áreas de interesse e a falta de visibilidade podem dificultar ainda mais a atração de novos membros. Nesse contexto, é fundamental que as instituições científicas adotem estratégias eficazes de comunicação e engajamento (SANTOS *et al.*, 2023).

Diante disso, foi proposto pela ACIPI ao *Laboratory of Intelligent Robotics, Automation and Systems* (LABIRAS), a construção de um *website* para o gerenciamento de informações da academia, assim, visando oferecer uma plataforma acessível para compartilhar conhecimento, oportunidades de pesquisa e eventos científicos. Além disso, essa aplicação também tem a responsabilidade de ser um meio de divulgação digital eficaz das atividades da ACIPI, deste modo, ampliando o alcance da academia para públicos de diferentes idades, garantindo que seu impacto e suas atividades sejam conhecidos e acessíveis a um público mais amplo.

1.1 OBJETIVOS

1.1.1 Objetivo geral

Desenvolver um sistema *web* para a divulgação e gestão de informações da ACIPI.

1.1.2 Objetivos específicos

- Levantar os requisitos funcionais e não funcionais do sistema, incluindo as necessidades específicas da ACIPI em termos de gestão de informações, como cadastro de membros, gerenciamento desses membros, publicações acadêmicas e notícias.
- Projetar a arquitetura do sistema *web*, definindo as tecnologias a serem utilizadas, a estrutura de banco de dados, a interface do usuário e os principais módulos e funcionalidades a serem implementados.
- Desenvolver o sistema *web* de acordo com as especificações definidas, seguindo as melhores práticas de desenvolvimento de software e garantindo a qualidade, segurança e usabilidade do produto final.
- Oferecer treinamento e capacitação aos membros da ACIPI para garantir o uso eficaz do sistema *web*, fornecendo orientações sobre como acessar e utilizar as diferentes funcionalidades disponíveis.

2 TECNOLOGIAS ENVOLVIDAS

Diante do cenário exposto acima, foram analisados diversos tipos de ferramentas para a construção desse sistema. Contudo, foi escolhida a utilização da linguagem de programação *JavaScript*. Sua escolha se deu por sua popularidade e ampla utilização no desenvolvimento web. Essa linguagem de programação tornou-se uma peça importante tanto para o desenvolvimento da interface gráfica, o meio pelo qual o usuário interage com o sistema, *frontend*, quanto para a construção da parte lógica do sistema, que detém as regras de negócio da instituição, *backend*. Sua popularidade garante acesso a um vasto acervo de conteúdos didáticos e suporte da comunidade, facilitando a consulta de documentações, bibliotecas, *frameworks*, e diversos outros tipos de ferramentas que visam a facilitação da construção de sistemas, e fóruns, portanto, acelerando o processo de desenvolvimento.

A flexibilidade que o *JavaScript* possui, permite o desenvolvimento de sistemas conhecidos como *fullstacks*, abrange tanto o *backend* quanto o *frontend*, reutilizando a mesma linguagem em todas as camadas. Sobre essa perspectiva, a ferramenta simplifica a construção e manutenção de *websites*, promovendo uma maior consistência no código. Além disso, a utilização de uma única linguagem para o desenvolvimento *fullstack* melhora a curva de aprendizado, pois quem está desenvolvendo pode focar apenas nas nuances do desenvolvimento de cada componente do sistema. Sendo assim, é possível agilizar o processo de construção e integração de novas funcionalidades e, além disso, torna a resolução de problemas mais simplificadas (SWAN, 2025).

A escolha de separar o desenvolvimento em dois módulos, *backend* e *frontend*, foi motivada pela necessidade de uma arquitetura mais flexível e escalável. Ao hospedar esses módulos separadamente e permitir que se comuniquem via HTTP, um protocolo de comunicação na *web*, adota-se uma abordagem desacoplada, permitindo que cada módulo escale de forma independente, algo essencial para manter a eficiência do sistema conforme o seu crescimento. Além disso, essa separação facilita a manutenção, pois as atualizações ou correções em um dos módulos podem ser realizadas sem impactar diretamente o outro, tornando o processo de desenvolvimento e suporte mais ágil e organizado (FAUSTINO *et al.*, 2024).

Partindo dessas escolhas, foi possível selecionar as demais ferramentas a serem utilizadas durante o processo de desenvolvimento do sistema. Logo em seguida, serão explorados os principais *frameworks* e bibliotecas empregados, exibindo suas funcionalidades e o papel de cada um na construção do sistema.

2.1 TECNOLOGIAS DE *FRONTEND*

2.1.1 React JS

O *React.js* foi desenvolvido pelo *Facebook* e lançado em 2013, sua construção surgiu da necessidade de criar interfaces de usuário mais eficientes e com um desempenho melhor. A ideia por trás do *React* foi introduzir um modelo de programação declarativo e baseado em componentes, que permite a criação de interfaces de usuário interativas e dinâmicas. Ao invés de ter que atualizar a interface de forma imperativa, o *React* introduz o conceito de Virtual DOM, que otimiza as atualizações de interface ao minimizar as operações de manipulação de DOM reais, resultando em um desempenho significativamente melhor (ROY, 2021). Esse enfoque inovador permitiu que desenvolvedores construíssem aplicações web mais rápidas e responsivas.

A sua popularidade, deve-se a vários fatores. Primeiramente, a sua curva de aprendizado relativamente baixa, combinada com uma poderosa capacidade de abstração, torna-o acessível tanto para iniciantes quanto para desenvolvedores experientes (AGGARWAL *et al.*, 2018). Além disso, a vasta comunidade de desenvolvedores contribui continuamente para a criação de bibliotecas, ferramentas e extensões, enriquecendo, assim, o seu ecossistema (JINDAN *et al.*, 2023). Grandes empresas como *Facebook*, *Instagram* e *Airbnb* utilizam o *React* em suas aplicações, o que solidifica sua reputação como uma ferramenta confiável e escalável para desenvolvimento de interfaces modernas (NAIM, 2017). A modularidade e a capacidade de reuso de componentes também permitem um desenvolvimento mais organizado e eficiente, resultando em uma adoção ampla e crescente no mercado de desenvolvimento web.

2.1.2 Formspree

Com o decorrer do projeto, observou-se a necessidade de criar uma solução para o envio de formulários de contato. Para isso, optou-se pelo uso do *Formspree*, uma ferramenta que simplifica a integração de formulários de *websites* com um serviço de envio de *e-mails*, eliminando a necessidade de configurar toda a infraestrutura no *backend*. O objetivo é capturar as informações inseridas pelos usuários nos formulários do *frontend* e enviá-las diretamente para os servidores do *Formspree*, que, por sua vez, as encaminham ao endereço de *e-mail* pré-cadastrado, já formatadas para facilitar a compreensão do destinatário. Isso é especialmente útil para projetos que precisam de uma solução rápida e eficiente para lidar com formulários, sem a complexidade de gerenciar um servidor próprio. Além disso, o *Formspree* oferece recursos como proteção contra *spam* e integração com outros serviços, tornando-o uma escolha versátil e segura para adicionar formulários em *sites* e aplicações web (FORMSPREE, 2024).

2.1.3 Mapbox

Mapbox é uma plataforma de mapeamento que permite aos desenvolvedores renderizar mapas personalizados e interativos para *websites* e aplicativos móveis. Com a sua tecnologia, ela oferece mapas de alto desempenho e uma experiência de usuário fluida. Sua interface é altamente flexível, permitindo que os mapas gerados sejam estilizados conforme as necessidades do projeto. Assim, o *Mapbox* torna-se um aliado imprescindível para a criação de mapas navegáveis em aplicações *web*, permitindo que os usuários visualizem facilmente a localização desejada (RZESZEWSKI, 2023).

2.2 TECNOLOGIAS DE *BACKEND*

2.2.1 Node JS

Node JS é uma plataforma de desenvolvimento criada por Ryan Dahl em 2009, que permite executar *JavaScript* no lado do servidor, e não apenas no navegador. Seu surgimento tornou-se uma grande inovação, pois possibilitou a criação de aplicações *web* mais rápidas e eficientes. Ele usa um modelo de execução assíncrono e orientado a eventos, o que significa que pode lidar com muitas tarefas ao mesmo tempo, sem travar, tornando-o muito mais rápido e escalável que outras plataformas. Atualmente, *Node JS* é essencial para o desenvolvimento *web*. Ele é perfeito para aplicações em tempo real, como *chats* e jogos online, devido à sua capacidade de lidar com muitas operações de entrada e saída de maneira eficiente. Além disso, o *npm*, ferramenta que auxilia na gestão de pacotes, oferece uma vasta biblioteca de módulos que facilitam o desenvolvimento de aplicações complexas (JADHAV; GONSALVES, 2020).

2.2.2 TypeScript

O *TypeScript* é um superconjunto tipado desenvolvido pela *Microsoft* que adiciona tipagem estática opcional ao *JavaScript* padrão. Isso permite aos desenvolvedores declarar tipos para variáveis, parâmetros de função e retornos de função, ajudando a detectar erros comuns durante o desenvolvimento. Além de proporcionar maior segurança e robustez ao código, o uso dessa ferramenta também melhora a experiência de desenvolvimento com recursos avançados como inferência de tipos, interfaces, *enums* e suporte a classes. Sua adoção é ampla em diversos tipos de projetos, pois permite escrever código mais legível, escalável e fácil de manter, resultando em um código claro e conciso (LEMPIÖ, 2023).

2.2.3 Express JS

O *Express JS*, criado por TJ Holowaychuk em 2010, é um framework minimalista e flexível que facilita o desenvolvimento de aplicações *web* e APIs no ecossistema

Node JS. Oferecendo ferramentas como roteamento robusto e integração simplificada, ele atua como *middleware*, ou seja, entre requisições e respostas do servidor. Aproveitando a arquitetura assíncrona do *Node JS*, o *Express* permite criar servidores eficientes e escaláveis, tornando-se uma das principais escolhas para desenvolvimento *backend* devido à sua simplicidade e extensibilidade (BAFNA *et al.*, 2022).

2.2.4 PostgreSQL

PostgreSQL é um sistema de gerenciamento de banco de dados relacional de código aberto, inicialmente desenvolvido no Departamento de Ciência da Computação da Universidade da Califórnia, Berkeley, sob a liderança do professor Michael Stonebraker. Originado em 1985 como uma evolução do banco de dados *Ingres*, o projeto buscava adicionar suporte a tipos de dados complexos e criar um mecanismo de regras para automação de tarefas. Em 1996, foi renomeado para *PostgreSQL* para destacar seu suporte à linguagem *SQL* (*Structured Query Language*), marcando o início de sua contínua evolução por meio de contribuições de uma comunidade global de desenvolvedores (ASANKA, 2020).

Atualmente, *PostgreSQL* é amplamente reconhecido por sua robustez, flexibilidade e aderência aos padrões *SQL*, sendo uma escolha comum em aplicações empresariais e *web*. Suas funcionalidades avançadas incluem suporte a transações ACID (Atomicidade, Consistência, Isolamento, Durabilidade), replicação de dados, extensibilidade e um sistema poderoso de tipos de dados, características que o tornam uma solução confiável e adaptável. Além disso, sua capacidade de escalar e atender a diversas necessidades consolida sua relevância no cenário tecnológico atual (EnterpriseDB Team, 2024; Hossted Team, 2024).

2.2.5 Amazon S3

O *Amazon S3* (*Simple Storage Service*) foi lançado pela *Amazon Web Services* (AWS) em 2006 como um serviço de armazenamento de objetos altamente escalável e durável (Amazon Web Services, 2006). Desde o seu surgimento, o *S3* revolucionou a maneira como os dados são armazenados e acessados na nuvem, oferecendo uma solução segura, confiável e econômica para armazenar uma quantidade praticamente ilimitada de dados. Empresas e desenvolvedores utilizam esse serviço para uma ampla gama de aplicações, desde *backups* e arquivamento até hospedagem de conteúdo estático e *big data analytics*. Com recursos avançados como versionamento de objetos, políticas de ciclo de vida e integração com outras ferramentas da AWS, o *S3* tornou-se um componente crucial na infraestrutura de muitas organizações (BHOSALE; DAUND; KALAMBE, 2024).

2.2.6 Redis

O *Redis* é um sistema de armazenamento de dados em memória, de código aberto, desenvolvido por Salvatore Sanfilippo em 2009. Inicialmente criado para melhorar a escalabilidade e o desempenho do banco de dados de sua empresa, o *Redis* rapidamente ganhou popularidade devido à sua capacidade de armazenar e manipular dados de maneira extremamente rápida. Ele suporta diversos tipos de estruturas de dados, o que o torna extremamente versátil. *Redis* é amplamente utilizado em aplicações modernas para *caching*, gerenciamento de sessões, filas de mensagens e análises em tempo real, graças à sua velocidade e eficiência (SILVA, 2025).

2.3 TECNOLOGIAS DE *DEPLOY*

2.3.1 Render

O *Render* é uma plataforma de hospedagem de aplicações e serviços *web* que surgiu em 2019, fundada por Anurag Goel. Com a missão de simplificar o processo de implantação e gerenciamento de aplicações, *Render* oferece uma experiência unificada para desenvolvedores ao automatizar tarefas complexas como configuração de servidores, escalabilidade e segurança. Desde o seu lançamento, *Render* ganhou destaque por sua facilidade de uso e pela integração contínua com ferramentas de desenvolvimento modernas. A plataforma suporta uma variedade de linguagens e *frameworks*, tornando-se uma escolha popular para empresas em crescimento e empresas que buscam uma solução eficiente e escalável para hospedar suas aplicações. Atualmente, o *Render* é valorizado por permitir que desenvolvedores se concentrem mais no desenvolvimento de software e menos na infraestrutura, acelerando o tempo de entrega e melhorando a confiabilidade dos serviços (GOEL, 2019).

2.4 TECNOLOGIAS DE *PROTOTIPAÇÃO*

2.4.1 Figma

O *Figma* foi lançado em 2016 por Dylan Field e Evan Wallace, visando criar uma ferramenta de design colaborativo baseada na web. A ideia era simplificar o processo de design de interfaces, permitindo que designers e desenvolvedores trabalhassem juntos em tempo real, independentemente de sua localização (RIBEIRO *et al.*, 2024). Ela facilita a criação de protótipos interativos, oferecendo uma plataforma que une ferramentas de design, prototipagem e colaboração em um só lugar. Isso permite uma maior eficiência no desenvolvimento de interfaces, desde a concepção até a implementação, tornando o processo mais ágil e integrado (PRESTES, 2021).

3 MODELAGEM DO PROJETO

A modelagem de software desempenha um papel crucial no desenvolvimento de sistemas, pois fornece uma representação abstrata dos diferentes componentes e processos envolvidos em um projeto de software. Essa prática é essencial para garantir que todos os envolvidos no processo, desde desenvolvedores até *stakeholders*, compreendam de forma clara a estrutura e o comportamento do sistema. Para alcançar esse entendimento amplo e unificado, utiliza-se uma variedade de diagramas e linguagens de modelagem, como a *UML (Unified Modeling Language)*, que permitem visualizar e comunicar as funcionalidades do sistema de maneira precisa (MENEGASSI; RUEDA, 2024).

Essas representações não apenas facilitam a comunicação entre os membros da equipe de desenvolvimento, mas também são fundamentais para que donos de produto e clientes possam ter uma visão clara do produto final e das etapas necessárias para sua implementação. A modelagem assegura que todos compartilhem uma visão comum do sistema, o que é vital para o sucesso do projeto e para a manutenção da qualidade e eficiência ao longo do ciclo de vida do software (SILVA; LENCASTRE; CASTRO, 2024).

Além de promover a comunicação eficaz, a modelagem de *software* desempenha um papel crucial na identificação e mitigação de riscos durante as fases iniciais do desenvolvimento. Ao criar modelos detalhados, é possível detectar inconsistências, lacunas e possíveis problemas de integração antes que o código seja escrito, economizando tempo e recursos. A modelagem também permite a análise e otimização de diferentes aspectos do sistema, como desempenho, escalabilidade e segurança, contribuindo para a construção de soluções mais robustas e eficientes.

Nos próximos capítulos, serão abordadas as etapas e os artefatos envolvidos no processo de modelagem de software. Inicialmente, será discutido o levantamento de requisitos, onde são identificadas e documentadas as necessidades e expectativas dos membros da Academia de Ciências do Piauí.

Em seguida, será apresentado o diagrama de classes, que fornece uma visão clara das funcionalidades esperadas, por meio das classes do sistema, explorando como sua estrutura está montada e revelando seus relacionamentos. Logo após, será a vez do diagrama de casos de uso, responsável por ilustrar as interações entre os atores, os responsáveis por interagir com o sistema, com os casos de uso, as funcionalidades do sistema. Na seção sobre arquitetura do sistema, será detalhado o design geral e a organização dos componentes do software.

Por fim, discutiremos a interface, focando na interação do usuário com o sistema e na usabilidade. Cada um desses capítulos apresenta uma visão dos componentes

essenciais da modelagem de software, contribuindo para um entendimento do processo de desenvolvimento do produto.

3.1 LEVANTAMENTO DE REQUISITOS

Os requisitos funcionais e não funcionais levantados para a construção do sistema foram:

3.1.1 Requisitos funcionais

- **[RF001]** O sistema deve permitir que os usuários encontrem e acessem as informações sobre a ACIPI.
- **[RF002]** O sistema deve permitir que os usuários encontrem e acessem todos os membros da diretoria.
- **[RF003]** O sistema deve permitir que os usuários possam filtrar os membros da diretoria por um intervalo de tempo.
- **[RF004]** O sistema deve permitir que os usuários encontrem e acessem todos os membros titulares da academia.
- **[RF005]** O sistema deve permitir que os usuários encontrem e acessem todas as notícias publicadas.
- **[RF006]** O sistema deve permitir aos usuários encontrar e acessar todas as edições de revistas publicadas digitalmente pela ACIPI.
- **[RF007]** O sistema deve permitir o envio de mensagem para o e-mail da ACIPI.
- **[RF008]** O sistema deve permitir a visualização e interação com a localização da ACIPI.
- **[RF009]** O sistema deve permitir que os administradores possam acessar a área administrativa.
- **[RF010]** O sistema deve permitir que os administradores criem, editem e excluam os membros titulares da academia.
- **[RF011]** O sistema deve permitir que os administradores busquem os membros pelo nome ou cadeira que ele ocupa.
- **[RF012]** O sistema deve permitir que os administradores criem, editem e excluam os membros da diretoria.

- **[RF013]** O sistema deve permitir que os administradores criem, editem e excluam notícias.
- **[RF014]** O sistema deve permitir que os administradores busquem as notícias pelo título.
- **[RF015]** O sistema deve permitir que os administradores criem, editem e excluam edições de revistas publicadas digitalmente pela ACIPI.
- **[RF016]** O sistema deve permitir que os administradores busquem as edições de revistas publicadas pelo nome.

3.1.2 Requisitos não funcionais

- **[RNF001]** O site deve implementar medidas de segurança adequadas para proteger os formulários.
- **[RNF002]** O site deve implementar HTTPS.
- **[RNF003]** O site deve garantir que o design seja responsivo e acessível em dispositivos móveis e tablets.
- **[RNF004]** O código do site deve ser bem documentado e seguir boas práticas de desenvolvimento.
- **[RNF005]** Estruturar o site para poder ser facilmente escalado conforme o aumento no número de usuários e conteúdo.
- **[RNF006]** O site deve carregar todas as páginas em menos de 30 segundos em uma conexão estável com pelo menos 5 Mbps de download.
- **[RNF007]** Garantir compatibilidade com os principais navegadores (Chrome, Firefox, Safari, Edge).

3.2 DIAGRAMA DE CLASSES

O diagrama de classes *UML* resumido apresentado na Figura 1 oferece uma visão clara e estruturada das relações entre as principais entidades do sistema, seguindo os padrões pre-estabelecidos pela Academia de Ciências do Piauí (ACIPI). Desenvolvido com base nos princípios da programação orientada a objetos, o diagrama organiza de forma lógica os diferentes tipos de elementos que compõe a regra de negócio da ACIPI, cada um com suas respectivas funções e características.

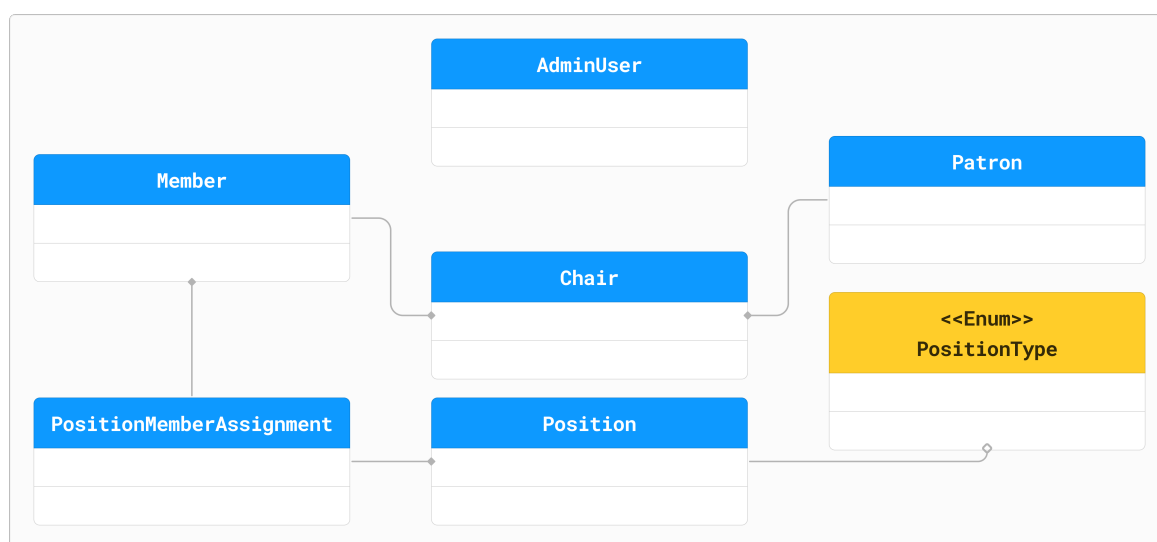
A classe *AdminUser* representa o administrador do sistema, que possui privilégios elevados e é responsável pelo gerenciamento das informações relativas aos membros, cadeiras, patronos e membros do conselho. A classe *Member*, por sua vez,

corresponde aos membros da ACIPI, que podem ocupar uma *Chair* (cadeira). Cada *Chair* está vinculada a um Patrono, um membro homenageado que é historicamente associado àquela cadeira, assegurando a preservação da memória e dos valores culturais da instituição.

O diagrama também captura a estrutura organizacional do conselho da ACIPI por meio da classe *PositionMemberAssignment*, que representa os membros que ocupam cargos específicos dentro do conselho. Cada membro do conselho está associado a uma classe *Position*, que define o cargo ocupado por esse membro na academia, como presidente, secretário, entre outros. Essa organização permite uma visualização clara e eficiente das funções e responsabilidades dentro da ACIPI.

Portanto, as inter-relações entre as classes e seus respectivos atributos refletem a hierarquia e as responsabilidades no interior da ACIPI. O sistema proposto procura assegurar que cada entidade e cargo sejam representados e administrados de maneira eficaz, assim, facilitando a organização das atividades e a preservação do patrimônio cultural da associação.

Figura 1 – Diagrama UML de Classes Resumido



Fonte: Elaborado pelo autor.

3.3 DIAGRAMA DE CASO DE USOS

Em uma reunião com o diretor da ACIPI e membros da organização, foram discutidas as necessidades e expectativas para o desenvolvimento do sistema. Durante esse processo, foram identificados dois sistemas principais: o Sistema Visitante e o Sistema Administrativo. Estes sistemas foram idealizados para atender às demandas

da organização, permitindo uma gestão do conteúdo do site e na interação com o público.

Os sistemas identificados a partir do levantamento de requisitos abrangem, em termos de casos de uso:

- Sistema 01 - Sistema Visitante: Focado na interação com os usuários visitantes, permitindo que eles acessem informações relevantes e acompanhem as atividades da ACIPI;
- Sistema 02 - Sistema Administrativo: Projetado para que o administrador gerencie de forma eficaz as operações do site da ACIPI, incluindo a gestão de membros, publicações, e a atribuição de funções para visualização do público.

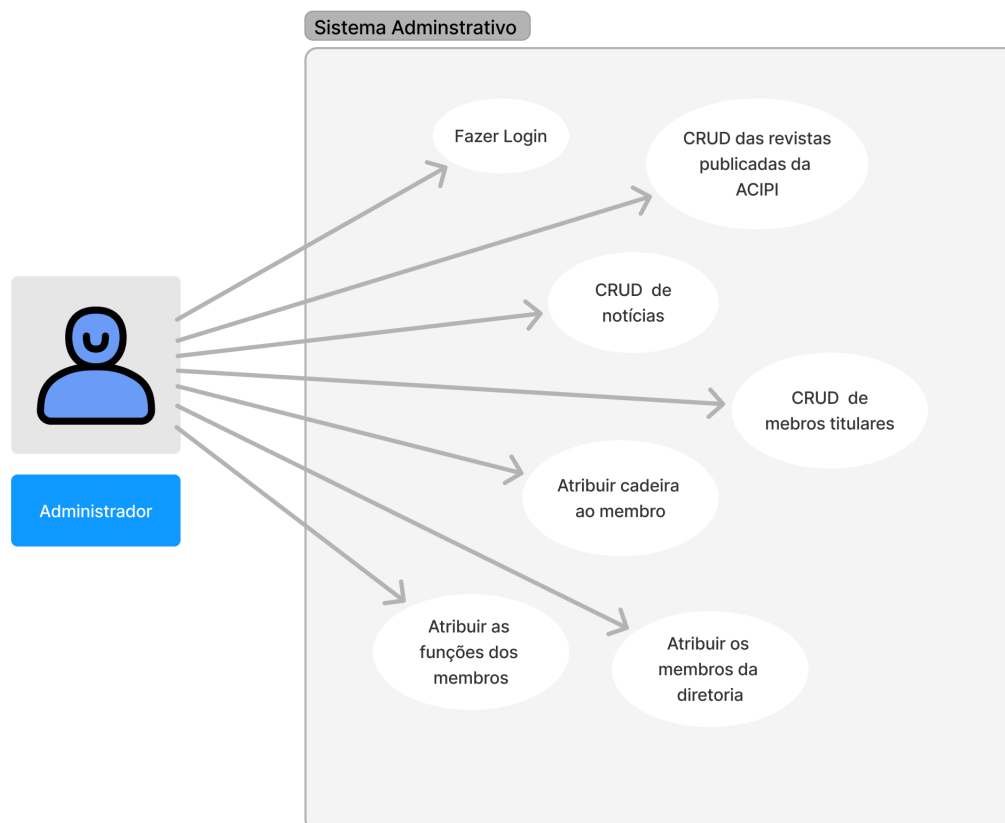
Com base nesses requisitos, foram desenvolvidos os diagramas de casos de uso que detalham as interações entre os atores e os sistemas, conforme descrito nas seções subsequentes. Estes diagramas oferecem uma visão clara das funcionalidades e responsabilidades de cada sistema, destacando a importância de um design estruturado e alinhado com as necessidades da ACIPI.

O diagrama de casos de uso apresentado abaixo, figura 2, ilustra as principais funcionalidades do sistema administrativo da ACIPI, com foco nas operações realizadas pelo Administrador. Este ator desempenha um papel central no sistema, sendo responsável por gerenciar tanto o conteúdo quanto os membros da organização. O diagrama mostra que o Administrador tem acesso a diversas funcionalidades, cada uma representada como um caso de uso específico.

Entre as funcionalidades mais importantes, estão as operações de *CRUD* (Create, Read, Update, Delete), que podem ser traduzidas como criar, ler, atualizar e deletar, para a gestão de revistas publicadas, notícias e membros titulares da ACIPI. Essas operações são essenciais para a manutenção e atualização das informações no sistema, permitindo que o conteúdo seja gerenciado de forma eficaz e precisa. Além disso, o Administrador pode atribuir cadeiras aos membros, indicando a capacidade de designar posições dentro da organização, o que reflete a estrutura hierárquica e funcional da ACIPI.

Outro ponto importante destacado no diagrama é a capacidade do Administrador de atribuir funções específicas aos membros e definir quem fará parte da diretoria. Isso enfatiza o papel estratégico do Administrador na gestão da visualização correta do papel de cada membro da ACIPI.

Figura 2 – Diagrama de Casos de Uso - Sistema Administrativo



Fonte: Elaborado pelo autor.

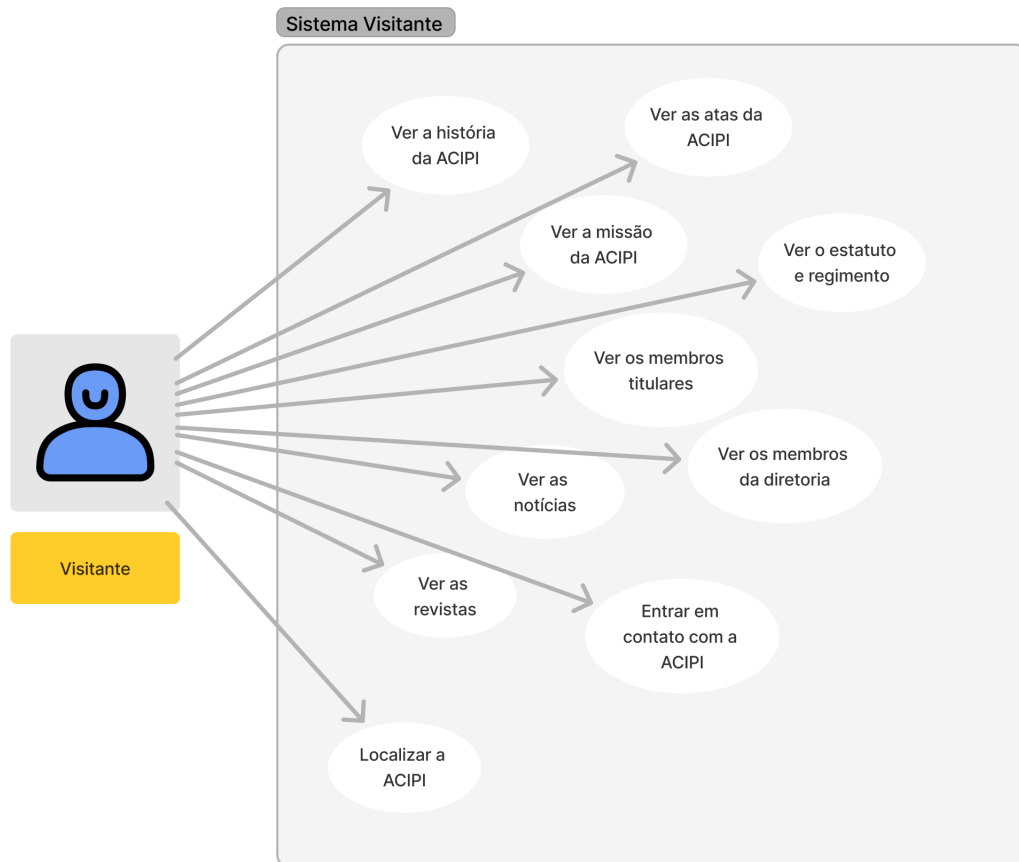
O diagrama de casos de uso do Sistema Visitante, que pode ser visualizado na figura 3, descreve as principais funcionalidades que estão disponíveis para o público geral, que pode acessar o sistema sem precisar fazer login. Nesse cenário, o “Visitante” torna-se qualquer usuário que interage com as opções de consulta e visualização oferecidas pela ACIPI. O sistema foi criado para facilitar o acesso a informações importantes sobre a instituição e suas atividades.

As funcionalidades destacadas no diagrama permitem que os visitantes conheçam a história, missão, estatuto e regimento da ACIPI. Além disso, eles podem visualizar atas, notícias e revistas publicadas, o que ajuda a manter o público atualizado sobre os eventos e novidades da organização. Esse acesso aberto é crucial para promover uma maior compreensão e envolvimento com a ACIPI, aumentando a transparência e a conexão com a instituição, reafirmando o que foi proposto desde o começo do projeto.

Outra vantagem importante do sistema é permitir que os visitantes localizem a ACIPI e entrem em contato com a organização. Essas funções facilitam a comunicação, oferecendo um canal direto para esclarecer dúvidas e obter mais informações. O diagrama demonstra o esforço da ACIPI em manter uma relação próxima e acessível com o público, assegurando que todas as informações relevantes estejam sempre

claras e disponíveis para todos.

Figura 3 – Diagrama de Casos de Uso - Sistema Visitante



Fonte: Elaborado pelo autor.

3.4 ARQUITETURA DO SISTEMA

A arquitetura do sistema, figura 4, foi baseada no modelo *API REST*, que possibilita a separação do *frontend* do *backend*, conceitos previamente explicados no capítulo 2. Essa separação permite que cada parte do sistema seja desenvolvida e mantida de forma independente, o que a torna mais flexível e escalável. No *frontend*, o foco principal é na exibição de informações e na interação com o usuário; para isso, foi utilizada a ferramenta *React JS*, sua hospedagem foi realizada na plataforma do *Render*. A escolha dessa plataforma foi baseada em uma pesquisa de preços, identificando a melhor opção dentro do orçamento da ACIPI no momento. No quesito *deploy*, a *Render* apresentou um custo mais vantajoso em comparação com a *AWS*, que também foi utilizada em outras partes do projeto. Por meio desta escolha, houve a criação de interfaces de usuário altamente responsivas e adaptáveis, assim, sendo visualizada tanto no *desktop*, computadores pessoais, quanto nos celulares, por meio da responsividade, que é a adaptação da interface ao dispositivo ao qual ela se encontra.

Optar por uma arquitetura *REST Web*, recomendada pela *W3C* (*World Wide Web Consortium*), reflete uma tentativa de utilizar as melhores práticas de desenvolvimento web (BOOTH *et al.*, 2004). Ao adotar esse modelo, o sistema beneficia-se de uma comunicação mais uniforme e de uma interface flexível, que não impõe limitações na escolha de tecnologias e frameworks pelos desenvolvedores. Essa flexibilidade é fundamental para criar aplicações modernas, que precisam se adaptar rapidamente às mudanças tecnológicas.

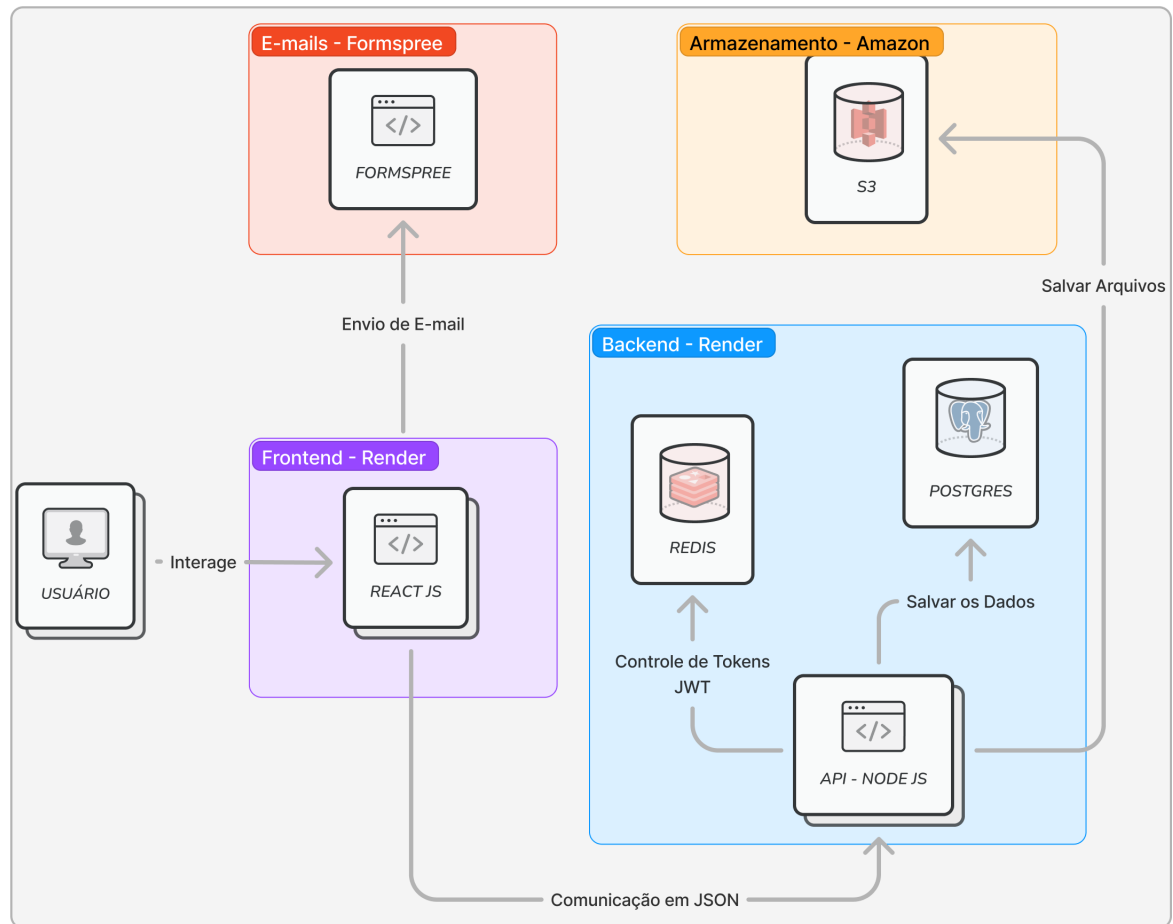
O *backend*, por sua vez, é responsável pelo processamento das requisições e pela lógica de negócio. Ele é implementado em *Node JS* com o auxílio do *micro-framework Express JS*, junto da linguagem de programação *Javascript*, e com a colaboração do *Typescript*, ferramenta que aumenta o poder da linguagem no momento do desenvolvimento de aplicações. O *backend* também foi hospedado na plataforma do *Render*, comunicando-se com o *frontend* por meio de *JSON* para troca de dados. Esse padrão de comunicação é eficiente e facilita a integração entre as diferentes partes do sistema, além de permitir a criação de *APIs* robustas. No segmento da autenticação do usuário, fez-se necessário a utilização da ferramenta *Redis* para gerenciar os tokens *JWT*, o que garante a segurança das sessões dos usuários e o gerenciamento eficaz da autenticação.

Para armazenamento de dados, a arquitetura utiliza o banco de dados *Postgres*, que é integrado ao *backend*. Quando uma aplicação *backend* é hospedada no *Render* o desenvolvedor tem a facilidade de poder integrar o *Postgres* e o *Redis* diretamente com a instância do *backend*, assim tendo maior confiabilidade e suporte por parte da plataforma, além de não expor elas ao acesso externo, somente ao *backend*.

O *Postgres* é conhecido por sua robustez, capacidade de lidar com grandes volumes de dados, o que o torna ideal para manter informações críticas do sistema. Além disso, o sistema usa o serviço *Amazon S3* para armazenar arquivos, oferecendo uma solução escalável e segura para guardar dados como imagens e documentos gerados e manipulados pela aplicação.

Outro ponto interessante dessa arquitetura é o uso do *Formspree* para envio de e-mails. Sua integração ao *frontend* é simplificada, tornando prático e rápido a implementação de formulários de contato e outras funcionalidades que exigem comunicação por e-mail, sem a necessidade de configurar o serviço no *backend*. Isso facilita o desenvolvimento e a manutenção do sistema, garantindo também a confiabilidade no envio de mensagens, portanto sempre que o usuário tentar entrar em contato com a Academia, terá seu contato entregue. Outro ponto a se mencionar, é que para a leitura da localização da ACIPI foi utilizada a ferramenta *Mapbox*, que exibe ao usuário que está visitando o site a posição exata da ACIPI.

Figura 4 – Diagrama da Arquitetura



Fonte: Elaborado pelo autor.

3.5 INTERFACES

As interfaces foram desenvolvidas com foco na usabilidade e na experiência do usuário, garantindo uma navegação intuitiva e visualmente agradável. Durante o processo de design, foram utilizados componentes reutilizáveis para manter a uniformidade estética e funcional em todas as telas. A seguir, serão apresentadas as principais interfaces criadas, detalhando suas funcionalidades.

Figura 5 – Página de início



Fonte: Elaborado pelo autor.

De início, serão exibidas as páginas destinadas ao usuário visitante do site, composta por página de início, a academia, diretoria, detalhe do membro da diretoria, membros, detalhe do membro, notícias, detalhe da notícia, revistas, contato e página não encontrada.

A página inicial da Academia de Ciências do Piauí, figura 5, é organizada para oferecer uma navegação clara e intuitiva. No topo e no rodapé, ambos presentes em outras páginas, há um barra de navegação com menu horizontal que permite acesso rápido às seções "A academia", "Diretoria", "Membros", "Notícias", "Revistas" e "Contatos". No rodapé, há também informações de navegação adicionais, contatos e links para redes sociais, assim, garantindo que os visitantes possam facilmente encontrar mais informações.

Nos elementos visuais restantes, um texto central destaca o nome da academia e logo abaixo tem-se um carrossel com algumas fotos de eventos da academia. Na seção de "Últimas Notícias" apresenta três cartões, cada um com uma imagem de destaque, título e descrição breve, permitindo ao usuário se manter atualizado com as novidades mais recentes. Em seguida, a seção "Sobre Nós" fornece uma introdução à ACIPI, acompanhada de um mapa do estado do Piauí remetendo diretamente ao estado, e logo abaixo tem-se um *link* convidativo para acessar a página de "A Academia", página que está representada na figura 6.

Na página "A Academia", um texto central destaca o nome da página, deixando claro ao usuário qual é a página em que ele está navegando. Esse elemento também é usado em outras seções, mas com o nome específico de cada uma. Além disso, nos textos dispostos nas seções de "História" e de "Estatuto e Regimento" há imagens ao lado dos textos, desde o início da construção de sua concepção, já se pensou em ter imagens que representassem bem a academia, como o logo, por exemplo. Outro ponto importante a se mencionar é que durante o desenvolvimento da *interface* houve também uma renovação da logo utilizada pela ACIPI, que pode ser encontrada em diversos componentes criados. O objetivo da construção dessa página foi esclarecer ao visitante a história, a missão e as regras atuais da ACIPI, tornando a página repleta de informações interessantes, novamente reforçando, o papel que esse sistema tem de divulgar a ACIPI ao público mais jovem.

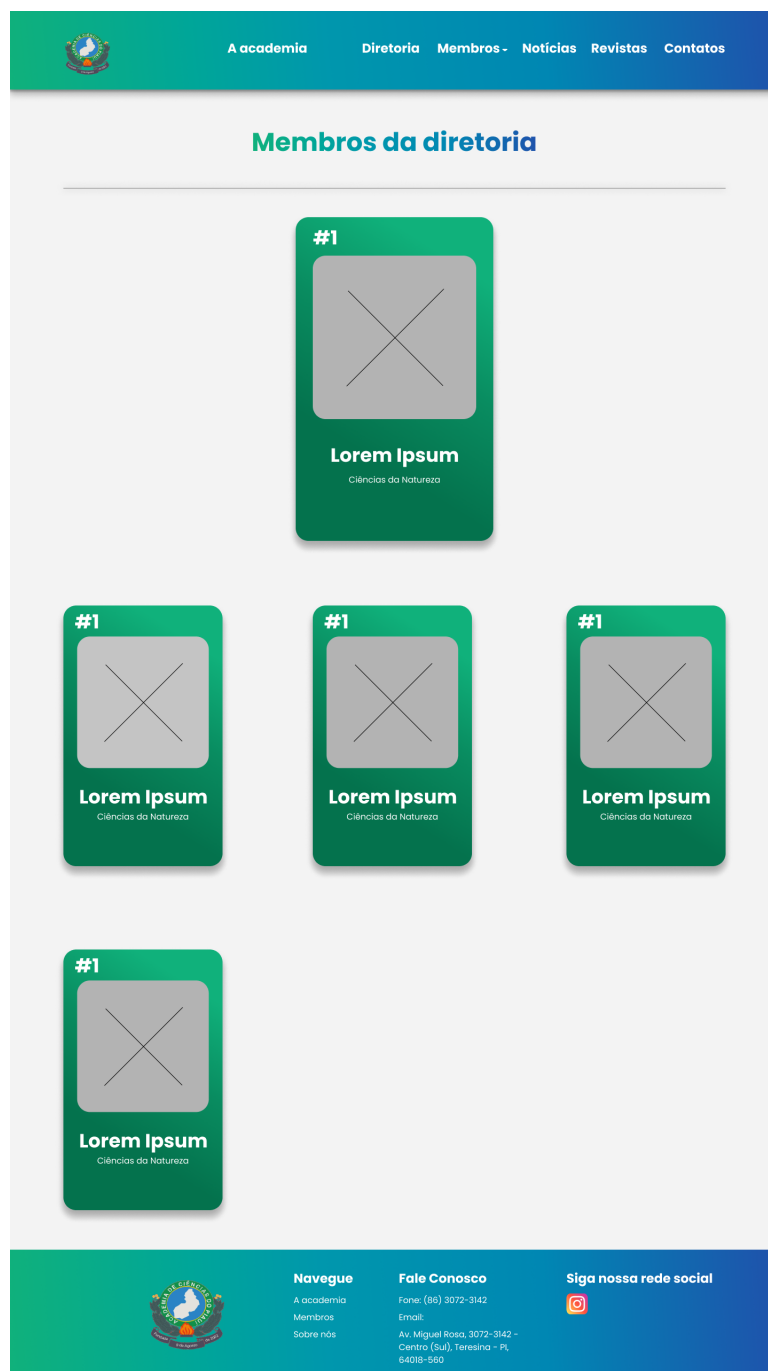
Na página "Membros da diretoria", que pode ser encontrada na figura 7, em sua parte central, encontram-se cartões, com objetivo de exibir os membros da diretoria, isso de forma hierárquica. Cada cartão contém uma imagem de destaque, nome do membro e qual é a sua área de atuação, atribui-se, adicionalmente, a característica de ser um componente clicável, encaminhando o usuário à página de detalhe daquele membro, denominada de "Página de Detalhe do Membro da Diretoria", que pode ser encontrada na figura 8.

Figura 6 – Página A Academia



Fonte: Elaborado pelo autor.

Figura 7 – Página de Membros da Diretoria



Fonte: Elaborado pelo autor.

Na página de "Detalhes do membro da diretoria", figura 8, além do texto central informativo sobre a página em que o visitante se encontra, também há outro texto central, mas com a finalidade de exibir o papel daquele membro dentro da Academia de Ciências do Piauí. O cartão que fica em destaque na esquerda ganha algumas funcionalidades que os demais cartões não tinham. Sua especificidade dá-se pela localização e inserção de *links* para as redes sociais do membro selecionado, logo abaixo da área de atuação, visando um acompanhamento do visitante dos trabalhos

acadêmicos que os membros da academia estão realizando.

Figura 8 – Página de Detalhe dos Membros da Diretoria

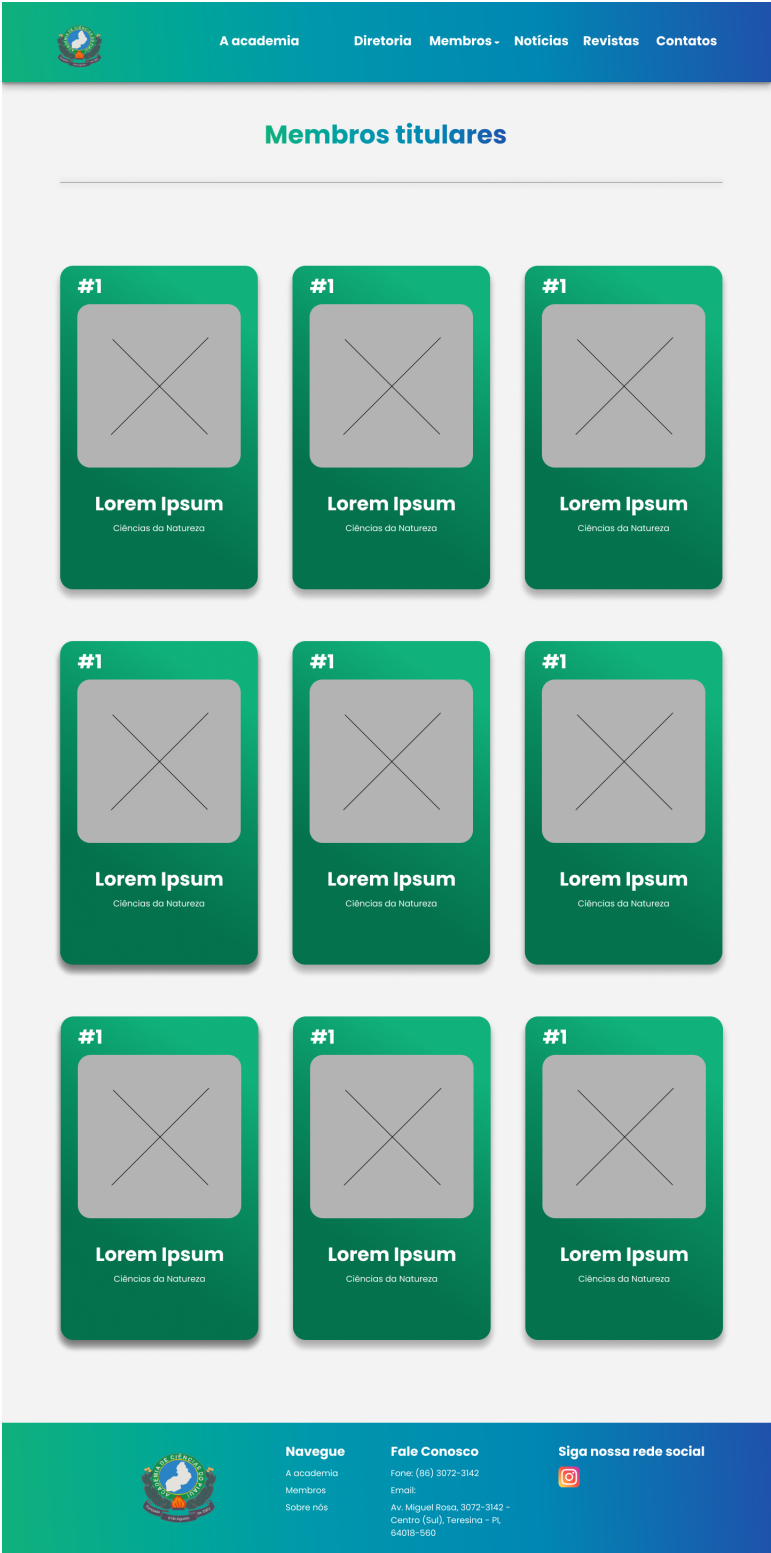


Fonte: Elaborado pelo autor.

Na página de "Membros", ilustrada na Figura 9, são apresentados todos os membros titulares da academia no centro da página. Cada membro é exibido em um cartão que mostra sua cadeira, nome, foto e área de atuação. Esses cartões são clicáveis, redirecionando o usuário para a página de "Detalhes do Membro" (Figura 10), onde são exibidas as informações detalhadas de cada membro titular da academia.

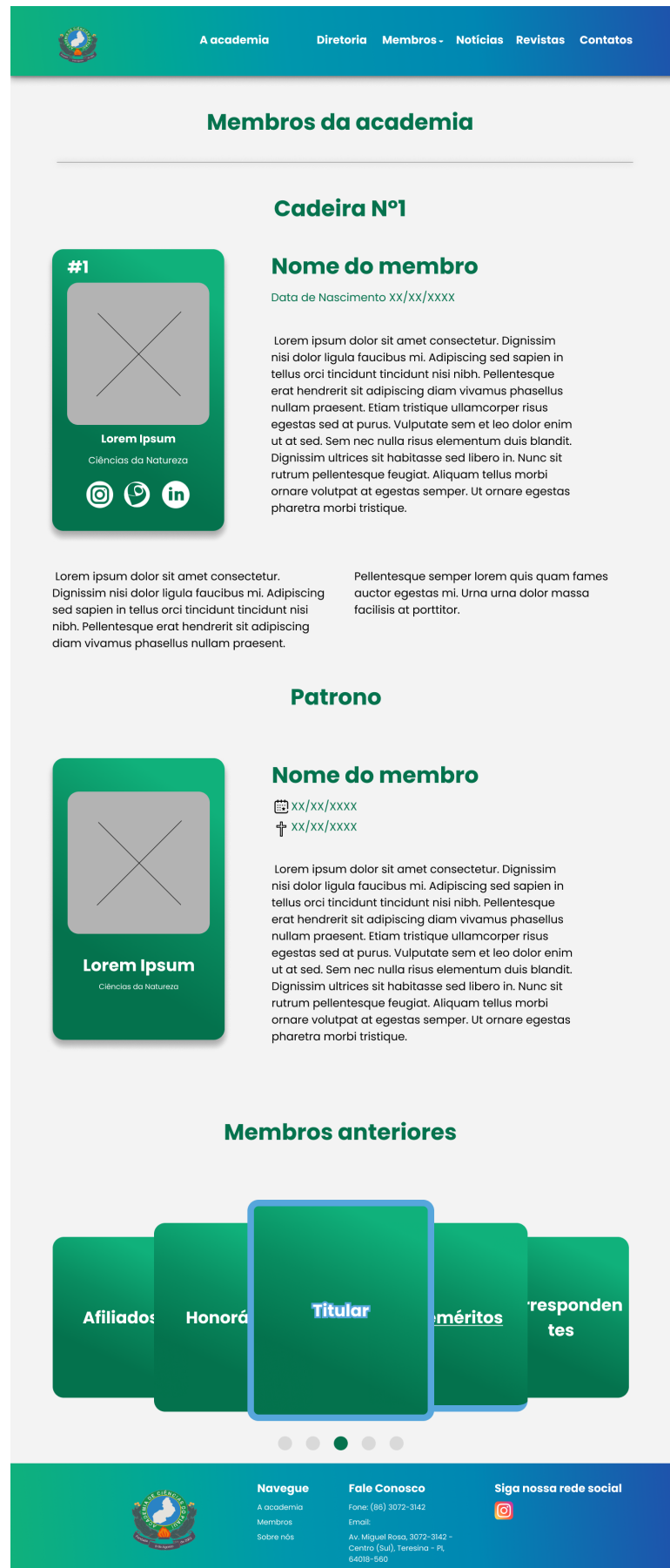
Na página de "Detalhes do Membro", ilustrada na Figura 10, foram adicionados novos recursos visuais e funcionais. Essa página agora inclui a exibição do membro patrono, destacando o legado da cadeira, além da visualização dos membros anteriores que ocuparam a mesma posição.

Figura 9 – Página de Membros



Fonte: Elaborado pelo autor.

Figura 10 – Página de Detalhe dos Membros



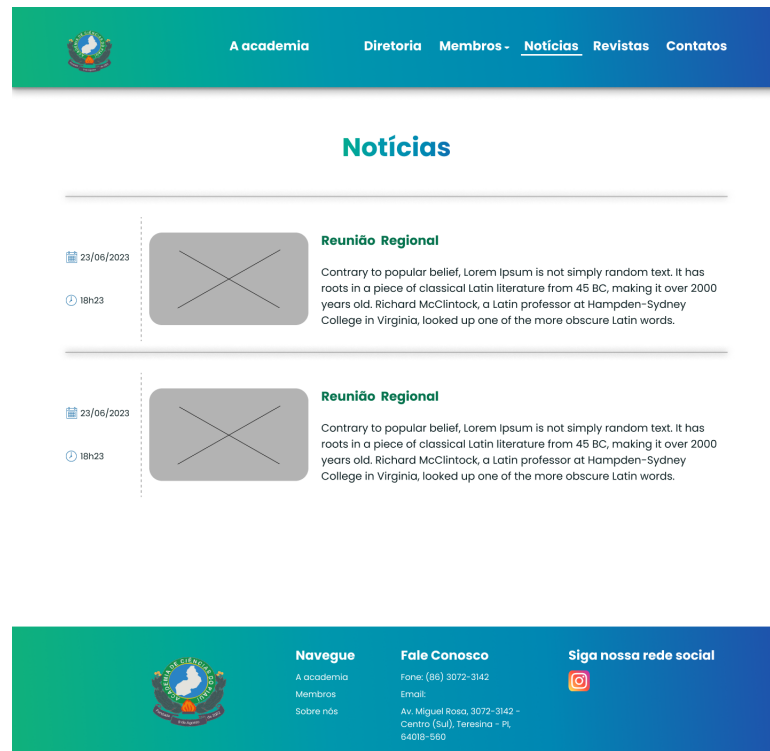
Fonte: Elaborado pelo autor.

Na Figura 10, o patrono é destacado no centro da página com uma seção dedicada à sua história, foto, data de nascimento e data de falecimento. Logo abaixo, há um carrossel exibindo os membros anteriores da cadeira. No início da página, é apresentado o atual membro titular, com uma breve biografia, nome, data de nascimento, foto, área de atuação e *links* para suas redes sociais. Esta página é de grande importância dentro do sistema, pois representa as pessoas que contribuíram para a história da Academia de Ciências do Piauí, tanto no passado quanto no presente.

Na página de "Notícias", ilustrada na Figura 11, são apresentados cartões de notícias que incluem a data e horário da publicação, uma foto, o título e uma descrição parcial. Esses elementos visam captar a atenção do visitante e incentivá-lo a acessar a página de "Detalhes da Notícia" (Figura 13). Caso o sistema não encontre nenhuma notícia, o usuário será redirecionado para a página "Notícia Não Encontrada" (Figura 12), um componente reutilizável em diversas seções da aplicação. Esse componente se adapta ao contexto em que está inserido, ajustando a mensagem exibida conforme a necessidade. Por exemplo, na página de membros, caso nenhum membro seja encontrado, o mesmo componente será utilizado, mas com a mensagem ajustada para "Membros não encontrados".

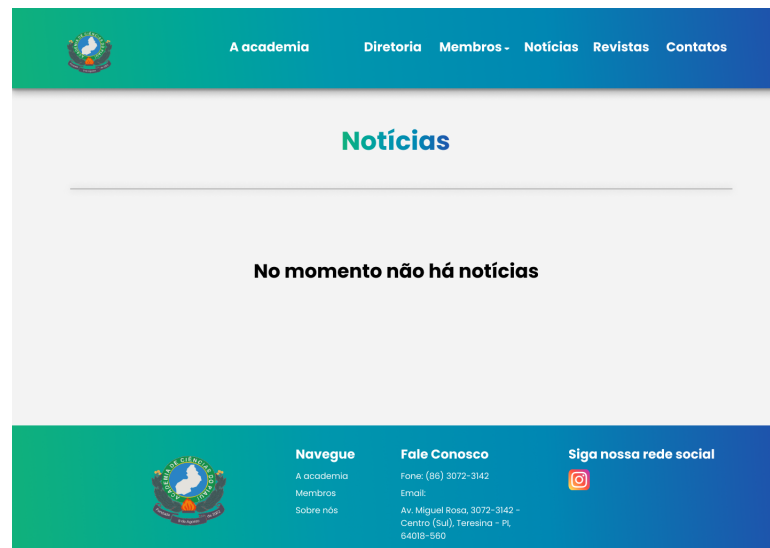
Na página de detalhes da notícia, há um carrossel para visualização de todas as imagens relacionadas à notícia. Abaixo do carrossel, são exibidos o título e a descrição completa da notícia. Adicionalmente, logo abaixo da descrição, há um link para acessar os anexos dos documentos relacionados.

Figura 11 – Página de Notícias



Fonte: Elaborado pelo autor.

Figura 12 – Página de Notícias - Variação sem notícia



Fonte: Elaborado pelo autor.

Figura 13 – Página de Detalhes da Notícia

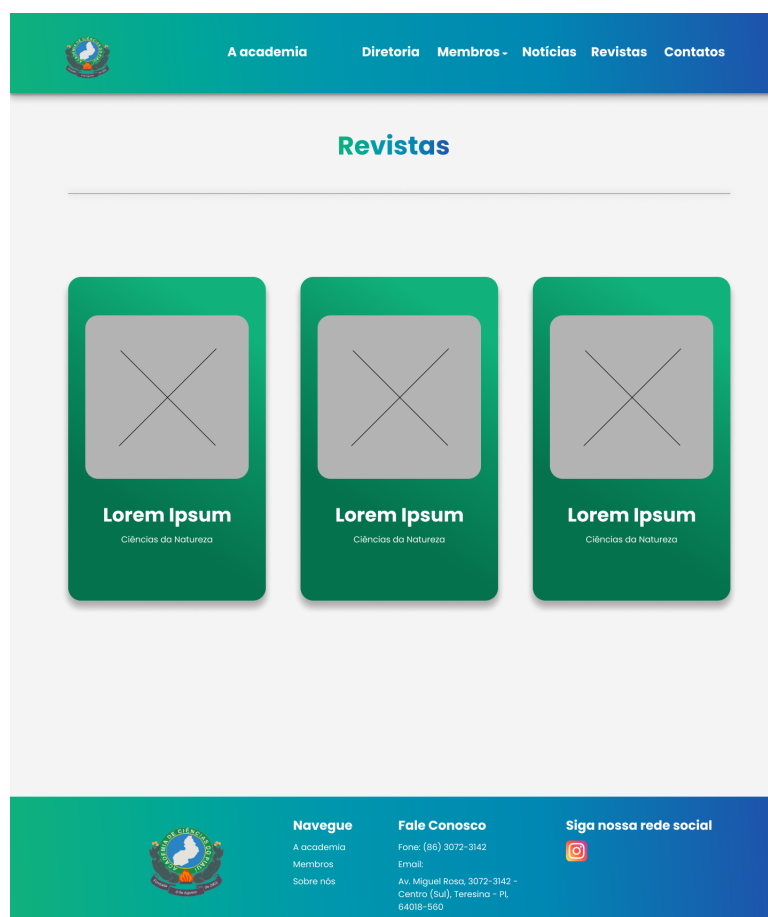


Fonte: Elaborado pelo autor.

Na página de "Revistas", ilustrada na Figura 14, os cartões de revistas são exibidos ao centro. Cada cartão é composto por uma imagem, título e descrição. Quando um cartão é clicado, o usuário é redirecionado para um site externo onde as revistas da ACIPI estão hospedadas.

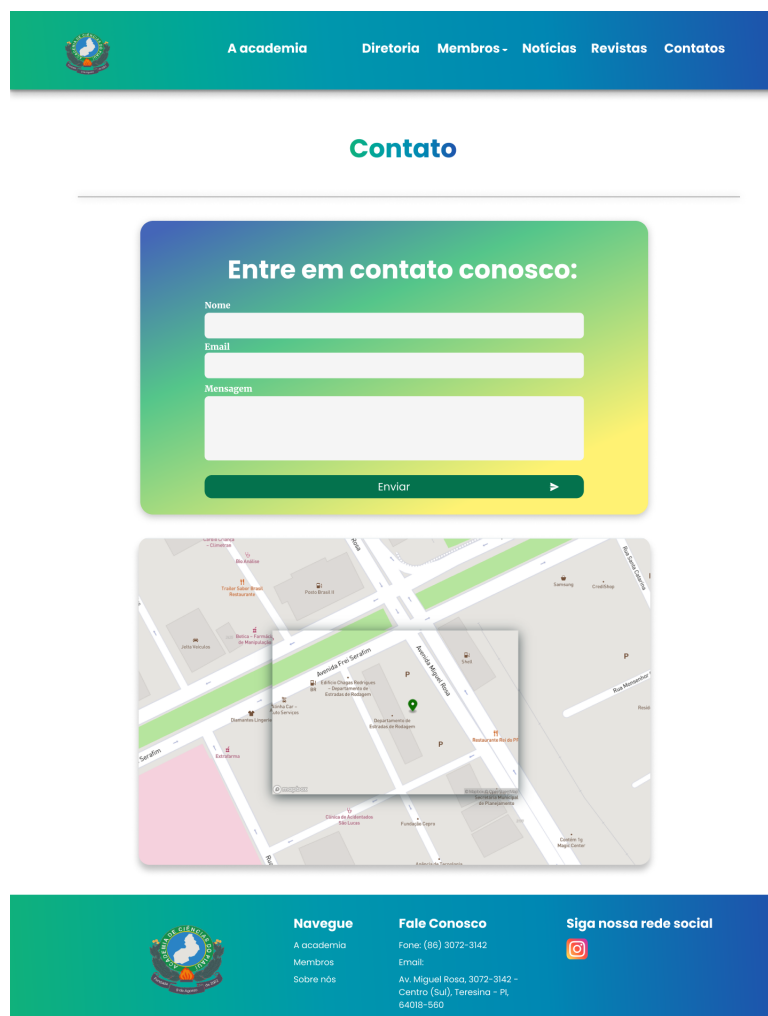
Na página de "Contato", figura 15, o usuário visitante, no formulário com o título de "Entre em contato conosco", pode submeter a ACIPI emails que serão enviados ao seu email oficial. As cores escolhidas para os gradientes do site, foram inspiradas nas cores da bandeira do Piauí, incluindo o gradiente da página de "Contato" e "Login" figura 16, logo abaixo tem-se a visualização da localização exata da Academia de Ciências do Piauí.

Figura 14 – Página das Revistas



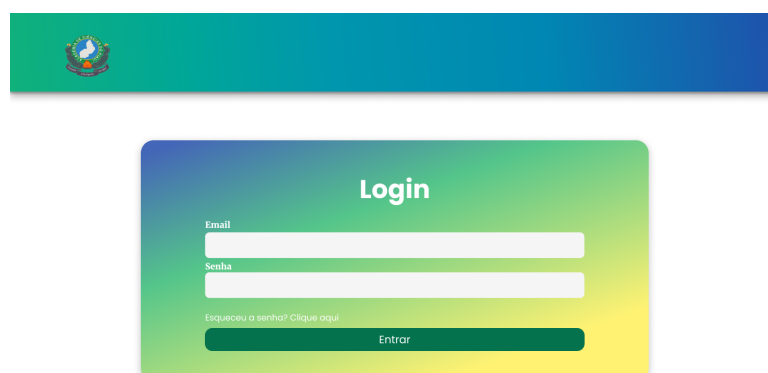
Fonte: Elaborado pelo autor.

Figura 15 – Página dos Contatos



Fonte: Elaborado pelo autor.

Figura 16 – Página de Login

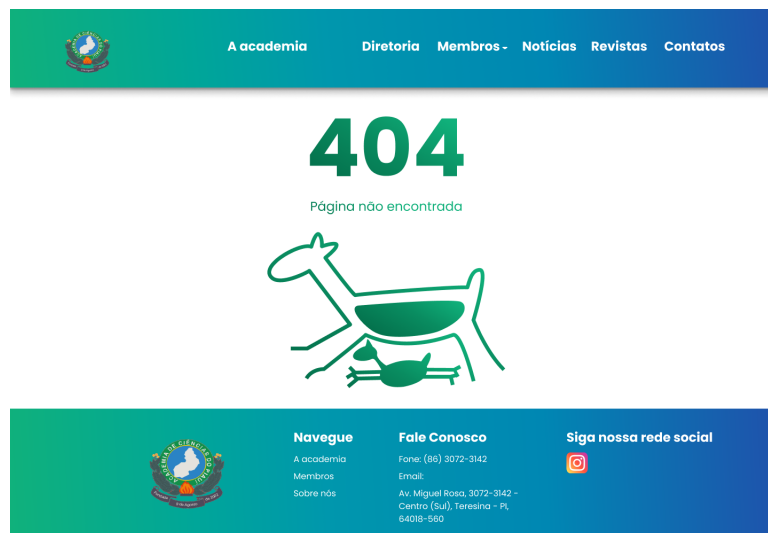


Fonte: Elaborado pelo autor.

Para os usuários que tentarem acessar uma página inexistente, foi desenvolvida uma página de "Não Encontrado", ilustrada na Figura 17. O elemento central dessa

página é uma representação estilizada da capivara, inspirada nos desenhos rupestres encontrados na Serra da Capivara. A página não só informa o usuário sobre o erro "404 - Página não encontrada", mas também incorpora um símbolo cultural importante, conectando a identidade visual do site ao patrimônio histórico da região.

Figura 17 – Página de Não Encontrado



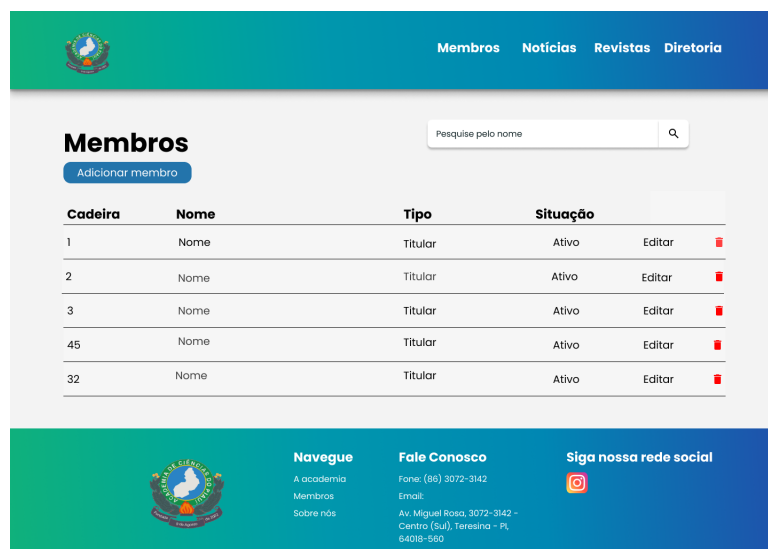
Fonte: Elaborado pelo autor.

Em seguida, serão apresentadas as páginas destinadas ao usuário administrador do sistema, incluindo os painéis de gerenciamento de membros, notícias, revistas e diretoria. Além disso, serão exibidas as páginas dedicadas à criação e edição de membros, notícias, revistas e diretoria.

As Figuras 18, 19, 20 e 21 ilustram diferentes telas de um painel administrativo, focadas na gestão de membros da Academia de Ciências do Piauí (ACIPI). Essas telas seguem um padrão consistente de interface, que visa facilitar a administração eficiente e intuitiva do sistema.

Cada tela do painel oferece ao administrador a capacidade de realizar as operações *CRUD* (Criação, Leitura, Atualização e Deleção) sobre os objetos do sistema. No contexto da gestão dos membros, notícias, revistas e diretoria, isso significa que o administrador pode adicionar novos, editar informações existentes e remover do sistema, conforme a necessidade.

Figura 18 – Paine de Membros



Membros

Pesquise pelo nome

Adicionar membro

Cadeira	Nome	Tipo	Situação	
1	Nome	Titular	Ativo	Editar
2	Nome	Titular	Ativo	Editar
3	Nome	Titular	Ativo	Editar
45	Nome	Titular	Ativo	Editar
32	Nome	Titular	Ativo	Editar

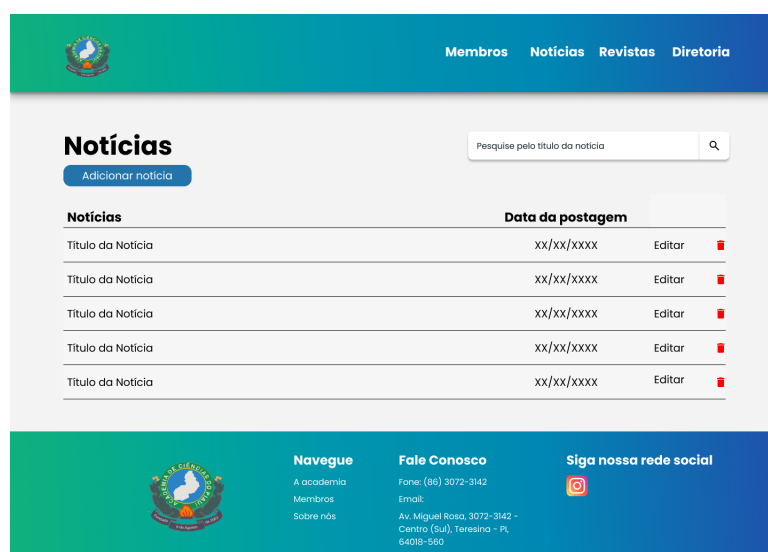
Navegue
A academia
Membros
Sobre nós

Fale Conosco
Fone: (86) 3072-3142
Email:
Av. Miguel Rosa, 3072-3142 -
Centro (Su), Teresina - PI,
64018-560

Siga nossa rede social

Fonte: Elaborado pelo autor.

Figura 19 – Paine de Notícias



Notícias

Pesquise pelo título da notícia

Adicionar notícia

Notícias	Data da postagem	
Título da Notícia	xx/xx/xxxx	Editar
Título da Notícia	xx/xx/xxxx	Editar
Título da Notícia	xx/xx/xxxx	Editar
Título da Notícia	xx/xx/xxxx	Editar
Título da Notícia	xx/xx/xxxx	Editar

Navegue
A academia
Membros
Sobre nós

Fale Conosco
Fone: (86) 3072-3142
Email:
Av. Miguel Rosa, 3072-3142 -
Centro (Su), Teresina - PI,
64018-560

Siga nossa rede social

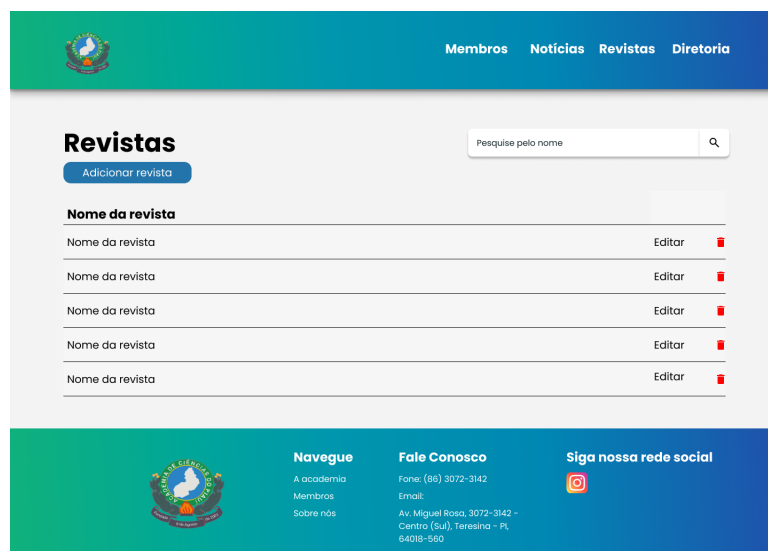
Fonte: Elaborado pelo autor.

Além das funcionalidades básicas, o painel inclui uma barra de pesquisa localizada no topo direito da tela, permitindo ao administrador localizar rapidamente um elemento específico dentro da página desejada. Essa funcionalidade é especialmente útil em cenários onde o número de elementos é grande, otimizando o tempo e a eficiência do trabalho administrativo.

Cada elemento listado na tabela é apresentado com informações essenciais, como o número da cadeira ocupada, o nome do membro, o tipo de membro (ex.: titular, associado), e sua situação atual (ativo, inativo), no caso da página de membro, mas essa estrutura segue para as demais páginas, com suas respectivas informações

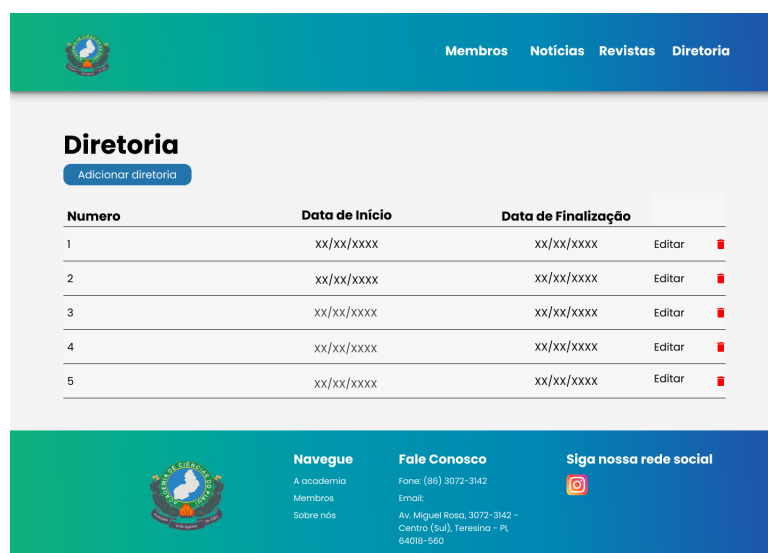
importantes. A coluna "Editar" oferece acesso rápido à funcionalidade de edição, enquanto o ícone de lixeira permite a exclusão do elemento com um único clique.

Figura 20 – Painel de Revistas



Fonte: Elaborado pelo autor.

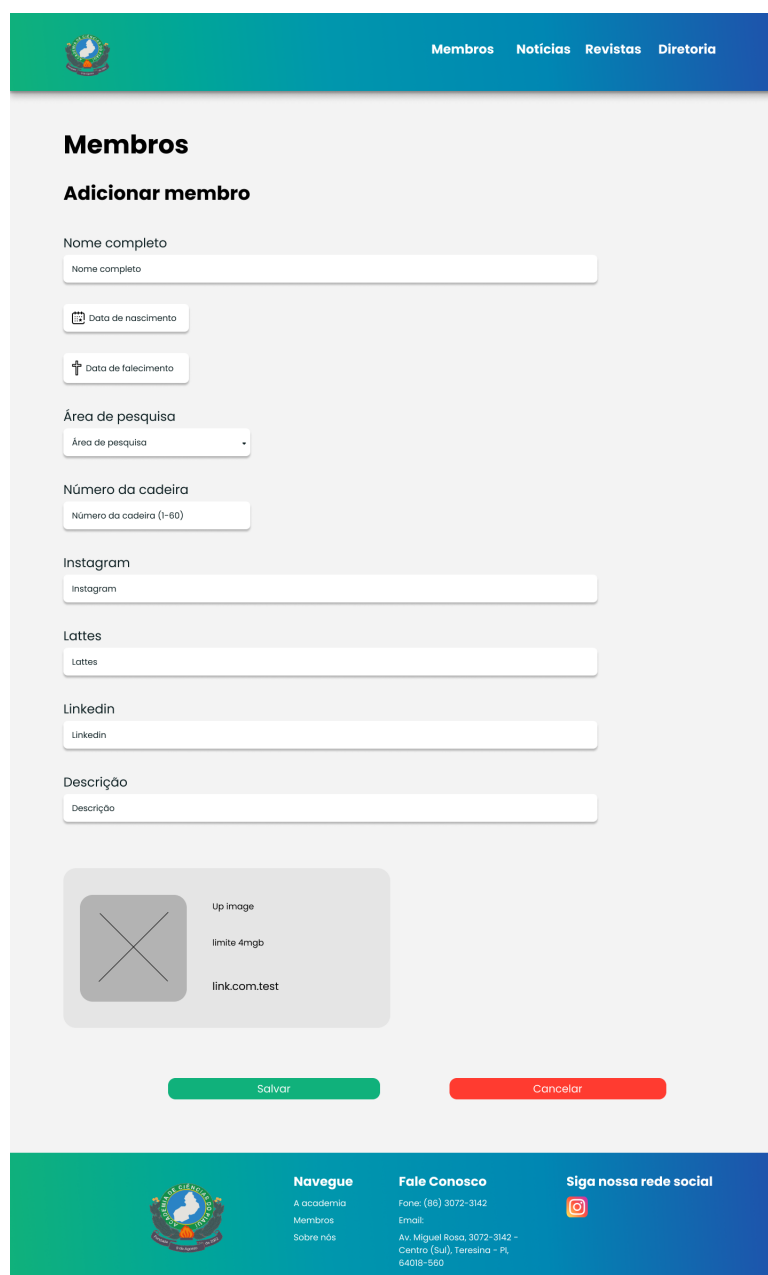
Figura 21 – Painel de Membros da Diretoria



Fonte: Elaborado pelo autor.

As Figuras 22, 23, 24 e 25 mostram formulários padronizados usados no painel administrativo da Academia de Ciências do Piauí (ACIPI) para gerenciar membros, notícias, revistas e diretoria.

Figura 22 – Página de Adicionar/Edição de Membro



Membros

Adicionar membro

Nome completo
Nome completo

Data de nascimento
Data de nascimento

Data de falecimento
Data de falecimento

Área de pesquisa
Área de pesquisa

Número da cadeira
Número da cadeira (1-60)

Instagram
Instagram

Lattes
Lattes

Linkedin
Linkedin

Descrição
Descrição

Up image
limite 4mb
link.com.test

Salvar Cancelar

Navegue
A academia
Membros
Sobre nós

Fale Conosco
Fone: (86) 3072-3142
Email:
Av. Miguel Rosa, 3072-3142 –
Centro (Sul), Teresina - PI,
64018-560

Siga nossa rede social
Instagram

Fonte: Elaborado pelo autor.

Figura 23 – Página de Adicionar/Editar Diretoria

Diretoria

Adicionar diretoria

Data de Inicio Data de Fim

Descrição

Descrição

Cargos da diretoria

Cargo

Nome do cargo Membro Remover

Cargo

Nome do cargo Remover

Adicionar cargo

Salvar Cancelar

Navegue
A academia
Membros
Sobre nós

Fale Conosco
Fone: (98) 3072-3142
Email:
Av. Miguel Rosa, 3072-3142 -
Centro (Sul), Teresina - PI,
64018-590

Siga nossa rede social

Fonte: Elaborado pelo autor.

Esses formulários são projetados para serem simples e diretos, permitindo que o administrador insira as informações necessárias para cada elemento do sistema. Por exemplo, o formulário de adicionar membros (Figura 22) inclui campos para nome completo, datas de nascimento e falecimento, área de pesquisa, número da cadeira, além de links para redes sociais como Instagram, Lattes e LinkedIn. Há também um espaço para descrição e a opção de carregar uma imagem do membro.

Figura 24 – Página de Adicionar/Editar Revistas

Revista

Adicionar revista

Nome
Título

Link da revista
Url para o link da revista

Up image
limite 4mb
link.com.test

Salvar Cancelar

Navegue
A academia
Membros
Sobre nós

Fale Conosco
Fone: (86) 3072-3142
Email:
Av. Miguel Rosa, 3072-3142 -
Centro (Sul), Teresina - PI,
64018-550

Siga nossa rede social

Fonte: Elaborado pelo autor.

De forma semelhante, o formulário de adicionar diretoria (Figura 23) possui campos para definir as datas de início e fim do mandato, adicionar uma descrição da diretoria, e incluir os cargos ocupados pelos membros, com a possibilidade de remover ou adicionar cargos conforme necessário.

As páginas para adicionar notícias (figura 25) e revistas (figura 24) seguem esse mesmo layout, proporcionando uma experiência consistente e intuitiva para o administrador. Isso facilita o gerenciamento das diversas informações do sistema de maneira eficiente e organizada.

O design dos formulários prioriza a usabilidade, com botões como "Salvar" e "Cancelar" bem visíveis, tornando o processo mais ágil.

Figura 25 – Interface - Página de Adicionar/Editar Notícias

The screenshot displays the 'Adicionar notícia' (Add news) form within the ACIPI administrative panel. The form is structured as follows:

- Header:** A green header bar contains the ACIPI logo on the left and navigation links for 'Membros', 'Notícias', 'Revistas', and 'Diretoria' on the right.
- Section Header:** The main heading is 'Notícia', followed by the sub-heading 'Adicionar notícia'.
- Form Fields:**
 - Título:** A text input field for the news title.
 - Data de postagem:** A date selection field.
 - Descrição:** A text area for the news description.
 - Arquivo:** A text input field for the archive name.
- Image Upload:** A section labeled 'Up image' with a placeholder image and instructions: 'limite 4mgb' and 'link.com.test'.
- Buttons:** Two buttons at the bottom: a green 'Salvar' (Save) button and a red 'Cancelar' (Cancel) button.
- Footer:** A green footer bar contains the ACIPI logo, contact information, and social media links.
 - Navegue:** Links to 'A academia', 'Membros', and 'Sobre nós'.
 - Fale Conosco:** Contact details including phone number (98) 3072-3142, email, and address (Av. Miguel Rosa, 3072-3142 - Centro (Su), Teresina - PI, 64018-560).
 - Siga nossa rede social:** A link to the organization's social media page.

Fonte: Elaborado pelo autor.

Em resumo, as páginas do painel administrativo da ACIPI mostram como um design padronizado e bem pensado pode fazer toda a diferença. Com um layout uniforme e campos claros, o sistema se torna acessível e fácil de usar, permitindo que os administradores façam seu trabalho de forma simples e eficiente. A possibilidade de gerenciar membros, notícias, revistas e diretoria em um único lugar torna o painel uma ferramenta indispensável para a gestão da Academia de Ciências do Piauí. A atenção aos detalhes no design e na funcionalidade garante que o sistema atenda às necessidades dos usuários de maneira prática e intuitiva, facilitando o dia a dia e contribuindo para o sucesso do sistema.

4 SOFTWARE

Após a conclusão do projeto, iniciou-se a fase de implantação do sistema, momento em que foram identificadas a necessidade de correções e melhorias.

Com o apoio dos membros da Academia de Ciências do Piauí (ACIPI), os ajustes foram realizados com sucesso. Cada alteração foi planejada para aprimorar a usabilidade e a eficiência do sistema, garantindo que ele estivesse plenamente preparado para o uso. Durante esse processo, assegurou-se que todos os requisitos previamente acordados fossem cumpridos.

Desde sua concepção, o sistema teve como objetivo principal atrair o público jovem, sem excluir outras pessoas interessadas em conhecer mais sobre a ACIPI. Novas funcionalidades e conteúdos foram incorporados para tornar a experiência mais interativa e envolvente. Um exemplo disso foi a adição de um filtro por data na página de membros da diretoria, visando facilitar a usabilidade por parte do usuário.

4.1 IMPLANTAÇÃO

Para a implantação do sistema, foi proposto aos membros responsáveis pela gestão do painel administrativo da ACIPI uma série de reuniões. Durante os encontros, foi oferecido um treinamento detalhado, onde cada passo do funcionamento do sistema foi cuidadosamente explicado. Além disso, foi estabelecido um canal de comunicação direto para que quaisquer dúvidas que surgissem pudessem ser rapidamente esclarecidas, garantindo um suporte contínuo.

A implementação foi bem-sucedida, com os responsáveis demonstrando um rápido entendimento e adaptação ao novo sistema. O feedback recebido indicou que o treinamento foi eficaz, permitindo que os administradores operassem o sistema de forma autônoma e confiante. Essa fase de implantação não apenas assegurou que o sistema fosse utilizado corretamente, mas também reforçou a importância do suporte contínuo para o sucesso na adoção do novo sistema.

5 CONSIDERAÇÕES FINAIS

Este trabalho desenvolveu um sistema web para divulgar e gerenciar as informações da Academia de Ciências do Piauí (ACIPI). A ferramenta foi criada com o objetivo de facilitar o acesso a essas informações, oferecendo uma plataforma simples e moderna. Tecnicamente, o sistema foi pensado para atender às demandas dos membros da ACIPI, seguindo boas práticas de desenvolvimento de software e utilizando tecnologias atuais.

A solução atingiu seus objetivos ao oferecer uma interface intuitiva e flexível, baseada em uma arquitetura moderna que permite criar interfaces mais dinâmicas e adaptadas aos usuários. A página web apresenta um visual agradável e utiliza estratégias de design para aumentar a visibilidade, com foco em atrair o público em geral, mas, principalmente, o público mais jovem.

Como trabalho futuro, planeja-se a adição de testes de unidade e integração no *backend*, visando garantir a robustez e a confiabilidade do sistema. Esses testes permitirão identificar e corrigir falhas de maneira mais eficiente, além de assegurar a qualidade do software à medida que novas funcionalidades forem implementadas. Além disso, a introdução de testes automatizados ajudará a reduzir o tempo de manutenção, facilitará a detecção de regressões durante atualizações e proporcionará uma maior estabilidade ao sistema, garantindo que os principais fluxos de operação continuem funcionando corretamente mesmo após mudanças no código. Além da implementação dos testes, é fundamental a criação de um modo rascunho, permitindo que os usuários do sistema administrativo realizem adições sem que essas informações sejam imediatamente exibidas para os visitantes da aplicação. Esse recurso possibilita a revisão e ajustes antes da publicação, garantindo maior controle sobre o conteúdo disponibilizado.

REFERÊNCIAS

AGGARWAL, S. *et al.* Modern web-development using reactjs. **International Journal of Recent Research Aspects**, v. 5, n. 1, p. 133–137, 2018. Acessado em: 19 set. 2024. Disponível em: <https://www.ijrra.net/Vol5issue1/IJRRRA-05-01-27.pdf>.

Amazon Web Services. **Anunciando o Amazon S3 - Simple Storage Service**. 2006. Acessado em: 01 out. 2024. Disponível em: <https://aws.amazon.com/pt/about-aws/whats-new/2006/03/13/announcing-amazon-s3---simple-storage-service/>.

ASANKA, P. **Database Design and Modeling with PostgreSQL**. [s.n.], 2020. 7 p. Acessado em: 20 out. 2024. Disponível em: https://www.researchgate.net/publication/341931233_Database_Design_and_Modeling_with_PostgreSQL.

BAFNA, S. *et al.* Review on study and usage of mern stack for web development. **International Journal for Research in Applied Science and Engineering Technology**, v. 10, n. 2, p. 178–186, 2022. Acessado em: 19 out. 2024. Disponível em: <https://www.ijraset.com/best-journal/study-and-usage-of-mern-stack-for-web-development>.

BHOSALE, S.; DAUND, K.; KALAMBE, R. Efficient data storage and retrieval using amazon web services s3. **International Journal of Research Publication and Reviews**, v. 5, n. 6, p. 3417–3421, 2024. Acessado em: 01 out. 2024. Disponível em: <https://ijrpr.com/uploads/V5ISSUE6/IJRPR30113.pdf>.

BOOTH, D. *et al.* **Web Services Architecture**. 2004. Acessado em: 12 dez. 2024. Disponível em: <https://www.w3.org/TR/2003/WD-webarch-20031209/>.

BUENO, C. A origem histórica das sociedades científicas no brasil: entidades foram fundamentais para a institucionalização da ciência e para o desenvolvimento científico no país. **Ciência e Cultura**, Sociedade Brasileira para o Progresso da Ciência, v. 74, n. 3, p. 1–6, 2022. Acessado em: 15 set. 2024. Disponível em: <http://cienciaecultura.bvs.br/pdf/cic/v74n3/v74n3a13.pdf>.

CHIBENI, S. S. O que é ciência. **Campinas: Unicamp**, 2004. Acessado em: 15 set. 2024. Disponível em: <https://unicamp.br/~chibeni/textosdidaticos/ciencia.pdf>.

EnterpriseDB Team. **PostgreSQL: An Introduction to This Modern DBMS**. 2024. Acessado em: 01 out. 2024. Disponível em: <https://www.enterprisedb.com/introduction-modern-dbms>.

FAUSTINO, A. S. *et al.* **Gerador de Aplicações**. Tese (Doutorado) — Faculdade de Ciências da Universidade de Lisboa, 2024. Acessado em: 14 out. 2024. Disponível em: <https://biblios.ciencias.ulisboa.pt/detalhes/61728>.

FORMSPREE. **Formspree - Email Forwarding for Forms**. 2024. Acessado em: 21 set. 2024. Disponível em: <https://formspree.io/>.

GOEL, A. **Founder Interviews: Anurag Goel of Render**. 2019. Acessado em: 12 dez. 2024. Disponível em: <https://hackernoon.com/founder-interviews-anurag-goel-of-render-cfcad8e92dae>.

Hossted Team. **Powering Modern Data Management with PostgreSQL: Empowering Scalability, Performance, and Reliability**. 2024. Acessado em: 01 out. 2024. Disponível em: <https://hossted.com/knowledge-base/osspeedia/data-management-and-analytics/database/powering-modern-data-management-with-postgresql-empowering-scalability-performance-and-rel>

JADHAV, G.; GONSALVES, F. Role of node.js in modern web application development. **Int. Res. J. Eng. Technol**, v. 7, n. 6, p. 6145–6150, 2020. Acessado em: 19 nov. 2024. Disponível em: <https://www.irjet.net/archives/V7/I6/IRJET-V7I61149.pdf>.

JINDAN, M. *et al.* Integrated web-based palu city blood donor service application model using reactjs and expressjs. **Advance Sustainable Science Engineering and Technology**, v. 5, n. 3, p. 02303013, 2023. Acessado em: 21 set. 2024. Disponível em: https://www.academia.edu/114505130/Integrated_Web_based_Palu_City_Blood_Donor_Service_Application_Model_Using_ReactJS_and_ExpressJS.

LEMPIÖ, A. **Enabling changeability with typescript and microservices architecture in web applications**. Dissertação (Mestrado), 2023. Acessado em: 26 nov. 2024. Disponível em: <https://trepo.tuni.fi/bitstream/handle/10024/147430/LempioAapo.pdf;jsessionid=222EB6E9B8C6B51BA18306E53400E3C4?sequence=2>.

MARTINS, M. R.; BUFFON, A. D.; NEVES, M. C. D. A astronomia na antiguidade: um olhar sobre as contribuições chinesas, mesopotâmicas, egípcias e gregas. **Revista Valore**, v. 4, n. 1, p. 810–823, 2019. Acessado em: 14 set. 2024. Disponível em: <https://revistavalore.emnuvens.com.br/valore/article/view/197/209>.

MENEGASSI, W. J. C.; RUEDA, L. H. H. Modelagem de sistema de informação para escritório de arquitetura. In: **Congresso de Tecnologia-Fatec Mococa**. [s.n.], 2024. v. 8, n. 1. Acessado em: 16 dez. 2024. Disponível em: <https://congresso.fatecmococa.edu.br/index.php/congresso/article/view/536>.

NAIM, N. I. Reactjs: An open source javascript library for front-end development. **Theseus**, Metropolia Ammattikorkeakoulu, 2017. Acessado em: 19 set. 2024. Disponível em: <https://www.theseus.fi/handle/10024/130495>.

PRESTES, B. **Design Thinking aplicado a produtos digitais**. 2021. Acessado em: 12 dez. 2024. Disponível em: <https://www.caiena.net/blog/design-thinking-produtos-digitais>.

RIBEIRO, Á. S. *et al.* Açaitour: guia turístico mobile de empreendimentos de lazer e passeio na cidade de açailândia. **Revista de Empreendedorismo e Gestão de Micro e Pequenas Empresas**, v. 9, n. 01, p. 125–140, 2024. Acessado em: 26 nov. 2024. Disponível em: <https://www.revistas.editoraenterprising.net/index.php/regmpe/article/view/736/945>.

ROY, A. React js for web developer. **Theseus**, Centria - University of Applied Sciencies, 2021. Acessado em: 22 set. 2024. Disponível em: <https://www.theseus.fi/handle/10024/511874>.

RZESZEWSKI, M. Mapbox. In: . Springer, 2023. p. 21–40. Acessado em: 28 set. 2024. Disponível em: https://link.springer.com/chapter/10.1007/978-3-031-19594-5_2.

SANTOS, A. S. *et al.* Uso de redes sociais para democratização da ciência: uma proposta chamada antroverso. **ScientiaTec**, v. 10, n. 1, 2023. Acessado em: 16 set. 2024. Disponível em: <https://periodicos.ifrs.edu.br/index.php/ScientiaTec/article/view/6533>.

SILVA, J. F. G. da; LENCASTRE, M.; CASTRO, J. Modelagem conceitual de dívida técnica na engenharia de requisitos. In: **25a Workshop em Engenharia de Requisitos (WER 24)**. [s.n.], 2024. Acessado em: 11 dez. 2024. Disponível em: https://www.inf.puc-rio.br/wer/WERpapers/artigos/artigos_WER24/WER2024_paper_16.pdf.

SILVA, L. R. d. Bancos de dados relacionais e bancos de dados chave-valor: comparação de desempenho em aplicações de troca de mensagens em tempo real. **Lume**, UFRGS, 2025. Acessado em: 10 mar. 2025. Disponível em: <https://lume.ufrgs.br/handle/10183/284326>.

SWAN, B. **Top Benefits of Using MERN Stack for Full-Stack JavaScript Development**. 2025. Acessado em: 30 set. 2025. Disponível em: https://www.goglides.dev/bella_swan_/top-benefits-of-using-mern-stack-for-full-stack-javascript-development-5b3e.

ANEXO A – DECLARAÇÃO DE ENTREGA DO CÓDIGO FONTE DO SOFTWARE A COORDENAÇÃO DO CURSO



MINISTÉRIO DA EDUCAÇÃO
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA-CENTRAL
DIRETORIA DE ENSINO
DEPARTAMENTO DE INFORMAÇÃO, AMBIENTE, SAÚDE E PRODUÇÃO ALIMENTÍCIA

DECLARAÇÃO

Declaro para devidos fins que **Halyson Itallo Cunha Pimentel**, Matrícula 2020111TADS0468, CPF 063.543.763-52, entregou o código fonte do software referente ao Relatório Técnico de Software (RTS) com título “DESENVOLVIMENTO DE UMA APLICAÇÃO WEB PARA A GESTÃO DE INFORMAÇÕES DA ACADEMIA DE CIÊNCIAS DO PIAUÍ (ACIPI)” e orientação do Prof. Francisco Marcelino Almeida de Araújo.

Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas

Autorização:

Portaria No 198, de 15 de Setembro de 2002.

Reconhecimento:

Portaria No 209, de 28 de Fevereiro de 2011 (publicado no DOU 03/03/2011)

Renovação de Reconhecimento:

Portaria SERES/MEC Nº 152, de 21 de Junho de 2023 (publicado no DOU 22/06/2023).

Teresina (PI), 06 de maio de 2025

Documento assinado digitalmente
gov.br RICARDO MARTINS RAMOS
Data: 06/05/2025 13:49:23-0300
Verifique em <https://validar.iti.gov.br>

Ricardo Martins Ramos
Coordenador do Curso Superior de Tecnologia
em Análise e Desenvolvimento de Sistemas
IFPI - Campus Teresina Central
SIAPE 1244478