

ANA CAROLINA DA CONCEIÇÃO
ANA VITÓRIA DA CONCEIÇÃO PIMENTA
GABRIELA CRISTINA DA SILVA
NABILA EDUARDA SAMPAIO ALLAH
THAYSA VITÓRIA FIRMINO DA SILVA

COMTUR

Projeto Integrador

Disciplinas Envolvidas: Engenharia de Software, Tópicos Avançado de Banco de Dados Relacional e não Relacional, Técnicas Avançadas de Programação, Técnicas Avançadas de Programação Web e Mobile, Gestão Ágil de Projetos de Software.

LISTA DE FIGURAS

Figura 1 — Menu Inferior do siste da Prefeitura	5
Figura 2 — Lista de Eventos no Siste da Prefeitura.....	5
Figura 3 — Imagens dos Eventos.....	6
Figura 4 — Informações sobre os estabelecimentos.	7
Figura 5 — Instagram da Prefeitura.	7
Figura 6 — Diagrama de Classes.	11
Figura 7 — Diagrama de caso de uso - Visão Administrador.....	22
Figura 8 — Diagrama de caso de uso - Visão do Empresário.....	22
Figura 9 — Diagrama de caso de uso - Visão do Empresário.....	23
Figura 10 — Diagrama de caso de uso - Visão do Sistema	23
Figura 11 — Diagrama de casos de uso específico – Cadastrar Tipo Turismo.....	24
Figura 12 — Diagrama de casos de uso específico – Alterar Tipo Turismo.....	26
Figura 13 — Diagrama de casos de uso específico – Excluir Tipo Turismo	28
Figura 14 — Diagrama de caso de uso específico – Consultar Tipo Turismo.....	29
Figura 15 — Diagrama de casos de uso específico – Cadastrar Tipo Atração	31
Figura 16 — Diagrama de casos de uso específico – Alterar Tipo Atração	33
Figura 17 — Diagrama de casos de uso específico – Excluir Tipo Atração	34
Figura 18 — Diagrama de casos de uso específico – Consultar Tipo Atração	36
Figura 19 — Diagrama de Sequência – Cadastrar Tipo Turismo	39
Figura 20 — Diagrama de Sequência – Alterar Tipo Turismo	39
Figura 21 — Diagrama de Sequência – Deletar Tipo Turismo.....	40
Figura 22 — Diagrama de Sequência – Listar Tipo Turismo	40
Figura 23 — Diagrama de Sequência – Cadastrar Notícia.....	41
Figura 24 — Diagrama de Sequência – Alterar Notícia.....	41
Figura 25 — Diagrama de Sequência – Deletar Notícia	42
Figura 26 — Diagrama de Sequência – Listar Noticia.....	42
Figura 27 — Imagem do primeiro cenário	43
Figura 28 — Imagem do segundo cenário	44
Figura 29 — Imagem da primeira persona.....	45
Figura 30 — Imagem da segunda persona	46
Figura 31 — Imagem do Wireframe da Tela de Login para o Adminitrador	47
Figura 32 — Imagem do Wire frame da Tela Inicial do Administrador.....	48

Figura 33 — Imagem do Wireframe da Tela de Cadastro de Noticia.....	48
Figura 34 — Imagem do Wireframe da Tela de Cadastro de Atrações	49
Figura 35 — Imagem do Protótipo da Tela de Login do Administrador	51
Figura 36 — Imagem do Prototipo da Tela Inicial do Administrador	51
Figura 37 — Imagem do Prototipo da Tela de Cadastro de Notícias.....	52
Figura 38 — Imagem do Protótipo da Tela de Cadastro de Atrações.....	53
Figura 39 — Mapeamento do Objeto Relacional.....	54
Figura 40 — Imagem da Logo do C#.....	62
Figura 41 — Logo do Visual Studio	63
Figura 42 — Logo do React	64
Figura 43 — Logo do Visual Studio Code.....	65
Figura 44 — Imagem da Tela Inicial do Administrador	68
Figura 45 — Imagem da Tela de Cadastro de Noticia	69
Figura 46 — Imagem da Tela de Cadastro de Turismo e Tipo de Turismo.....	69
Figura 47 — Imagem da Tela de Cadastro de Atrações e Tipo de Atrações	70

LISTA DE QUADROS

Quadro 1 — Descrição Classe Administrador	11
Quadro 2 — Descrição Classe Turismo	12
Quadro 3 — Descrição Classe Usuário	12
Quadro 4 — Descrição Classe Avaliação.....	13
Quadro 5 — Descrição Classe Cidade	13
Quadro 6 — Descrição Classe Notícia	14
Quadro 7 — Descrição Classe Tipo Turismo.....	14
Quadro 8 — Descrição Classe Atração	14
Quadro 9 — Descrição Classe Tipo Atração.....	15
Quadro 10 — Descrição Classe Anúncio	15
Quadro 11 — Descrição Classe Empresa.....	15
Quadro 12 — Descrição Classe Empresário	16
Quadro 13 — Lista de casos de uso do software.....	17
Quadro 14 — Documentação do caso de uso Cadastrar Tipo Turismo.	24
Quadro 15 — Documentação do caso de uso Alterar Tipo Turismo.	26
Quadro 16 — Documentação do caso de uso Excluir Tipo Turismo.	28
Quadro 17 — Documentação do caso de uso Consultar Tipo Turismo.	30
Quadro 18 — Documentação do caso de uso Cadastrar Tipo Atração	31
Quadro 19 — Documentação do caso de uso Alterar Tipo Atração.	33
Quadro 20 — Documentação do caso de uso Excluir Tipo Atração.....	35
Quadro 21 — Documentação do caso de uso Consultar Tipo Atração.....	36
Quadro 22 — Script SQL – Tabela TipoTurismo	55
Quadro 23 — Script SQL – Tabela TipoAtracao	55
Quadro 24 — Script SQL – Tabela Administrador	55
Quadro 25 — Script SQL – Tabela Noticia	55
Quadro 26 — Script SQL – Tabela Empresario.....	56
Quadro 27 — Script SQL – Tabela Usuario	56
Quadro 28 — Script SQL – Tabela Cidade.....	56
Quadro 29 — Script SQL – Tabela Avaliação	57
Quadro 30 — Script SQL – Tabela Turismo.....	57
Quadro 31 — Script SQL – Tabela Atração	58
Quadro 32 — Script SQL – Tabela Unúncio	58

Quadro 33 — Script SQL – Tabela Empresa 58

SUMÁRIO

1 INTRODUÇÃO.....	3
2 LEVANTAMENTO DE REQUISITOS DE SOFTWARE.....	4
2.1 DESCRIÇÃO DOS OBJETIVOS DO SISTEMA.....	4
2.2 DESCRIÇÃO DO SISTEMA ATUAL	5
2.3 DESCRIÇÃO DOS PRINCIPAIS PROBLEMAS	8
2.4 DESCRIÇÃO DOS REQUISITOS FUNCIONAIS	8
2.5 DESCRIÇÃO DOS REQUISITOS NÃO FUNCIONAIS	9
3 VISÃO DE CASO DE USO – UML	11
3.1 DIAGRAMA DE CLASSES	11
3.2 DICIONÁRIO DE CLASSES	11
3.3 DEFINIÇÃO DOS ATORES	16
3.4 LISTA DE CASOS DE USO.....	17
3.4. DIAGRAMA DE CASOS DE USO	21
3.5. DIAGRAMA DE CASOS DE USO INDIVIDUAIS	24
3.6. DIAGRAMA DE SEQUÊNCIA	38
4 DEFINIÇÃO DA INTERFACE COM O USUÁRIO (UX) (3º semestre).....	43
4.1 DESCRIÇÃO DE CENÁRIO.....	43
4.2 DESCRIÇÃO DE PERSONAS.....	44
4.3 ESBOÇOS DE TELA (WIREFRAMES).....	46
4.4 PROTÓTIPOS DE TELA	50
5 BANCO DE DADOS	54
5.1 MODELO ENTIDADE RELACIONAMENTO	54
5.2 SCRIPT DAS TABELAS	54
6 ARQUITETURA DE SOFTWARE.....	60
6.1 ARQUITETURA DE DESENVOLVIMENTO UTILIZADA	60
6.1.1 LINGUAGEM DE PROGRAMAÇÃO C#	62
6.1.2 VISUAL STUDIO	63
6.1.3 LINGUAGEM DE PROGRAMAÇÃO REACT	64
6.1.4 VISUAL STUDIO CODE	65
6.1.5 POSTGRESQL	66
6.2 TELAS DO SISTEMA	67
7 CONCLUSÃO.....	71
8 REFERÊNCIAS.....	72

1 INTRODUÇÃO

A internet desempenha um papel cada vez mais essencial no planejamento e promoção de destinos turísticos. Os consumidores recorrem à internet em todas as etapas de suas viagens, e muitas vezes, a gestão de destinos se limita a métricas simples baseadas no tráfego de cliques em seus sites. No entanto, a internet oferece às organizações de turismo a oportunidade de alcançar os consumidores com informações de alta qualidade, de forma rápida e econômica, independentemente de sua localização geográfica ou horário. A integração da internet nas operações de mercado e marketing é essencial para o sucesso.

A compreensão da necessidade de adotar estratégias que impulsionem o turismo, valorizando os patrimônios públicos nas cidades, é o que motivou o desenvolvimento desse sistema, um website que promove e apoia o conselho de turismo municipal (Comtur). A internet é cada vez mais utilizada como uma plataforma de divulgação e promoção de destinos turísticos pelas organizações de gestão de destinos. Ao mesmo tempo, os consumidores recorrem à internet em todas as fases de sua jornada de viagem

No entanto, a análise dos dados da web ainda é subutilizada pelas organizações de gestão de destinos, muitas vezes se limitando a examinar o tráfego de cliques em seus sites e métricas básicas. Para iniciar o desenvolvimento deste sistema, foi realizada uma análise dos pontos turísticos, notícias e eventos da cidade, proporcionando um maior entendimento das informações necessárias para sua criação.

Pesquisas demonstraram que o uso de ferramentas digitais possibilita uma maior interação e divulgação das atividades culturais relacionadas ao turismo, incluindo a disseminação de práticas educativas, divulgação de eventos culturais, preservação do patrimônio e promoção da comunidade local. Com todos os tópicos necessários devidamente organizados, o processo de desenvolvimento teve início, criando um ambiente acolhedor que instrui os usuários sobre o setor turístico e aumenta a visibilidade da cidade.

Este trabalho propõe a utilização da internet como uma aliada fundamental na divulgação de pontos turísticos, eventos e notícias relacionadas às cidades que adotarem o sistema. O software permite o cadastro de tópicos mencionados anteriormente, tornando-se um guia indispensável para quem deseja visitar uma cidade ou simplesmente conhecer mais sobre o lugar em que vivem.

2 LEVANTAMENTO DE REQUISITOS DE SOFTWARE

O levantamento de requisitos é um estágio crucial no ciclo de desenvolvimento de software, desempenhando um papel fundamental na compreensão das necessidades dos usuários e na definição clara do escopo do projeto. Para Roger S. Pressman:

A engenharia de requisitos fornece o mecanismo apropriado para entender o que o cliente deseja, analisando as necessidades, avaliando a exequibilidade, negociando uma condição razoável, especificando e gerindo os requisitos à medida que eles são transformados em um sistema operacional (PRESSMAN, 2011, p.118).

Conforme observado por Sommerville (2018), o levantamento de requisitos não apenas estabelece a base fundamental, mas também se configura como um estágio essencial no ciclo de vida do desenvolvimento de software. Sua importância reside na capacidade de construir um software que atenda às demandas dos usuários e agregue valor ao negócio. Esse processo crítico assegura o sucesso do projeto ao garantir uma compreensão clara e abrangente das expectativas e requisitos do cliente, delineando diretrizes cruciais para a construção de um sistema que verdadeiramente satisfaça as necessidades e objetivos estabelecidos.

2.1 DESCRIÇÃO DOS OBJETIVOS DO SISTEMA

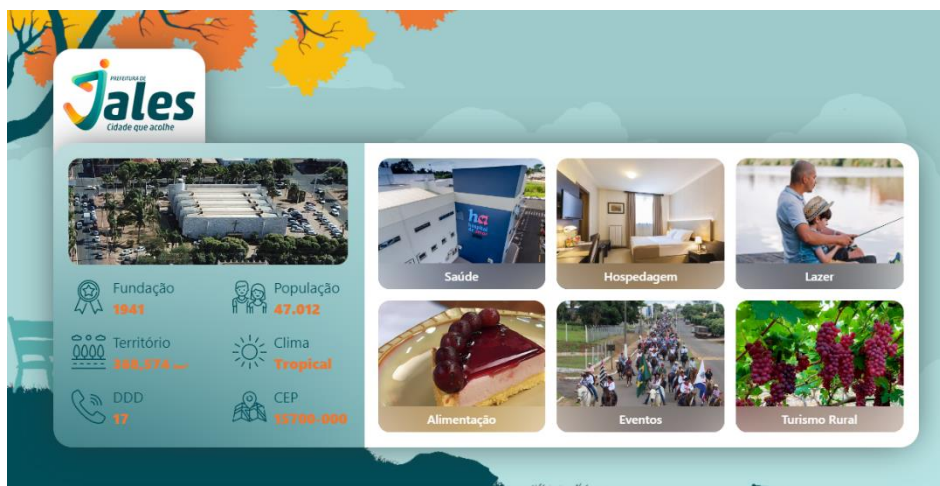
Os conceitos e ideias mencionados ganham vida por meio do desenvolvimento de um software concebido para tornar o uso do site mais acessível na identificação das principais atrações turísticas, notícias e eventos da cidade. Além disso, o software tem como missão proporcionar à comunidade em geral um acesso facilitado ao conhecimento mais profundo do patrimônio público de sua própria cidade. O Comtur, em sua essência, atua como um apoio fundamental para o setor de turismo, o qual carece de maior visibilidade e aprimoramento em relação aos seus pontos-chave.

Ao analisar as implicações do aprimoramento mencionado anteriormente, torna-se evidente o impacto positivo nos aspectos econômicos e sociais. Isso se traduz na capacidade de gerar empregos tanto diretamente relacionados ao turismo quanto de maneira indireta, na criação de novas oportunidades de negócios, na distribuição de renda e na melhoria da qualidade de vida dos residentes e visitantes.

2.2 DESCRIÇÃO DO SISTEMA ATUAL

Atualmente a prefeitura não conta com um site exclusivo para o turismo, apenas uma página dentro do website geral da cidade. Nessa página as categorias são divididas em: Saúde, Hospedagem, Lazer, Alimentação, Eventos e Turismo Rural, com um menu inferior que ao selecionar qual categoria deseja acessar, redireciona a parte da página relacionada.

Figura 1 — Menu Inferior do siste da Prefeitura



Fonte: Site da Prefeitura de Jales.

Os Eventos dentro da página não possuem descrições próprias e sua data, sendo apresentado apenas como uma lista.

Figura 2 — Lista de Eventos no Siste da Prefeitura

Eventos

- Aniversário da Cidade
- Cavalgada Ecológica
- Romaria Diocesana
- Festa dos Motoristas e Encontro dos Caminhoneiros
- Feira da Uva e do Mel
- Manhã de lazer nos bairros
- Feira de Arte – mensal
- Feira Nacional de Artesanato
- Festa do Bon Odori
- Festa do Sukiyaki
- Natal Iluminado (Decoração Natalina)

Fonte: Site da Prefeitura de Jales.

As fotos no site não possuem um padrão de tamanho e ocupam boa parte da tela, atrapalhando o fluxo de leitura das postagens.

Figura 3 — Imagens dos Eventos



Fonte: Site da Prefeitura de Jales.

As postagens e informações dentro do site estão mal distribuídas, os estabelecimentos que correspondem as categorias não recebem destaque, fotos ou descrição, contando apenas com nome, endereço e telefone, além de que essas informações estão desatualizadas.

Figura 4 — Informações sobre os estabelecimentos

-Hotel Caseiro Jales

Endereço: R. Nova Iorque, 1914 – Vila Santa Ignês, Jales – SP, 15703-298

Telefone: (17) 99661-3070

-Hotel Terraço

Endereço: R. Dez, 2548 – Centro, Jales – SP, 15700-068

Telefone: (17) 3632-6806

-Hotel Espaço Vida

Endereço: Rua Suécia, 1973, Anexo ao Hospital do Amor – Entrada Pelo Hospital do Amor, Jales – SP, 15706-286

Telefone: (17) 3621-5976

Alimentação

-Picanharia Jales

Endereço: R. das Palmeiras, 1604 – Vila Pinheiro, Jales – SP, 15704-050

Telefone: (17) 3632-2095

-Clube do Garfo

Endereço: R. Onze, 2252 – Centro, Jales – SP, 15700-030

Telefone: (17) 3632-2746

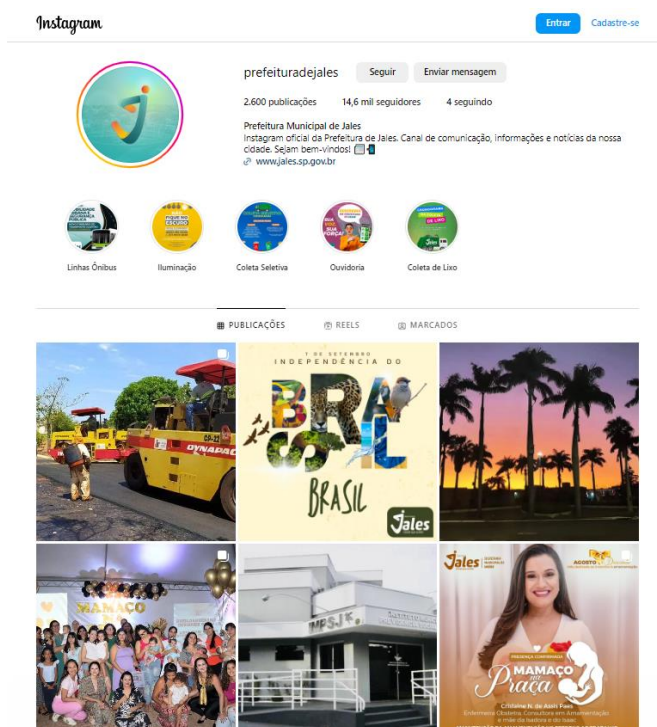
-Restaurante Saborita Jales

Endereço: Av. Francisco Jalles, 67 – Jardim Romero, Jales – SP, 15703-220

Fonte: Site da Prefeitura de Jales.

Outra plataforma que é utilizada para divulgação da cidade, é o Instagram da prefeitura, onde é publicado os eventos e mudanças que ocorrem na cidade. Diferente da página do turismo no site, o Instagram é atualizado.

Figura 5 — Instagram da Prefeitura



Fonte: Instagram da Prefeitura.

2.3 DESCRIÇÃO DOS PRINCIPAIS PROBLEMAS

O software foi desenvolvido com o propósito de promover e impulsionar o turismo em Jales. Seu principal objetivo é destacar os pontos turísticos da cidade que, embora sejam desconhecidos para a maioria, oferecem experiências incríveis. No entanto, enfrentamos desafios e questões que podem afetar a experiência dos visitantes.

Uma das principais preocupações é a falta de divulgação dos locais turísticos menos conhecidos e dos estabelecimentos comerciais locais. Além disso, há muitos espaços privados na cidade que têm potencial para se tornarem atrações turísticas, mas seus proprietários não demonstram interesse em abrir suas portas ao público.

Outro desafio significativo é que o turismo local tem sido predominantemente relacionado à área de saúde, com muitos visitantes chegando à cidade devido a hospitais, especialmente o Hospital de Amor, que atende pacientes de várias regiões. O grande desafio é encontrar maneiras de encantar essas pessoas, de modo que retornem a Jales não apenas por motivos de saúde, mas também para explorar a cidade que, de alguma forma, as conquistou.

2.4 DESCRIÇÃO DOS REQUISITOS FUNCIONAIS

Os requisitos funcionais de um sistema definem suas ações e funcionalidades, sendo influenciados pelo tipo de software em desenvolvimento, pelos usuários esperados e pela abordagem global adotada pela organização ao formular os requisitos. Quando apresentados como requisitos de usuários e gerentes do sistema, é crucial que os requisitos funcionais forneçam uma descrição detalhada das funções do sistema, suas entradas, saídas e tratamento de exceções.

O Comtur destaca-se pela sua capacidade de conectar os pontos turísticos e culturais do município de Jales com os seus utilizadores. Com base nesse princípio, os seus requisitos funcionais delineiam os serviços que a aplicação deve oferecer.

Botões de Ação: Os botões de ação, tais como incluir, alterar e excluir registros, desempenham um papel crucial na organização e melhoria das informações dentro do sistema. Eles permitem a inclusão de novos registros de pontos turísticos, a atualização de informações sobre eventos e até mesmo a remoção de monumentos da lista.

Sistema de Pesquisa: O sistema de pesquisa é uma característica fundamental do Comtur e será implementado através de campos de pesquisa e menus que guiarão os utilizadores. Dada a complexidade da arquitetura do Comtur, é evidente a necessidade de um

sistema de busca rápido e eficiente que ajude os utilizadores a encontrar o que estão procurando. O sistema proposto seguirá uma estrutura de categorização de conteúdo por grupos e subgrupos.

Acessibilidade: O Comtur é projetado para se adaptar à configuração de usabilidade definida pelo utilizador. A inclusão e a acessibilidade para todos os utilizadores são prioridades no desenvolvimento deste projeto. Para alcançar esse objetivo, o Comtur possui uma arquitetura robusta de funcionalidades de inclusão.

Controle de Interações: O sistema incluirá um filtro de controle de ações dentro do site, abrangendo interações dos utilizadores, comentários e avaliações. Esse controle permite monitorar as ações dos utilizadores e intervir em situações que violem os direitos da aplicação.

Conexão com Sistemas Externos: O Comtur estabelecerá conexões com sistemas externos, como o portal de notícias da prefeitura, geolocalização e mapas, ampliando assim a sua utilidade e alcance.

Leitor de QR Code: Uma característica adicional do Comtur é o leitor de QR Code, que permitirá a leitura e direcionamento para a página de conteúdo relevante, tornando a experiência dos utilizadores mais interativa e eficaz.

2.5 DESCRIÇÃO DOS REQUISITOS NÃO FUNCIONAIS

Segundo Wilson de Pádua Paula Filho (2015), os requisitos não funcionais abrangem os requisitos de desempenho e outros atributos de qualidade do produto, incluindo requisitos lógicos relacionados aos dados persistentes e requisitos de natureza técnica, como restrições ao design e implementação. A recomendação é expressar os requisitos não funcionais de maneira precisa e, sempre que possível, quantitativa. Mesmo que seja desafiador formular valores específicos durante o levantamento de requisitos para a primeira versão de um produto, essa precisão é crucial.

Compatibilidade com Navegadores e Dispositivos: O Comtur foi projetado para ser acessível através de uma ampla gama de navegadores, incluindo os mais recentes, como o Chrome, bem como outros navegadores populares. Além disso, o sistema é totalmente adaptável, funcionando perfeitamente tanto em dispositivos móveis quanto em desktops, garantindo que os usuários possam aproveitar todas as funcionalidades, independentemente do dispositivo que estiverem utilizando.

Banco de Dados: Para otimizar nossos recursos e reduzir custos, o Comtur utilizará um banco de dados gratuito para armazenar informações. Isso permite que o sistema gerencie dados de forma eficiente, sem comprometer a qualidade do serviço.

Tecnologia de Desenvolvimento: O Comtur será desenvolvido utilizando a linguagem C#, proporcionando um ambiente robusto e versátil que impulsionará a criação de um sistema de alta qualidade.

Desempenho e Carregamento Rápido: Priorizamos um desempenho eficiente no Comtur. Nosso sistema foi meticulosamente projetado para garantir um carregamento rápido, proporcionando aos usuários uma experiência ágil e satisfatória.

Disponibilidade e Atualizações: Mantemos a disponibilidade do Comtur em um nível mínimo de 99,8%, para que os usuários possam contar com seu acesso ininterrupto. Em caso de erros ou problemas, comprometemo-nos a implementar atualizações e correções em um prazo máximo de 7 dias, garantindo a continuidade e o aprimoramento do serviço.

Compatibilidade com Diversos Dispositivos: O Comtur foi projetado para ser compatível com uma ampla variedade de dispositivos, atendendo às diversas necessidades dos usuários, independentemente do hardware que possuam.

Segurança: A segurança é nossa prioridade máxima no Comtur. Implementamos rigorosas medidas de segurança para proteger os dados e a privacidade dos usuários, garantindo que suas informações permaneçam estritamente confidenciais e protegidas. Para alcançar esse objetivo, nossa equipe de administradores supervisiona, modera e valida todas as publicações. Todas as contribuições ao Comtur passam por um minucioso processo de revisão antes de serem disponibilizadas publicamente. Essa abordagem rigorosa garante que apenas conteúdo seguro e apropriado seja exibido, contribuindo para a criação de um ambiente online confiável e protegido.

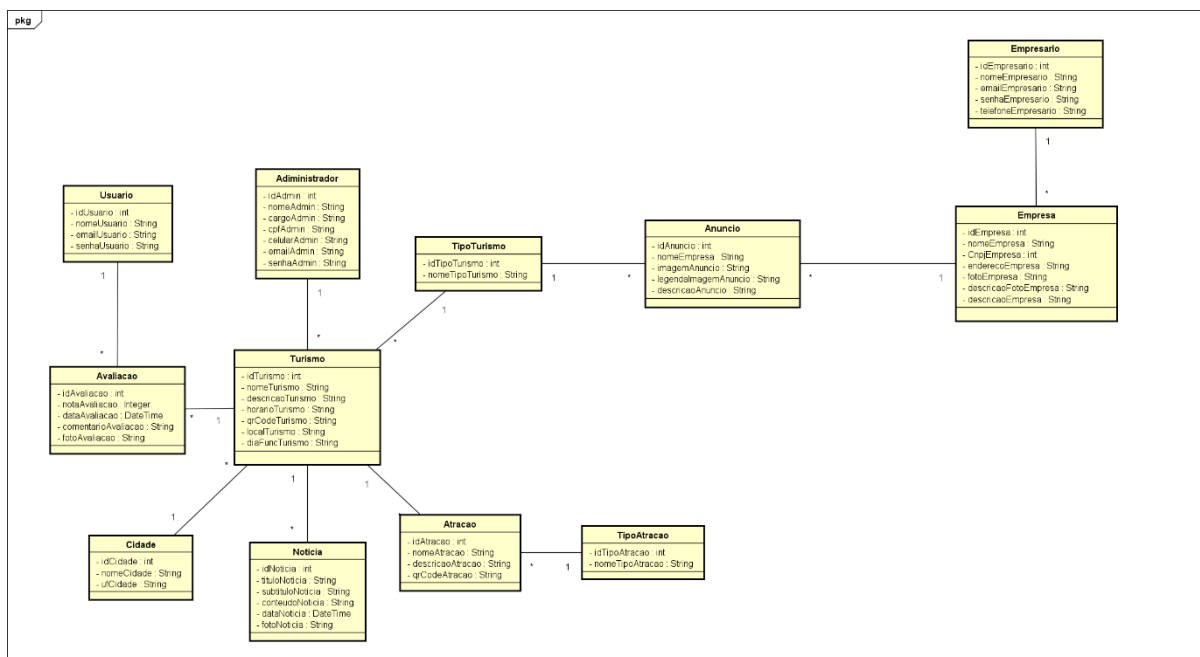
3 VISÃO DE CASO DE USO – UML

3.1 DIAGRAMA DE CLASSES

No contexto do livro "UML2: uma abordagem prática" de Guedes, o diagrama de classes destaca-se como um dos elementos mais cruciais da UML, desempenhando um papel fundamental ao oferecer suporte para a maioria dos demais diagramas. Conforme abordado na obra, esse diagrama assume a responsabilidade de definir a estrutura das classes utilizadas pelo sistema, delineando seus atributos, métodos e estabelecendo as inter-relações entre as classes. (GUEDES, 2018).

Nesta subseção, será exibido o diagrama de classes do sistema, que engloba a configuração da camada de negócios correspondente.

Figura 6 — Diagrama de Classes



Fonte: Elaborado pelos autores.

3.2 DICIONÁRIO DE CLASSES

Administrador: tem como objetivo gerenciar o acesso, disponibilizar conteúdo aos usuários e avaliar as métricas de utilização do sistema.

Quadro 1 — Descrição Classe Administrador

Atributo	Tipo	Descrição
----------	------	-----------

IdAdmin	Integer	Código que irá identificar o administrador.
NomeAdmin	String	Nome de identificação
CargoAdmin	String	Identificação de cargo dentro da prefeitura
CpfAdmin	String	Documento de reconhecimento
CelularAdmin	String	Número de contato do administrador
EmailAdmin	String	Endereço de e-mail do administrador, utilizado para realizar o login ao site
SenhaAdmin	String	Senha de acesso a plataforma

Fonte: Desenvolvidos pelos autores.

Turismo: tem como objetivo o controle dos pontos turísticos inseridos ao software, a verificação e atualização de informação a serem apresentadas aos utentes da plataforma.

Quadro 2 — Descrição Classe Turismo

Atributo	Tipo	Descrição
IdTurismo	Integer	Código que irá identificar o ponto turismo.
NomeTurismo	String	Nome de identificação ao local
DescricaoTurismo	String	Descrição e guia ao local
HorarioTurismo	String	Horário de funcionamento
QrCodeTurismo	String	Qrcode para projeção e descrição do ponto turístico visitado.
LocalTurismo	String	Endereço do local
DiaFuncTurismo	String	Dias de funcionamento

Fonte: Desenvolvidos pelos autores.

Usuários: classe responsável pelo gerenciamento dos interessados, que tem como projeção a interação com o site e realização do uso de suas ferramentas de busca.

Quadro 3 — Descrição Classe Usuário

Atributo	Tipo	Descrição
IdUsuario	Integer	Código que irá identificar o usuário.

NomeUsuario	String	Nome de identificação
EmailUsuario	String	Endereço de e-mail do usuário, utilizado para realizar o login ao site
SenhaUsuario	String	Senha de acesso a plataforma

Fonte: Desenvolvidos pelos autores.

Avaliação: encarregada do armazenamento e gerenciamento de satisfação e avaliações dentro da plataforma.

Quadro 4 — Descrição Classe Avaliação

Atributo	Tipo	Descrição
IdAvaliacao	Integer	Código que irá identificar avaliação do usuário
NotaAvaliacao	Integer	Nota em numeração
DataAvaliacao	DateTime	Data da avaliação
ComentarioAvaliacao	String	Comentário do utente
FotoAvaliacao	String	Fotografia do local com comentário de avaliação.

Fonte: Desenvolvidos pelos autores.

Cidades: Categoria encarregada do gerenciamento das cidades turísticas pertencentes ao site.

Quadro 5 — Descrição Classe Cidade

Atributo	Tipo	Descrição
IdCidade	Integer	Código que irá identificar cidade
NomeCidade	String	Nome de identificação
UfCidade	String	Siglas de identificação

Fonte: Desenvolvidos pelos autores.

Notícias: categoria encarregada do gerenciamento de notícias relacionadas a cidades, eventos e pontos turísticos.

Quadro 6 — Descrição Classe Notícia

Atributo	Tipo	Descrição
IdNoticia	Integer	Código que irá identificar notícia
TituloNoticia	String	Título de referenciamento de assunto
SubtituloNoticia	String	Subtítulo de referenciamento de assunto
ConteudoNoticia	String	Assunto da notícia
DataNoticia	DateTime	Data de publicação
FotoNoticia	String	Fotografia de referência ao assunto

Fonte: Desenvolvidos pelos autores.

Tipo Turismo: Categoria encarregada de identificar a natureza do turismo adicionado ao site, como Pontes, Praças, entre outros.

Quadro 7 — Descrição Classe Tipo Turismo

Atributo	Tipo	Descrição
IdTipoTurismo	Integer	Código que irá identificar tipo turismo
NomeTipoTurismo	String	Nome de identificação

Fonte: Desenvolvidos pelos autores.

Atração: visa gerenciar as atrações associadas aos eventos ou pontos turísticos do site.

Quadro 8 — Descrição Classe Atração

Atributo	Tipo	Descrição
IdAtracao	Integer	Código que irá identificar a atração
NomeAtracao	String	Nome de identificação
DescricaoAtracao	String	Descrição da atração
QrCodeAtracao	String	QR Code utilizado para projetar e fornecer informações sobre a atração visitada.

Fonte: Desenvolvidos pelos autores.

Tipo Atração: categoria responsável por identificar a natureza da atração adicionada ao site, como circo, show, entre outros.

Quadro 9 — Descrição Classe Tipo Atração

Atributo	Tipo	Descrição
IdTipoAtracao	Integer	Código que irá identificar tipo atração
NomeTipoAtracao	String	Nome de identificação

Fonte: Desenvolvidos pelos autores.

Anúncio: Categoria encarregada do gerenciamento dos anúncios divulgados na aplicação por empresas parceiras.

Quadro 10 — Descrição Classe Anúncio

Atributo	Tipo	Descrição
IdAnuncio	Integer	Código que irá identificar anúncio
NomeEmpresa	String	Identificação da empresa responsável pelo anúncio
ImagemAnuncio	String	Imagem de referência ao anúncio
LegendaImagemAnuncio	String	Legenda de referenciamento a imagem
DescricaoAnuncio	String	Descrição do anúncio publicado

Fonte: Desenvolvidos pelos autores.

Empresa: categoria encarregada do gerenciamento das empresas parceiras, cuja projeção envolve a interação com o site e a divulgação de seus anúncios.

Quadro 11 — Descrição Classe Empresa

Atributo	Tipo	Descrição
IdEmpresa	Integer	Código que irá identificar empresa
NomeEmpresa	String	Razão social da empresa
CnpjEmpresa	Integer	Cadastro Nacional da Pessoa Jurídica
EnderecoEmpresa	String	Endereço da empresa

FotoEmpresa	String	Imagem da empresa
DescricaoFotoEmpresa	String	Legenda da imagem
DescricaoEmpresa	String	Atuação da empresa

Fonte: Desenvolvidos pelos autores.

Empresário: encarregado de analisar o perfil de sua empresa, a eficácia de seus anúncios e a avaliação dos usuários em relação às suas postagens.

Quadro 12 — Descrição Classe Empresário

Atributo	Tipo	Descrição
IdEmpresario	Integer	Código que irá identificar empresa
NomeEmpresario	String	Nome de identificação
EmailEmpresario	String	Endereço de e-mail, utilizado para realizar o login ao site
SenhaEmpresario	String	Senha de acesso a plataforma
TelefoneEmpresario	String	Telefone de contato
CpfEmpresario	String	Documento de identificação

Fonte: Desenvolvidos pelos autores.

3.3 DEFINIÇÃO DOS ATORES

O diagrama de caso de uso se concentra em dois itens principais: os atores e os casos de uso. Os atores podem representar qualquer usuário, algum hardware especial ou mesmo um sistema integrado que poderá utilizar, de alguma maneira, as funcionalidades e serviços do sistema em questão. A representação gráfica utiliza símbolos, como bonecos, para identificar atores, acompanhados de uma breve descrição que detalham suas funções no sistema. Esta abordagem visual facilita a compreensão dos papéis desempenhados por diferentes componentes, a descrição associada a cada ator destaca sua função específica no contexto do sistema, facilitando a análise e compreensão durante o processo de modelagem, seguindo as orientações apresentadas no livro "UML 2 - Uma Abordagem Prática".

Usuário: responsável pelo uso da aplicação. Esse usuário deve realizar o cadastro, e interagir com o sistema de maneira que avalie o ponto turístico deixando um comentário.

Empresário: desempenha o papel de inserir dados referentes a pontos turísticos (estabelecimentos) e atrações privadas, sendo encarregado de manter a atualização constante dessas informações.

Administrador: Este usuário desempenha um papel crucial no Sistema de Gestão de Pontos Turísticos, sendo o responsável por supervisionar e manter a integridade do site. Suas responsabilidades abrangem o gerenciamento abrangente de todos os aspectos do sistema, incluindo pontos turísticos, notícias, eventos e atrações públicas.

3.4 LISTA DE CASOS DE USO

Nessa subseção serão apresentadas todas as ações que os usuários poderão realizar dentro da aplicação.

Quadro 13 — Lista de casos de uso do software

Nº	Descrição do Caso de Uso	Entrada	Caso de Uso	Resposta
1	Alterar Administrador	informa dados administrador	Alterar administrador	Retorna mensagem “Alterado com sucesso”
2	Administrador altera evento	Informa dados de evento	Alterar evento	Retorna mensagem “Alterado com sucesso”
3	Administrador altera notícia	Informa dados de notícia	Alterar notícia	Retorna mensagem “Alterado com sucesso”
4	Administrador altera ponto turístico	Informa dados de ponto turístico	Alterar ponto turístico	Retorna mensagem “Alterado com sucesso”
5	Administrador altera portal transparência	Informa dados portal transparência	Altera portal transparência	Retorna mensagem “Alterado com sucesso”
6	Administrador altera tipo turismo	Informa dados tipo turismo	Altera tipo turismo	Retorna mensagem “Alterado com sucesso”
7	Administrador altera tipo atração	Informa dados tipo atração	Altera tipo atração	Retorna mensagem “Alterado com sucesso”
8	Administrador cadastra ponto turístico	Informa dados de ponto turístico	Cadastrar ponto turístico	Retorna mensagem “Cadastrado com sucesso”

Nº	Descrição do Caso de Uso	Entrada	Caso de Uso	Resposta
9	Administrador cadastra portal transparência	Informa dados de portal transparência	Cadastrar portal transparência	Retorna mensagem “Cadastrado com sucesso”
10	Administrador cadastra administrador	Informa dados de administrador	Cadastrar administrador	Retorna mensagem “Cadastrado com sucesso”
11	Administrador cadastra usuário comum	Informa dados de usuário comum	Cadastrar usuário comum	Retorna mensagem “Cadastrado com sucesso”
12	Administrador cadastra notícia	Informa dados de notícia	Cadastrar notícia	Retorna mensagem “Cadastrado com sucesso”
13	Administrador cadastra evento	Informa dados de evento	Cadastrar evento	Retorna mensagem “Cadastrado com sucesso”
14	Administrador desativa administrador	Id administrador	Desativa administrador	Retorna mensagem” Desativado com sucesso”
15	Administrador desativa ponto turístico	Id ponto turístico	Desativar ponto turístico	Retorna mensagem” Desativado com sucesso”
16	Administrador desativa portal transparência	Id portal transparência	Desativar portal transparência	Retorna mensagem” Desativado com sucesso”
17	Administrador desativa evento	Id evento	Desativar evento	Retorna mensagem” Desativado com sucesso”
18	Administrador desativa notícia	Id notícia	Desativar notícia	Retorna mensagem” Desativado com sucesso”
19	Administrador exclui tipo turismo	Id tipo turismo	Excluir tipo Turismo	Retorna mensagem ” Excluído com sucesso”
20	Administrador exclui tipo atração	Id tipo atração	Excluir tipo atração	Retorna mensagem ” Excluído com sucesso”

Nº	Descrição do Caso de Uso	Entrada	Caso de Uso	Resposta
21	Administrador consulta portal transparência	Id portal transparência	Consultar portal transparência	Retorna dados portal transparência
22	Administrador consulta ponto turístico	Id ponto turístico	Consultar ponto turístico	Retorna dados ponto turístico
23	Administrador consulta notícia	Id notícia	Consultar notícia	Retorna dados notícia
24	Administrador consulta evento	Id evento	Consultar evento	Retorna dados evento
25	Administrador consulta administrador	Id administrador	Consultar administrador	Retorna dados administrador
26	Administrador consulta tipo turismo	Id tipo turismo	Consultar tipo turismo	Retorna dados tipo turismo
27	Administrador consulta tipo atração	Id tipo atração	Consultar tipo atração	Retorna dados tipo atração
28	Empresário cadastra empresário	Informa dados do empresário	Cadastrar empresário	Retorna mensagem “Cadastrado com sucesso”
29	Empresário cadastra empresa	Informa dados da empresa	Cadastrar empresa	Retorna mensagem “Cadastrado com sucesso”
30	Empresário cadastra anúncio	Informa dados anúncio	Cadastrar anúncio	Retorna mensagem “Cadastrado com sucesso”
31	Empresário excluir empresário	Id empresário	Excluir empresário	Retorna mensagem “Excluído com sucesso”
32	Empresário excluir empresa	Id empresa	Excluir empresa	Retorna mensagem “Excluído com sucesso”
33	Empresário excluir anúncio	Id anúncio	Excluir anúncio	Retorna mensagem “Excluído com sucesso”
34	Empresário altera empresário	Informa dados do empresário	Alterar empresário	Retorna mensagem “Alterado com sucesso”
35	Empresário altera anúncio	Informa dados do anúncio	Alterar anúncio	Retorna mensagem “Alterado com sucesso”
36	Empresário altera empresa	Informa dados da empresa	Alterar empresa	Retorna mensagem

Nº	Descrição do Caso de Uso	Entrada	Caso de Uso	Resposta
				“Alterado com sucesso”
37	Usuário cadastra avaliação	Informa dados da avaliação	Cadastrar avaliação	Retorna mensagem “Cadastrado com sucesso!”
38	Usuário cadastra usuário	Informa dados usuário	Cadastrar usuário	Retorna mensagem “Cadastrado com sucesso”
39	Usuário consultar evento	Id evento	Consultar evento	Retorna dados do evento
40	Usuário consultar notícia	Id notícia	Consultar notícia	Retorna dados da notícia
41	Usuário consultar ponto turístico	Id ponto turístico	Consultar ponto turístico	Retorna dados do ponto turístico
42	Usuário consultar portal transparência	Id portal transparência	Consultar portal transparência	Retorna dados do portal transparência
43	Usuário consultar usuário	Id usuário	Consultar usuário	Retorna dados do usuário
44	Usuário excluir avaliação	Id avaliação	Excluir avaliação	Retorna mensagem “Excluído com sucesso”
45	Usuário login usuário	Informa dados de acesso	Efetuar login	Retorna mensagem “Login realizado com sucesso”
46	Usuário altera avaliação	Informa dados avaliação	Alterar avaliação	Retorna mensagem “Alterado com sucesso”
47	Usuário altera usuário	Informa dados do usuário	Alterar usuário	Retorna mensagem “Alterado com sucesso”
48	Sistema lista administrador		Listar administrador	Retorna lista administrador
49	Sistema lista avaliação		Listar avaliação	Retorna lista avaliação
50	Sistema lista Portal Transparência		Listar Portal Transparência	Retorna lista Portal Transparência
51	Sistema lista Tipo Evento		Listar Tipo Evento	Retorna lista tipo evento

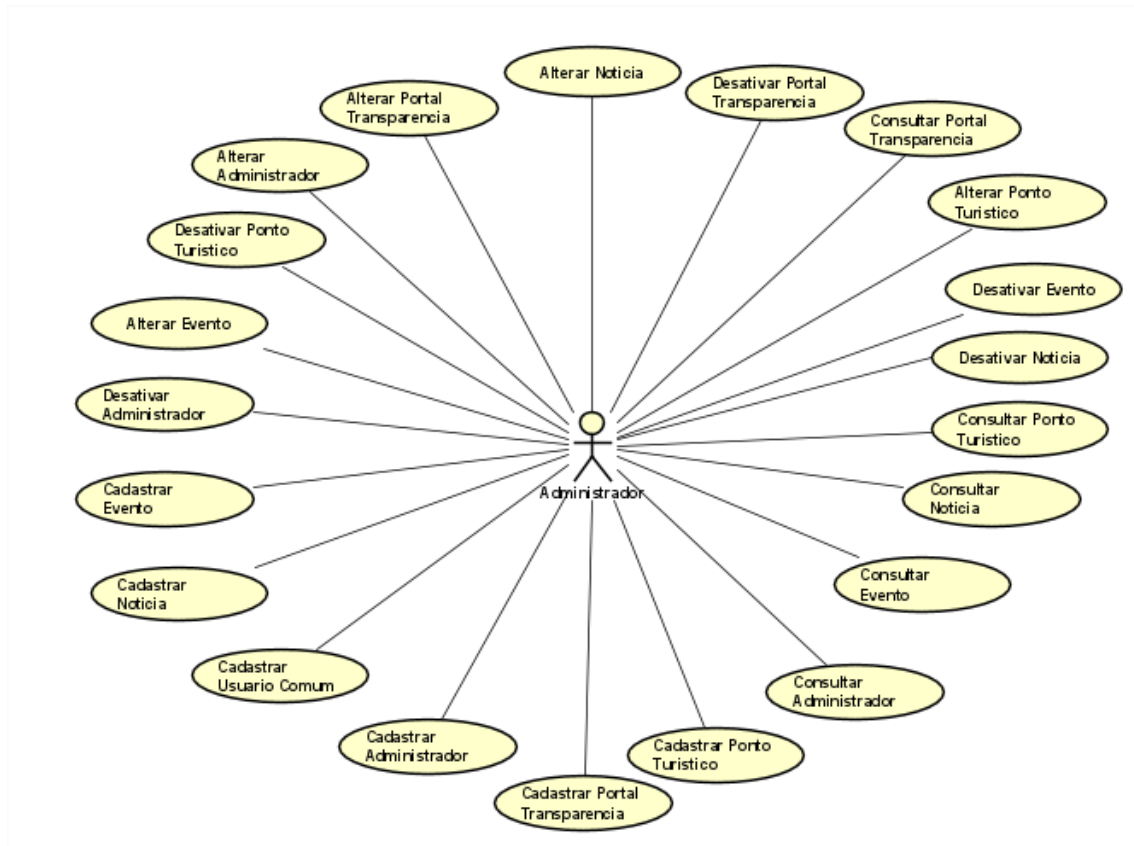
Nº	Descrição do Caso de Uso	Entrada	Caso de Uso	Resposta
52	Sistema lista usuários		Listar usuários	Retorna lista usuários
53	Sistema lista cidade		Listar Cidade	Retorna lista cidade
54	Sistema lista usuário		Listar usuário	Retorna dados usuário
55	Sistema lista anúncio		Listar Anúncio	Retorna lista anúncio
56	Sistema lista notícia		Listar notícia	Retorna lista notícia
57	Sistema lista dados empresa		Listar Dados Empresa	Retorna dados empresa
58	Sistema lista tipo atração		Listar tipo atração	Retorna dados Tipo atração
59	Sistema lista evento		Listar evento	Retorna dados evento
60	Sistema lista dados empresário		Listar empresário	Retorna dados empresário
61	Sistema lista tipo turismo		Listar tipo turismo	Retorna dados tipo turismo

Fonte: Desenvolvidos pelos autores.

3.4. DIAGRAMA DE CASOS DE USO

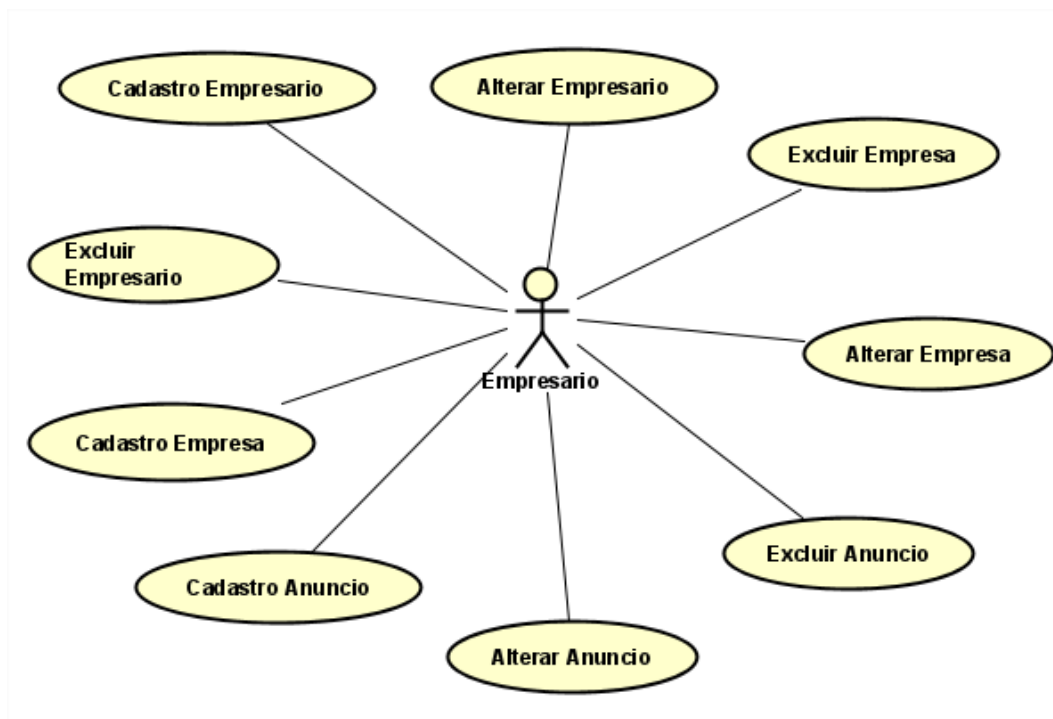
Conforme destacado no livro "UML 2 - Uma Abordagem Prática", o Diagrama de Caso de Uso é o mais abstrato, portanto, o mais flexível e informal, utilizado para representar uma visão abrangente do sistema. Seu propósito é identificar os atores que interagirão com o software, juntamente com suas respectivas funcionalidades. Este diagrama é comumente empregado durante as fases de levantamento e análise de requisitos, embora sua aplicação se estenda ao longo de todo o processo de modelagem, podendo inclusive servir como fundamento para a elaboração de outros diagramas.

Figura 7 — Diagrama de caso de uso - Visão Administrador



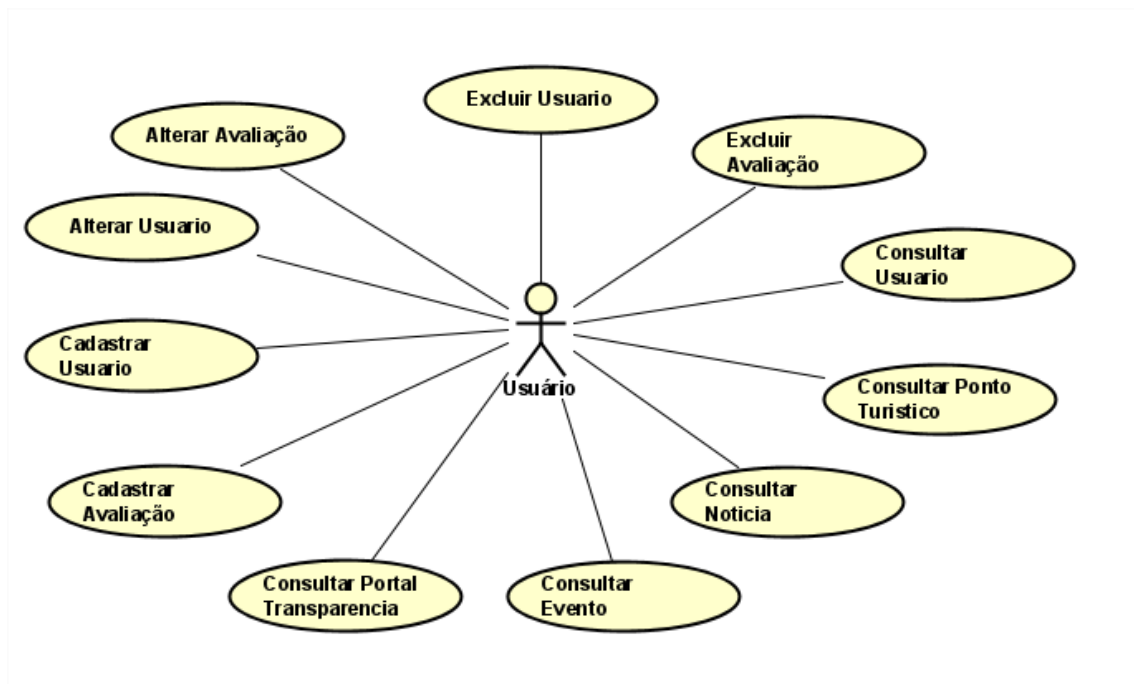
Fonte: Elaborado pelos autores.

Figura 8 — Diagrama de caso de uso - Visão do Empresário



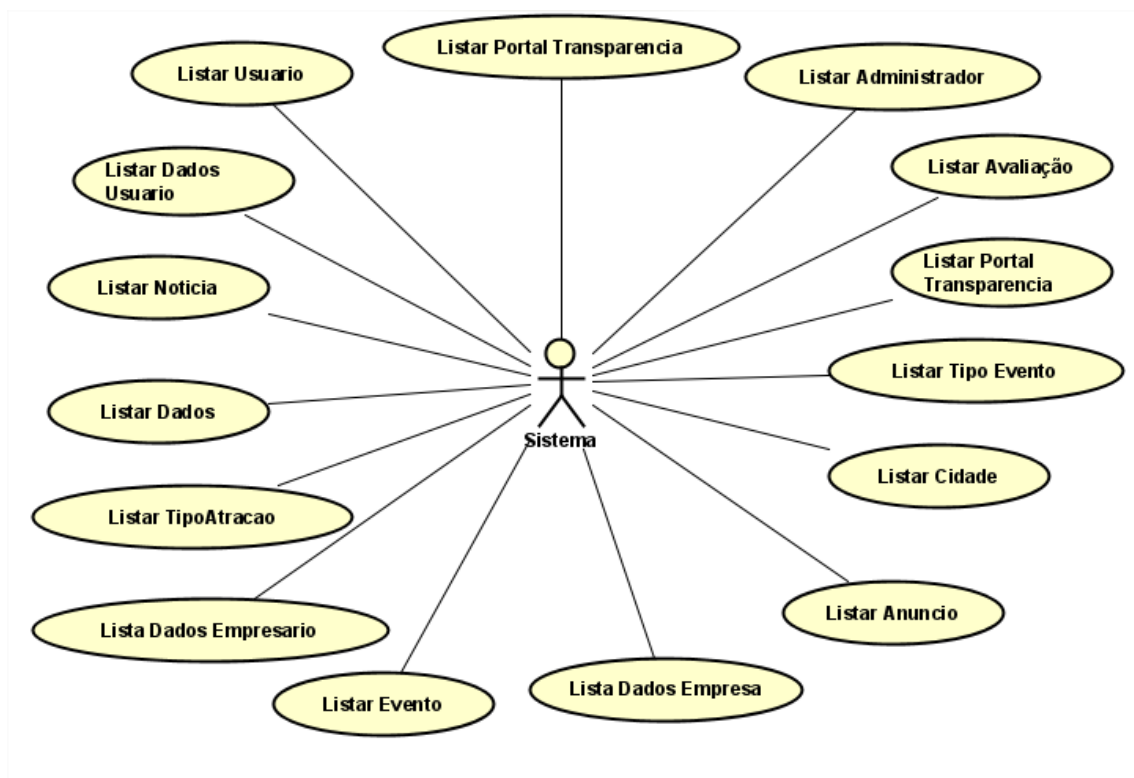
Fonte: Elaborado pelos autores.

Figura 9 — Diagrama de caso de uso - Visão do Empresário



Fonte: Elaborado pelos autores.

Figura 10 — Diagrama de caso de uso - Visão do Sistema

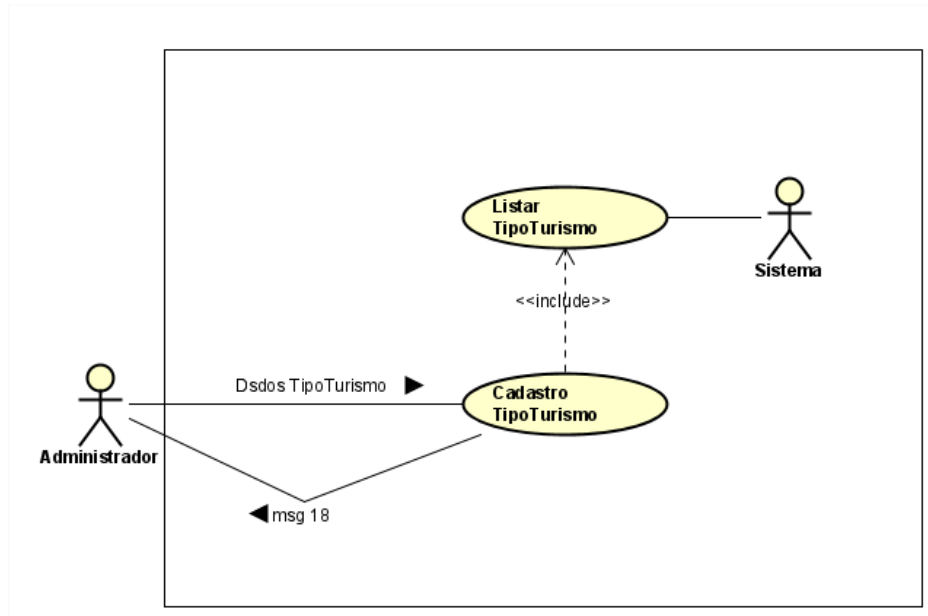


Fonte: Elaborado pelos autores.

3.5. DIAGRAMA DE CASOS DE USO INDIVIDUAIS

Os diagramas apresentados nessa subseção visam as ações específicas do usuário Administrador na inserção de tipo turismo e tipo atração à aplicação.

Figura 11 — Diagrama de casos de uso específico – Cadastrar Tipo Turismo



Fonte: Elaborado pelos autores.

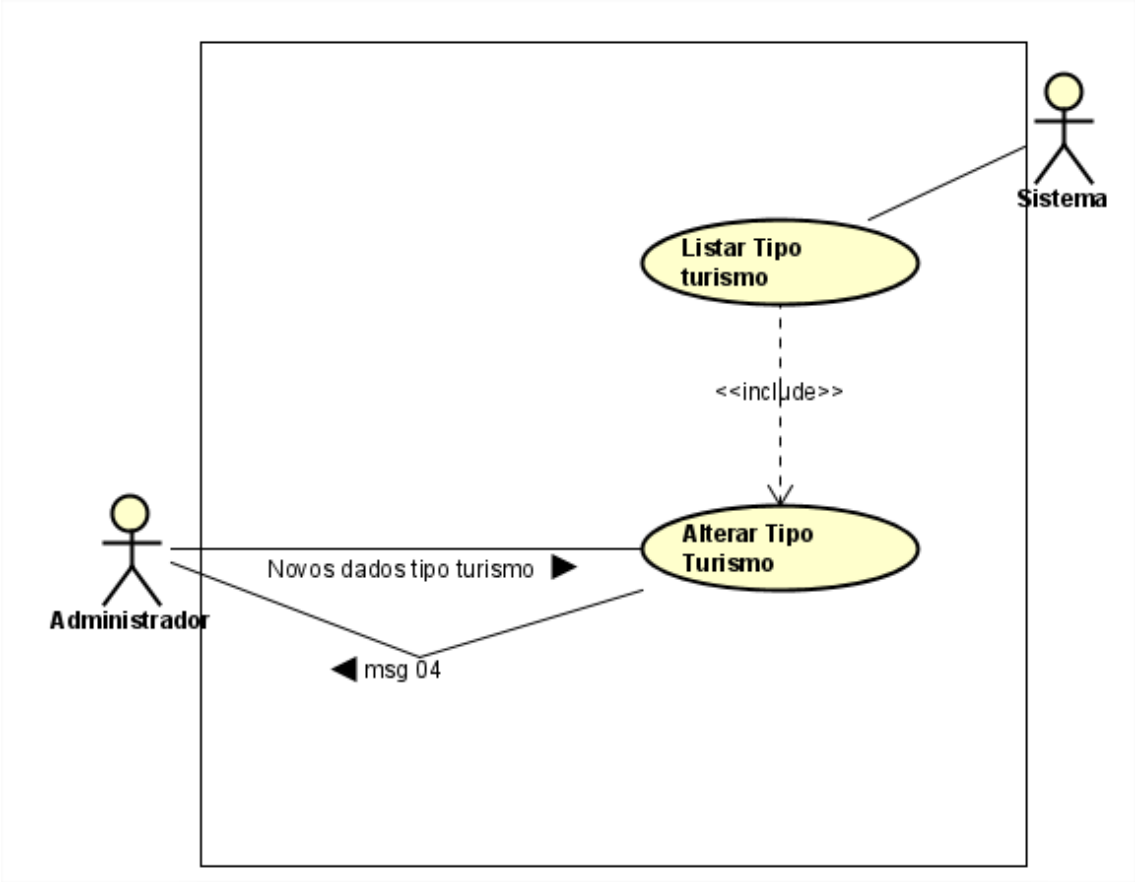
Quadro 14 — Documentação do caso de uso Cadastrar Tipo Turismo

Documentação	
Nome do caso de uso	Cadastrar Tipo Turismo
Ator principal	Usuário (Administrador)
Resumo	Esse caso de uso descreve as ações necessárias para que o usuário (Administrador) adicione novos tipos turismo
Pré-condições	Preencher todos os campos do formulário de inserção
Pós-condições	O usuário é direcionado ao módulo correspondente do sistema
Fluxo normal	

Ações do ator	Ações do sistema
1. Usuário solicita a inserção de um novo tipo turismo	
	2. Sistema recebe solicitação e mostra tela com formulário de inserção
3. Usuário informa dados do tipo turismo ao formulário e clica em “Salvar”	
	4. Sistema valida campos e dados adicionados
	5. Sistema retorna mensagem “Tipo Turismo adicionado!”
Fluxo alternativo	
	4. Sistema valida campos e dados
	4.1. Sistema identifica inconsistência de dados informados ou campos em branco
	4.1.1. Sistema envia mensagem “Preencha todos os campos”
	4.1.2. Sistema retorna ao fluxo normal pelo item 2

Fonte: Elaborado pelos autores.

Figura 12 — Diagrama de casos de uso específico – Alterar Tipo Turismo



Fonte: Elaborado pelos autores.

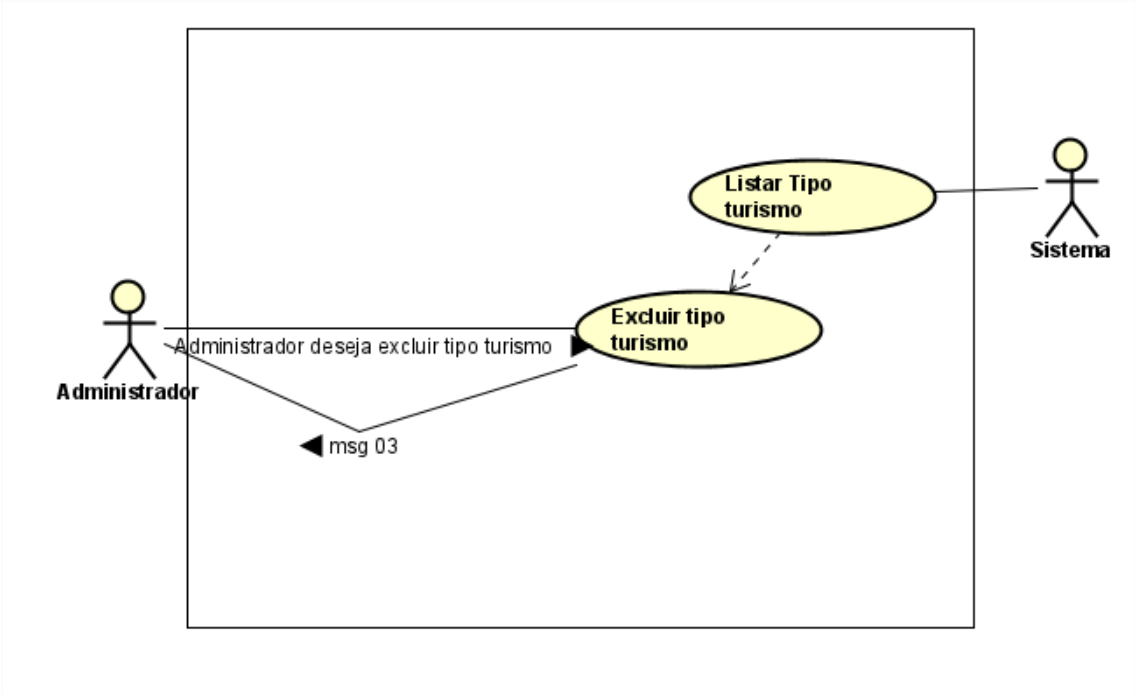
Quadro 15 — Documentação do caso de uso Alterar Tipo Turismo

Documentação	
Nome do caso de uso	Alterar Tipo turismo
Ator principal	Usuário (Administrador)
Resumo	Esse caso de uso descreve as ações necessárias para que o usuário (Administrador) altere informações referentes aos tipos turismo adicionadas anteriormente.
Pré-condições	Tipo turismo cadastrado
Pós-condições	O usuário é direcionado ao módulo correspondente do sistema

Fluxo normal	
Ações do ator	Ações do sistema
1. Usuário solicita a alteração de um tipo turismo.	
	2. Sistema recebe solicitação e mostra tela com os dados do tipo turismo.
3. Usuário altera os dados do tipo turismo e seleciona o botão “Alterar”	
	4. Sistema valida solicitação e altera dados.
	5. Sistema envia mensagem “Alteração realizada com sucesso”
	6. Sistema retorna para a tela de listagem.
Fluxo alternativo	
	4. Sistema valida solicitação e altera dados.
	4.1. Sistema falha ao conectar-se com o banco de dados
	4.2. Sistema envia mensagem “Erro de conexão com o banco de dados”
	4.3. Sistema retorna ao fluxo normal pelo item 2

Fonte: Elaborado pelos autores.

Figura 13 — Diagrama de casos de uso específico – Excluir Tipo Turismo



Fonte: Elaborado pelos autores.

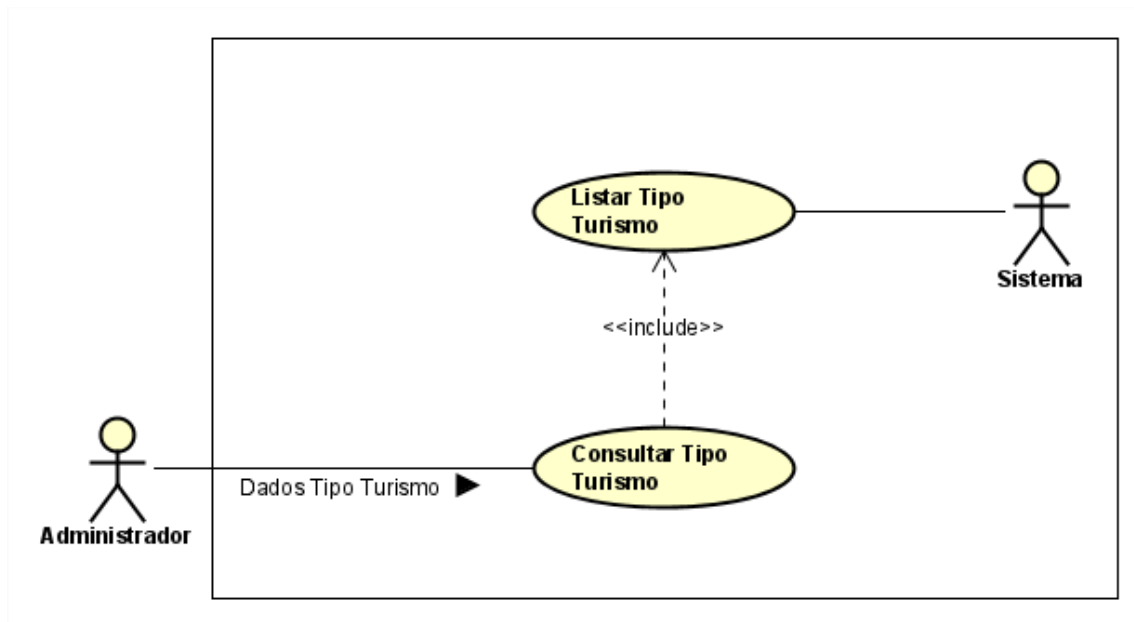
Quadro 16 — Documentação do caso de uso Excluir Tipo Turismo

Documentação	
Nome do caso de uso	Excluir Tipo turismo
Ator principal	Usuário (Administrador)
Resumo	Esse caso de uso descreve as ações necessárias para que o usuário (Administrador) apague informações referentes aos tipos turismo cadastrados.
Pré-condições	Tipos turismo cadastrados
Pós-condições	O usuário é direcionado ao módulo correspondente do sistema
Fluxo normal	
Ações do ator	Ações do sistema
1. Usuário solicita exclusão dos tipos turismo cadastrados	

	2. Sistema recebe solicitação e mostra tela com os tipos turismo
3. Usuário seleciona tipo turismo que deseja ter acesso às informações.	
	4. Sistema valida solicitação
	5. Sistema direciona usuário ao módulo correspondente de operações
6. Usuário seleciona botão “Excluir”	
	7. Sistema carrega tela de exclusão.
Fluxo alternativo	
	4. Sistema lista solicitação
	4.1. Sistema falha ao conectar-se com o banco de dados
	4.1.1. Sistema envia mensagem “Erro de conexão com o banco de dados”
	4.1.2. Sistema retorna ao fluxo normal

Fonte: Elaborado pelos autores.

Figura 14 — Diagrama de caso de uso específico – Consultar Tipo Turismo



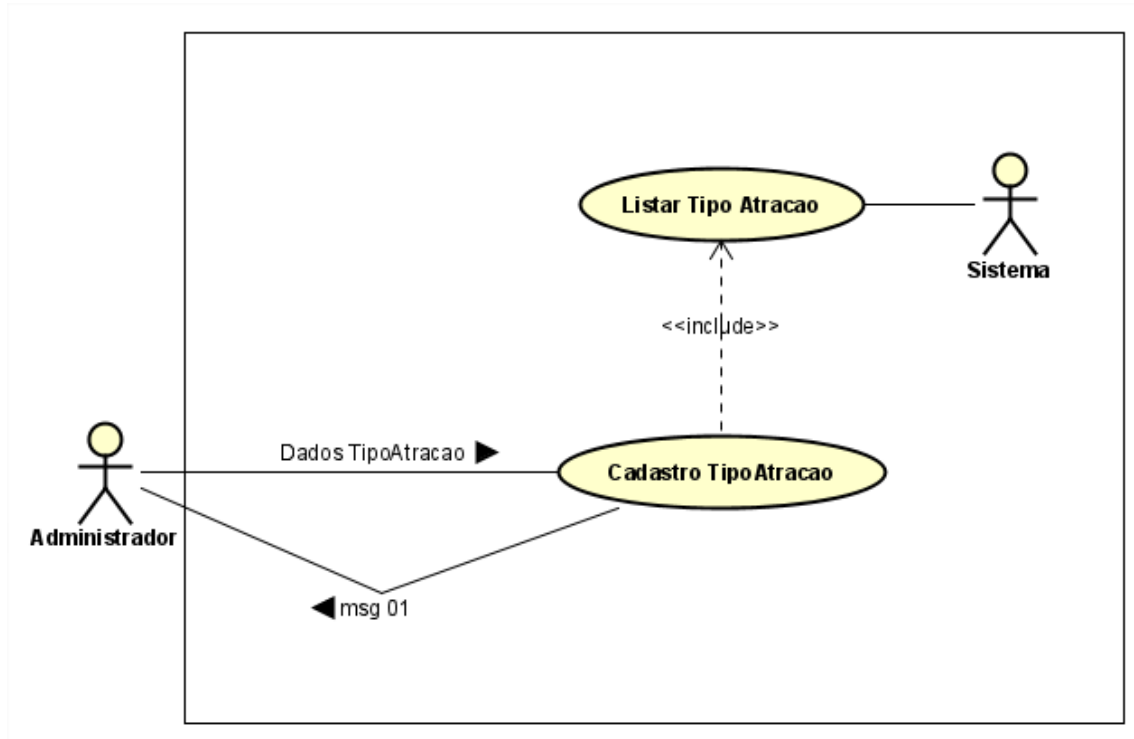
Fonte: Elaborado pelos autores.

Quadro 17 — Documentação do caso de uso Consultar Tipo Turismo

Documentação	
Nome do caso de uso	Consultar Tipo Turismo
Ator principal	Usuário (Administrador)
Resumo	Esse caso de uso descreve as ações necessárias para que o usuário (Administrador) consulta informações referentes aos tipos turismo cadastrados.
Pré-condições	Tipos turismo cadastradas
Pós-condições	O usuário é direcionado ao módulo correspondente do sistema
Fluxo normal	
Ações do ator	Ações do sistema
1. Usuário solicita tipo turismo cadastrados	
	2. Sistema recebe solicitação e mostra tela com os tipos turismo
3. Usuário seleciona tipo turismo que deseja ter acesso às informações.	
	4. Sistema valida solicitação
	5. Sistema direciona usuário ao módulo correspondente de operações
6. Usuário seleciona botão “Consultar”	
	7. Sistema carrega tela de consulta.
Fluxo alternativo	
	5. Sistema lista solicitação
	5.1. Sistema falha ao conectar-se com o banco de dados
	5.1.1. Sistema envia mensagem “Erro de conexão com o banco de dados”
	5.1.2. Sistema retorna ao fluxo normal

Fonte: Elaborado pelos autores.

Figura 15 — Diagrama de casos de uso específico – Cadastrar Tipo Atração



Fonte: Elaborado pelos autores.

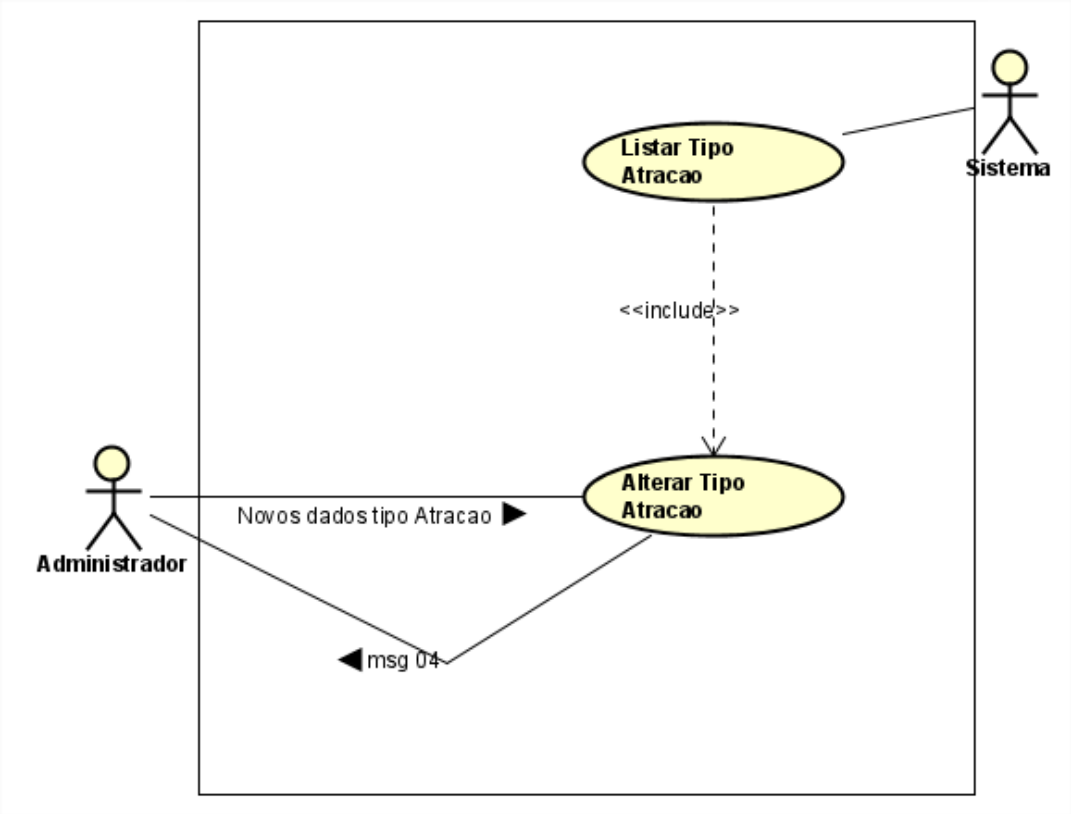
Quadro 18 — Documentação do caso de uso Cadastrar Tipo Atração

Documentação	
Nome do caso de uso	Cadastrar Tipo Atração
Ator principal	Usuário (Administrador)
Resumo	Esse caso de uso descreve as ações necessárias para que o usuário (Administrador) adicione novos tipos atração
Pré-condições	Preencher todos os campos do formulário de inserção
Pós-condições	O usuário é direcionado ao módulo correspondente do sistema

Fluxo normal	
Ações do ator	Ações do sistema
6. Usuário solicita a inserção de um novo tipo atração	
	7. Sistema recebe solicitação e mostra tela com formulário de inserção
8. Usuário informa dados do tipo atração ao formulário e clica em “Salvar”	
	9. Sistema valida campos e dados adicionados
	10. Sistema retorna mensagem “Tipo Atração adicionado!”
Fluxo alternativo	
	6. Sistema valida campos e dados
	6.1. Sistema identifica inconsistência de dados informados ou campos em branco
	6.1.1. Sistema envia mensagem “Preencha todos os campos”
	6.1.2. Sistema retorna ao fluxo normal pelo item 2

Fonte: Elaborado pelos autores.

Figura 16 — Diagrama de casos de uso específico – Alterar Tipo Atração



Fonte: Elaborado pelos autores.

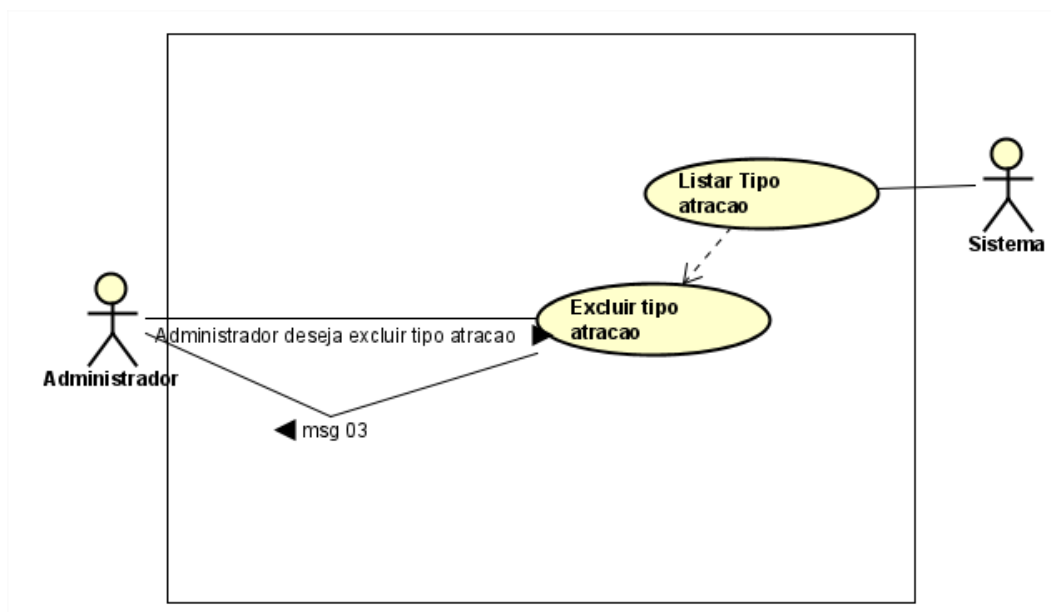
Quadro 19 — Documentação do caso de uso Alterar Tipo Atração

Documentação	
Nome do caso de uso	Alterar Tipo atração
Ator principal	Usuário (Administrador)
Resumo	Esse caso de uso descreve as ações necessárias para que o usuário (Administrador) altere informações referentes aos tipos atração adicionadas anteriormente.
Pré-condições	Tipo atração cadastrado
Pós-condições	O usuário é direcionado ao módulo correspondente do sistema
Fluxo normal	

Ações do ator	Ações do sistema
7. Usuário solicita a alteração de um tipo atração.	
	8. Sistema recebe solicitação e mostra tela com os dados do tipo atração.
9. Usuário altera os dados do tipo atração e seleciona o botão “Alterar”	
	10. Sistema valida solicitação e altera dados.
	11. Sistema envia mensagem “Alteração realizada com sucesso”
	12. Sistema retorna para a tela de listagem.
Fluxo alternativo	
	5. Sistema valida solicitação e altera dados.
	5.1. Sistema falha ao conectar-se com o banco de dados
	5.2. Sistema envia mensagem “Erro de conexão com o banco de dados”
	5.3. Sistema retorna ao fluxo normal pelo item 2

Fonte: Elaborado pelos autores.

Figura 17 — Diagrama de casos de uso específico – Excluir Tipo Atração



Fonte: Elaborado pelos autores.

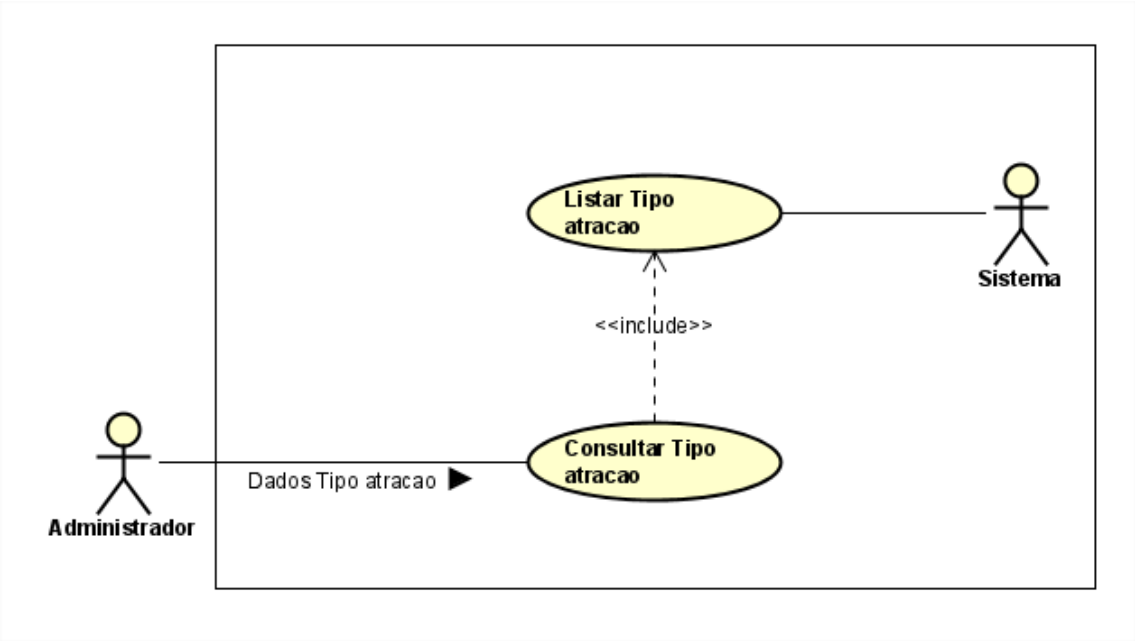
Quadro 20 — Documentação do caso de uso Excluir Tipo Atração

Documentação	
Nome do caso de uso	Excluir Tipo atração
Ator principal	Usuário (Administrador)
Resumo	Esse caso de uso descreve as ações necessárias para que o usuário (Administrador) apague informações referentes aos tipos atração cadastrados.
Pré-condições	Tipos atração cadastrados
Pós-condições	O usuário é direcionado ao módulo correspondente do sistema
Fluxo normal	
Ações do ator	Ações do sistema
11. Usuário solicita exclusão dos tipos atração cadastrados	
	12. Sistema recebe solicitação e mostra tela com os tipos atração
13. Usuário seleciona tipo atração que deseja ter acesso às informações.	
	14. Sistema valida solicitação
	15. Sistema direciona usuário ao módulo correspondente de operações
8. Usuário seleciona botão “Excluir”	
	9. Sistema carrega tela de exclusão.
Fluxo alternativo	
	7. Sistema lista solicitação
	7.1. Sistema falha ao conectar-se com o banco de dados
	7.1.1. Sistema envia mensagem “Erro de conexão com o banco de dados”

	7.1.2.Sistema retorna ao fluxo normal
--	---------------------------------------

Fonte: Elaborado pelos autores.

Figura 18 — Diagrama de casos de uso específico – Consultar Tipo Atração



Fonte: Elaborado pelos autores.

Quadro 21 — Documentação do caso de uso Consultar Tipo Atração

Documentação	
Nome do caso de uso	Consultar Tipo Atração
Ator principal	Usuário (Administrador)
Resumo	Esse caso de uso descreve as ações necessárias para que o usuário (Administrador) consulta informações referentes aos tipos atração cadastrados.
Pré-condições	Tipos atração cadastradas
Pós-condições	O usuário é direcionado ao módulo correspondente do sistema
Fluxo normal	

Ações do ator	Ações do sistema
6. Usuário solicita tipo atração cadastrados	
	7. Sistema recebe solicitação e mostra tela com os tipos atração
8. Usuário seleciona tipo atração que deseja ter acesso às informações.	
	9. Sistema valida solicitação
	10. Sistema direciona usuário ao módulo correspondente de operações
8. Usuário seleciona botão “Consultar”	
	9. Sistema carrega tela de consulta.
Fluxo alternativo	
	8. Sistema lista solicitação
	8.1. Sistema falha ao conectar-se com o banco de dados
	8.1.1. Sistema envia mensagem “Erro de conexão com o banco de dados”
	8.1.2. Sistema retorna ao fluxo normal

Fonte: Elaborado pelos autores.

3.5.1 – Caso de uso: Cadastrar Atendimento

Figura 02 — Diagrama de Contexto – Visão do Paciente



Fonte: Silva, 2016.

Fluxo Normal

- 1-AtorAdm solicita exclusão do registro da pessoa
- 2-Sistema envia mensagem 06: "Deseja excluir o registro?"
- 3-AtorAdm confirma exclusão do registro
- 4-Sistema exclui registro
- 5-Sistema envia msg07: "Registro excluído com sucesso!"
- 6-Sistema atualiza banco de dados e retorna ao Caso de Uso "Alterar Pessoa"

Fluxo Exceção

- 2-Sistema envia mensagem 06: "Deseja excluir o registro?"
- 2.1-AtorAdm não confirma exclusão do registro
- 2.2-Sistema cancela ação
- 2.3-Sistema retorna ao caso de Uso "Alterar Pessoa"

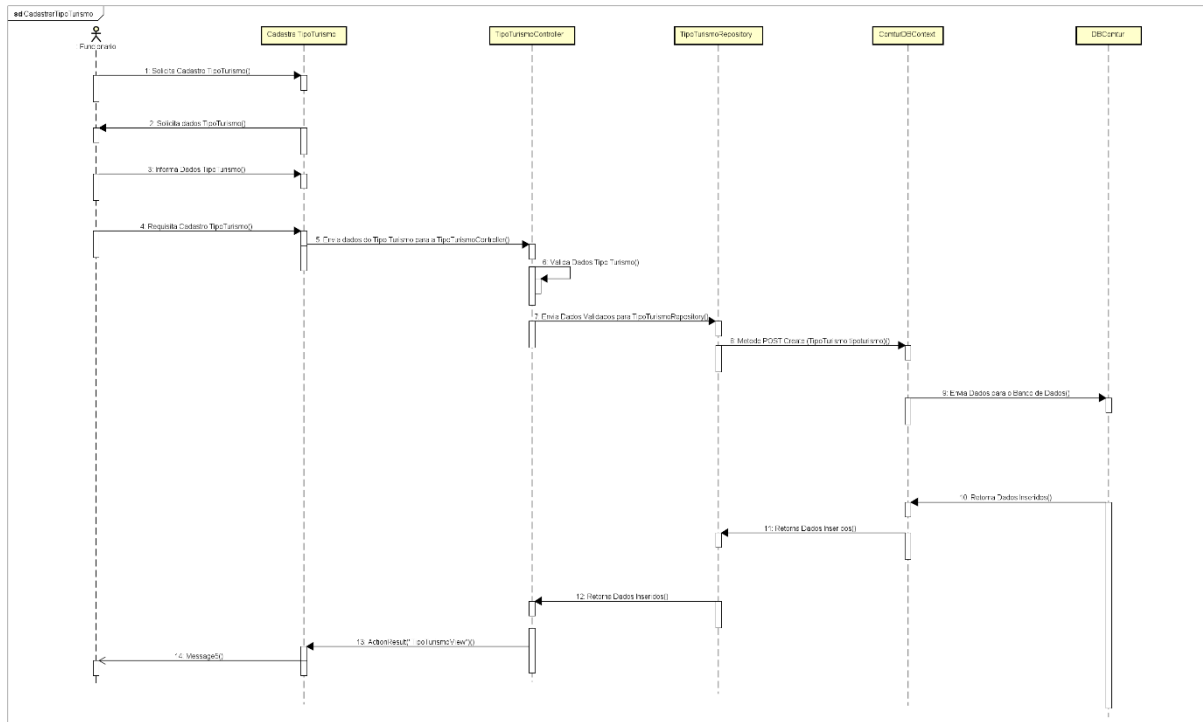
Fluxo Alternativo

- 4-Sistema exclui registro
- 4.1-Sistema identifica erros de conexão com banco de dados ao excluir registro
- 4.2-Sistema envia msg07:"Erros de conexão de BD"
- 4.3-Sistema retorna ao item 1

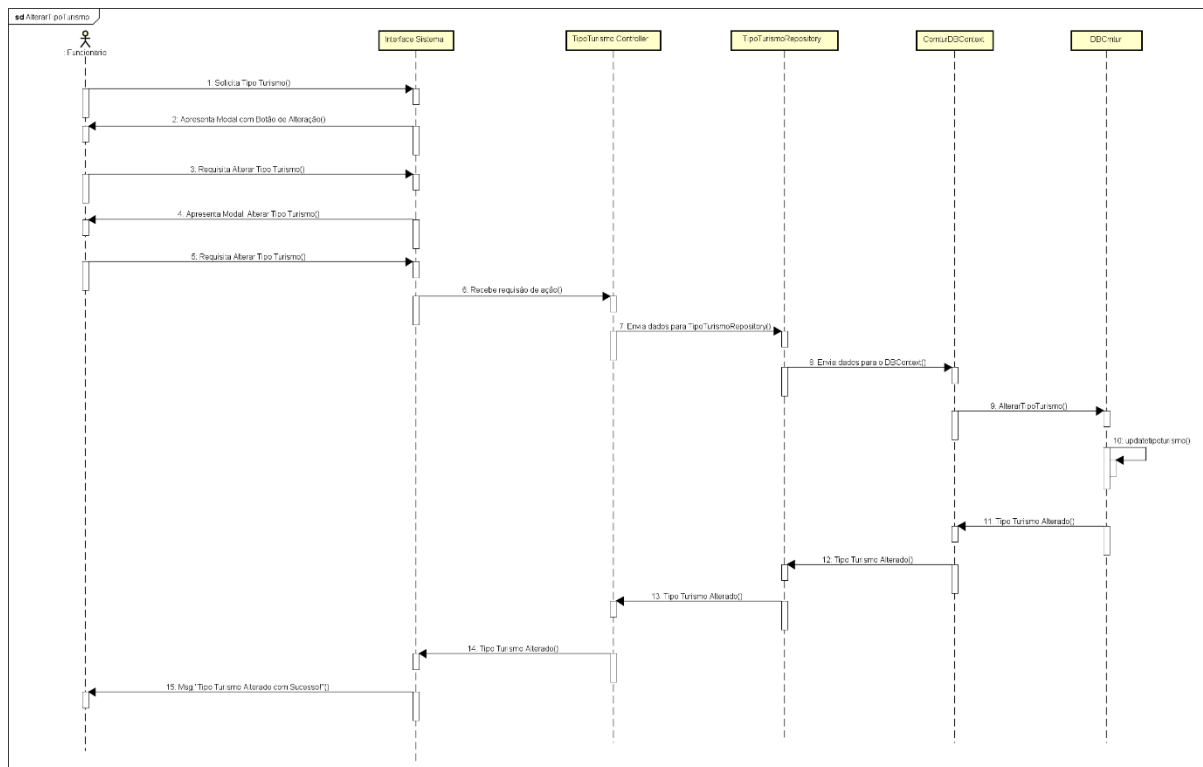
3.6. DIAGRAMA DE SEQUÊNCIA

O diagrama de sequência adota um modelo que representa o comportamento do usuário ao realizar ações dentro da aplicação, atentando para a ordem temporal das ações no desenvolvimento do processo. Este diagrama visa identificar o evento gerador do processo, o ator responsável por esse evento e a maneira como o desenvolvimento do mesmo deve ocorrer. Em suma, ele oferece uma representação visual que destaca a interação entre diferentes elementos da aplicação, facilitando a compreensão da lógica temporal e das responsabilidades associadas às ações realizadas. (GUEDES, 2018).

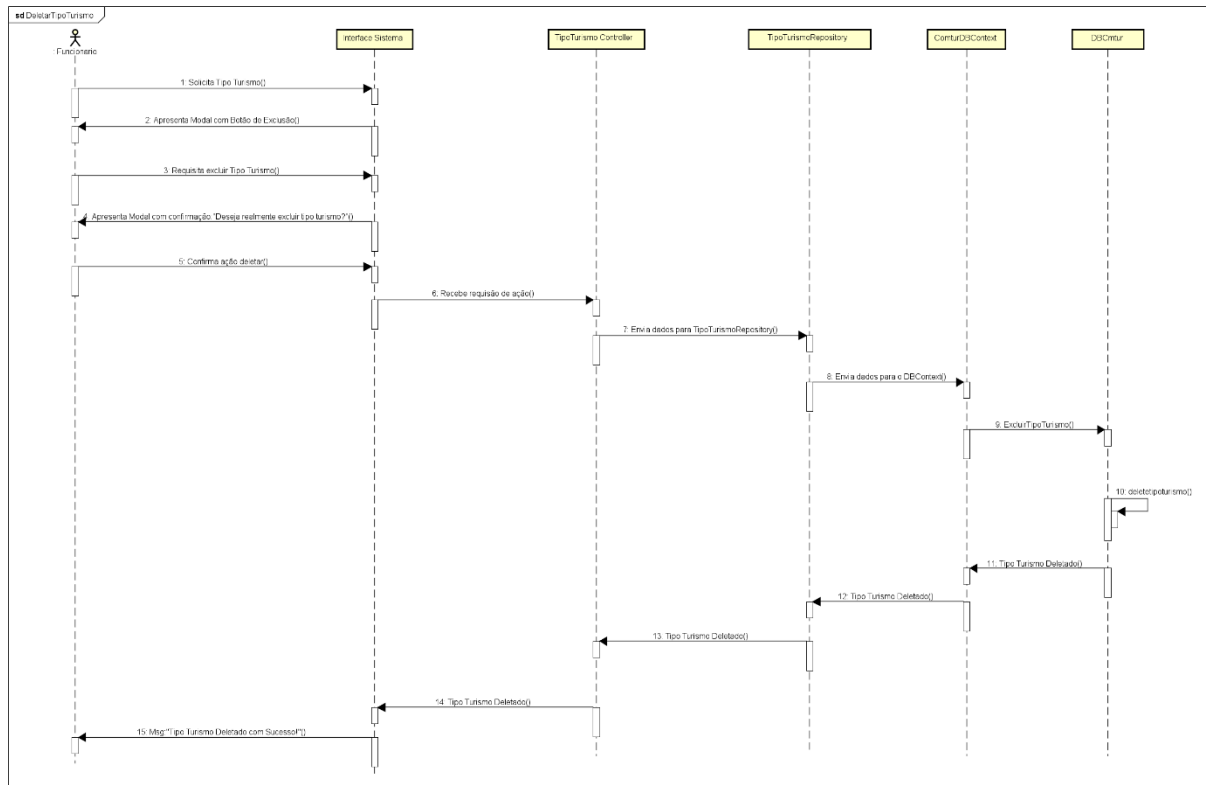
Nesta subseção, apresentaremos o diagrama de sequência para tipo turismo e notícia, abrangendo a sequência até a conclusão da ação.

Figura 19 — Diagrama de Sequência – Cadastrar Tipo Turismo

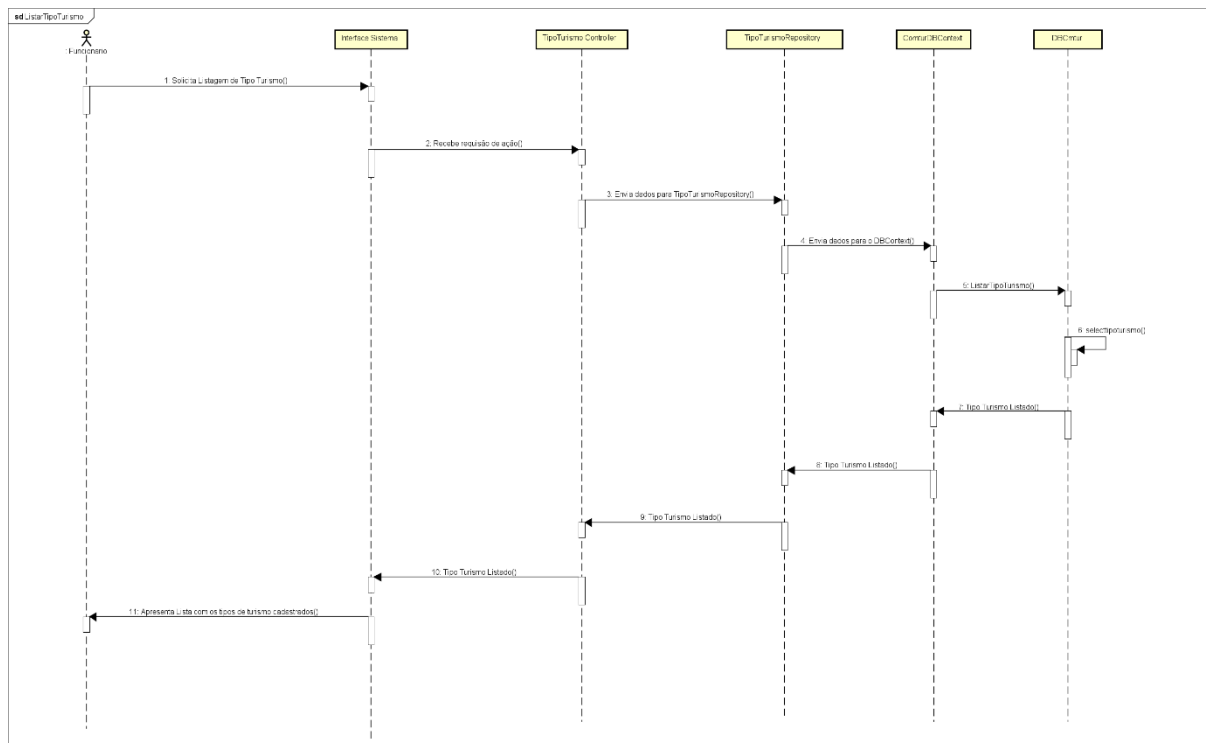
Fonte: Desenvolvidos pelos autores.

Figura 20 — Diagrama de Sequência – Alterar Tipo Turismo

Fonte: Desenvolvidos pelos autores.

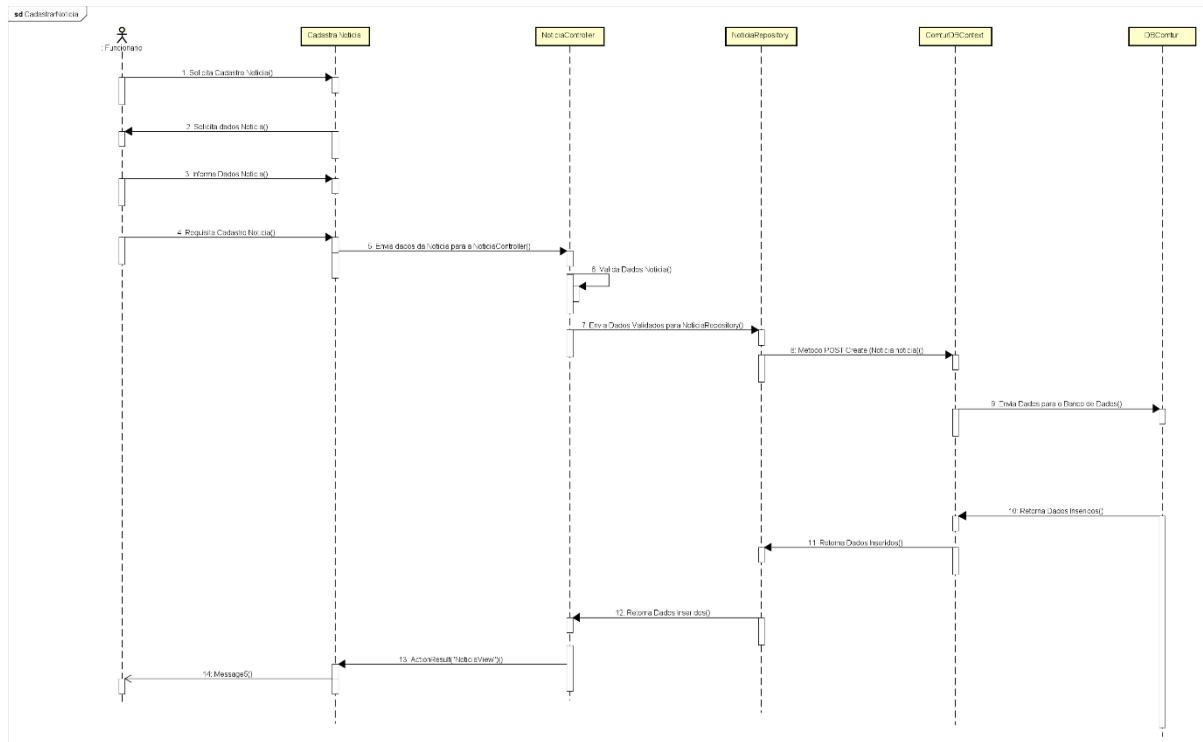
Figura 21 — Diagrama de Sequência – Deletar Tipo Turismo

Fonte: Desenvolvidos pelos autores.

Figura 22 — Diagrama de Sequência – Listar Tipo Turismo

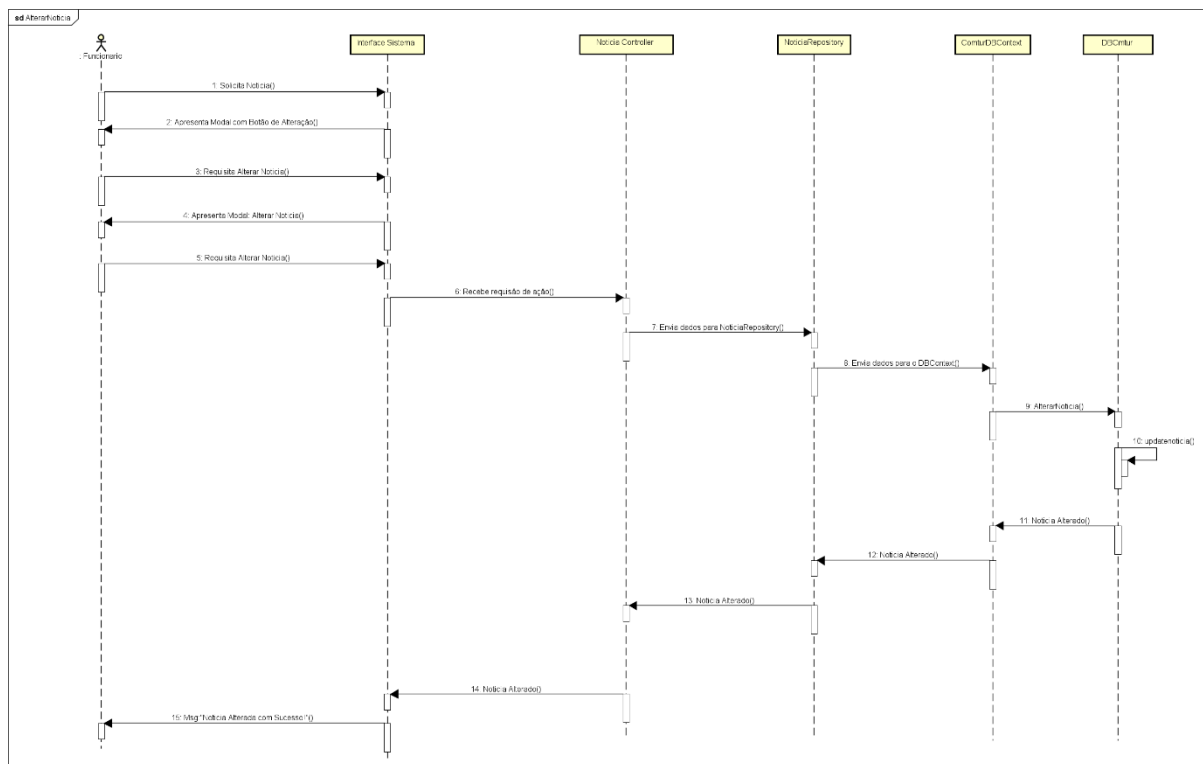
Fonte: Desenvolvidos pelos autores.

Figura 23 — Diagrama de Sequência – Cadastrar Notícia



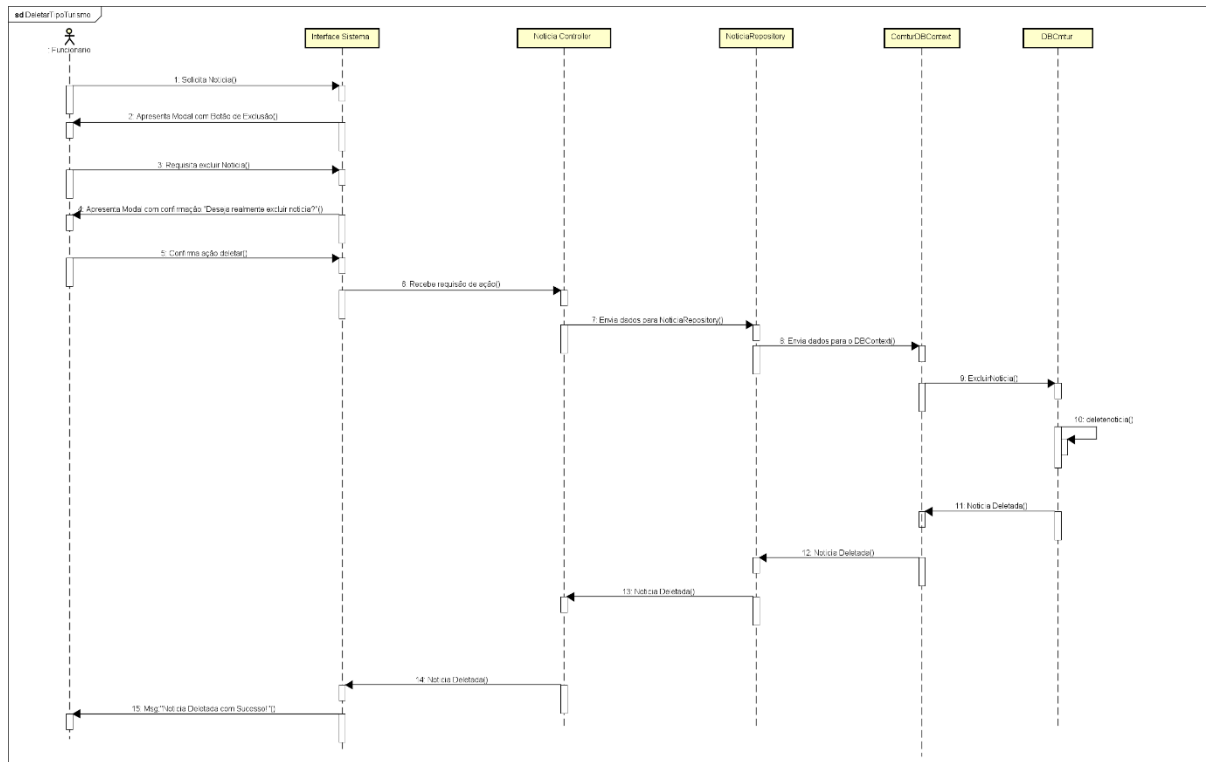
Fonte: Desenvolvidos pelos autores.

Figura 24 — Diagrama de Sequência – Alterar Notícia



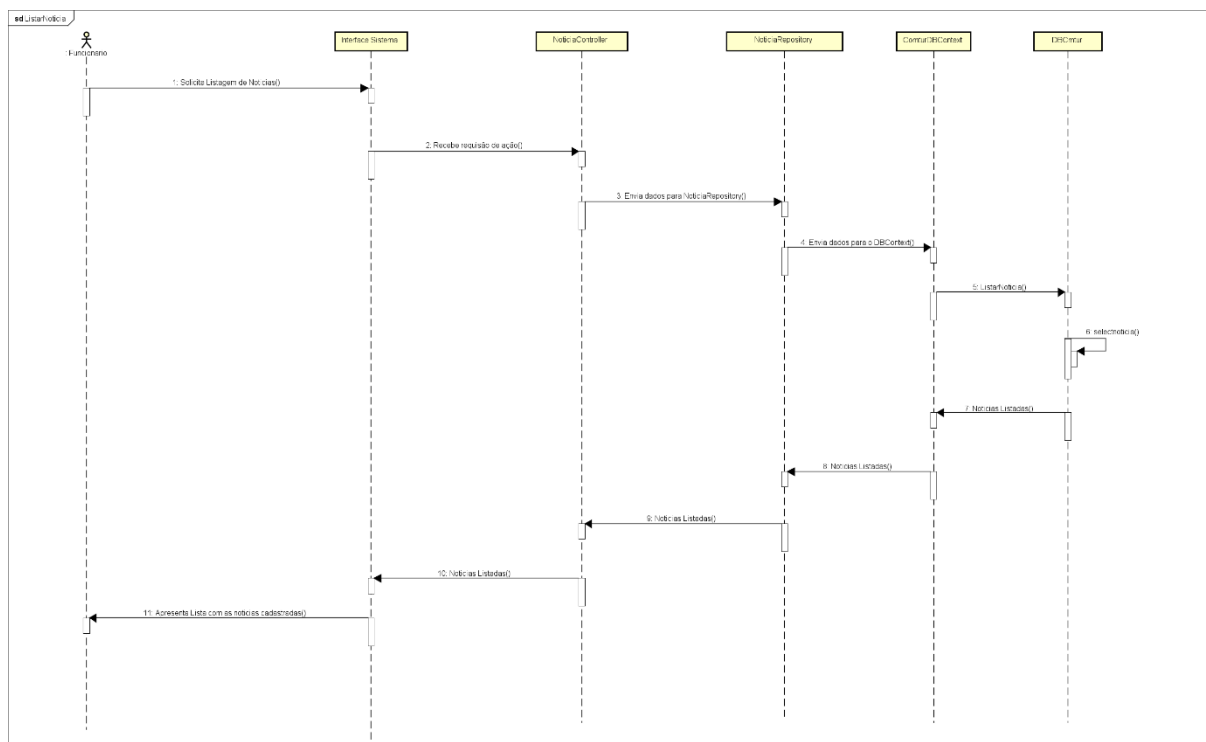
Fonte: Desenvolvidos pelos autores.

Figura 25 — Diagrama de Sequência – Deletar Notícia



Fonte: Desenvolvidos pelos autores.

Figura 26 — Diagrama de Sequência – Listar Noticia




Fonte: Desenvolvidos pelos autores.

4 DEFINIÇÃO DA INTERFACE COM O USUÁRIO (UX) (3º SEMESTRE)

4.1 DESCRIÇÃO DE CENÁRIO

Conforme destacado em "O Guia para Projetar UX" de Russ Unger e Carolyn Chandler, os cenários são narrativas detalhadas que capturam as interações entre um usuário e um sistema. Essas histórias fornecem uma compreensão abrangente ao abordar o contexto no qual a persona está inserida, os objetivos que ela busca alcançar, as ações específicas que realiza para atingir esses objetivos, os desafios que pode encontrar no processo e, por fim, os resultados dessas interações. Os cenários de persona é uma das ferramentas fundamentais para orientar o design de experiência do usuário, permitindo uma avaliação detalhada da eficácia do sistema sob diversas circunstâncias.

Figura 27 — Imagem do primeiro cenário

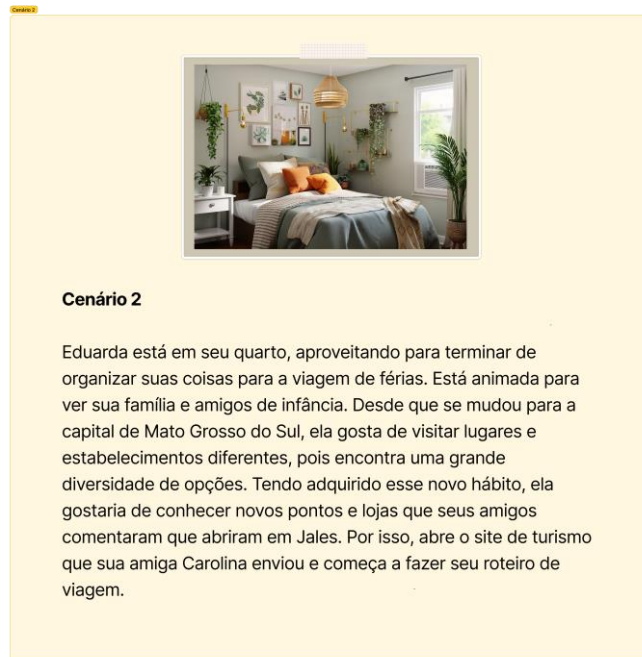


Cenário 1

Luiz está na sua loja em uma tarde ensolarada de quinta-feira. O movimento está tranquilo e, como há quase nenhum cliente, os funcionários estão organizando o estoque. Aproveitando a calma, ele entra no Instagram do estabelecimento. A quantidade de seguidores é a mesma de dois meses atrás, e as interações nas fotos e stories são escassas, algo que o preocupa há dias. Rolando o feed, Luiz vê a nova publicação de um evento da prefeitura, e a legenda falando sobre o site de turismo de Jales o interessa. Ele acessa o site e vê o destaque sobre divulgar seu negócio.

Fonte: Elaborado pelos autores.

Figura 28 — Imagem do segundo cenário



Fonte: Elaborado pelos autores.

4.2 DESCRIÇÃO DE PERSONAS

De acordo com "O Guia para Projetar UX" de Russ Unger e Carolyn Chandler, personas são representações fictícias de usuários que interagem com um sistema específico. Elas são desenvolvidas com foco no usuário para garantir uma compreensão aprofundada das necessidades, comportamentos e motivações dos usuários. Ao criar personas, os designers e desenvolvedores buscam uma visão mais humanizada e centrada no usuário, permitindo a criação de soluções que atendam de maneira mais eficaz às expectativas e requisitos dos usuários finais.

Figura 29 — Imagem da primeira persona

Persona 1



Luiz Antonio

"Empresário"

- Idade: 45 anos.
- Gênero: Masculino.
- Profissão: Empresário.
- Localidade(cidade): Jales.
- Familiaridade com a tecnologia (0 a 10): 6.

Bio

Luiz Antonio abriu uma loja de sapatos em 2019 e gerencia ela com sua esposa. Atualmente ele tem 5 funcionários entre 25-40 anos.

Desafios e Objetivos:

Com a pandemia, Luiz viu a necessidade de criar uma página no Instagram para a loja, visto que esse tipo de divulgação se popularizou entre os estabelecimentos no geral. Porém o acesso e o crescimento de seguidores continua sendo uma grande barreira. Tendo esse empecilho, ele gostaria de encontrar uma maneira de melhorar a divulgação da sua loja, atraindo um público mais amplo.

Fonte: Elaborado pelos autores.

Figura 30 — Imagem da segunda persona



Fonte: Elaborado pelos autores.

4.3 ESBOÇOS DE TELA (WIREFRAMES)

Os wireframes são representações visuais esquemáticas e simplificadas de interfaces de usuário, desempenhando um papel crucial no processo de design de experiência do usuário. Como descrito por Jakob Nielsen, um renomado especialista em usabilidade, "wireframes são esboços visuais que representam a estrutura e o layout básico de uma página da web ou aplicativo". Essas representações gráficas, muitas vezes em preto e branco e com detalhes mínimos, servem como um guia inicial para a disposição de elementos na interface, proporcionando uma visão clara e simplificada da estrutura do design.

Ao criar wireframes, os designers visuais buscam estabelecer a hierarquia visual, distribuição de conteúdo e interações primárias, conforme sugerido por Don Norman, autor de "The Design of Everyday Things". Os wireframes permitem que a equipe de design e desenvolvimento colabore eficazmente na definição de layouts e fluxos de usuário, antes de se comprometerem com detalhes visuais mais refinados.

Os benefícios dos wireframes se estendem à economia de tempo e recursos, proporcionando uma plataforma para testar conceitos e realizar iterações antes da

implementação final. Como mencionado por Steve Krug, autor de "Don't Make Me Think", "os wireframes são uma ferramenta valiosa para garantir que todos na equipe estejam alinhados quanto à estrutura e à funcionalidade antes de investir tempo em detalhes gráficos".

Figura 31 — Imagem do Wireframe da Tela de Login para o Administrador

← Voltar

O wireframe da tela de login para o administrador é dividido em duas seções principais. À esquerda, há um formulário contendo o texto "Bem-vindo de volta!" e "Efetuar Login". Abaixo disso, há três campos de entrada rotulados "Campo 1", "Campo 2" e "Campo 3". O campo "Campo 3" possui um ícone de olho para alternar a visibilidade da senha. Abaixo dos campos, há um botão "Login" e, no rodapé, o texto "Ainda não tem uma conta? [Cadastre-se](#)". À direita do formulário, há um espaço reservado para a logo, indicado pelo texto "LOGO" em um retângulo cinza.

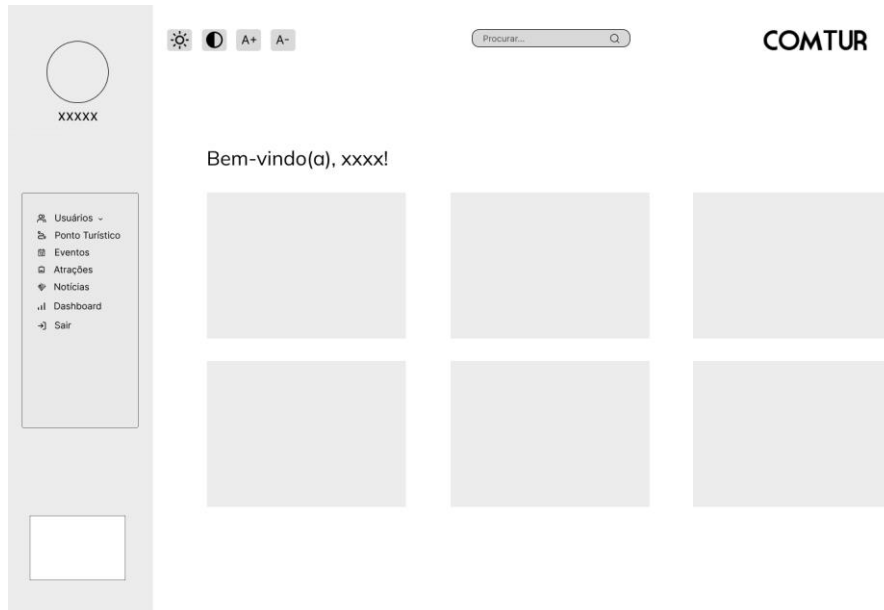
Fonte: Elaborado pelos autores

O wireframe de login para o Administrador no Comtur exibido acima apresenta uma disposição simplificada e estruturada para garantir uma experiência de usuário eficiente. Na parte superior da página, são claramente identificados os campos essenciais, como "Nome", "E-mail" e "Senha", refletindo a abordagem minimalista típica de Wire frames. Esses campos estão dispostos de forma lógica e acessível, facilitando a inserção rápida e precisa de informações.

Além dos campos tradicionais de login, o wireframe inclui uma opção de "Cadastrar-se", estrategicamente posicionada para oferecer uma alternativa imediata a usuários que ainda não possuem uma conta. Ao clicar nessa opção, seria esperado que o usuário seja redirecionado para uma página de registro, simplificando o fluxo para novos administradores.

Essa abordagem, inspirada nas melhores práticas de usabilidade, segue os princípios de Jakob Nielsen sobre a clareza na apresentação de informações importantes em interfaces de usuário. O wireframe, por sua natureza esquemática, destaca a funcionalidade central de login, facilitando a compreensão e a interação intuitiva por parte do administrador do Comtur.

Figura 32 — Imagem do Wire frame da Tela Inicial do Administrador



Fonte: Elaborado pelos autores.

O wireframe da tela inicial para o administrador no Comtur é estruturado para garantir uma navegação intuitiva e eficiente. Inclui um Navbar na parte superior para acesso rápido às principais seções, um Sidebar à esquerda para uma navegação mais detalhada e um botão "Logout" para encerrar sessões facilmente. Seis cards no corpo principal oferecem atalhos visuais para informações e funcionalidades importantes. A disposição simétrica e a clareza visual destacam a preocupação com a usabilidade e a experiência do usuário.

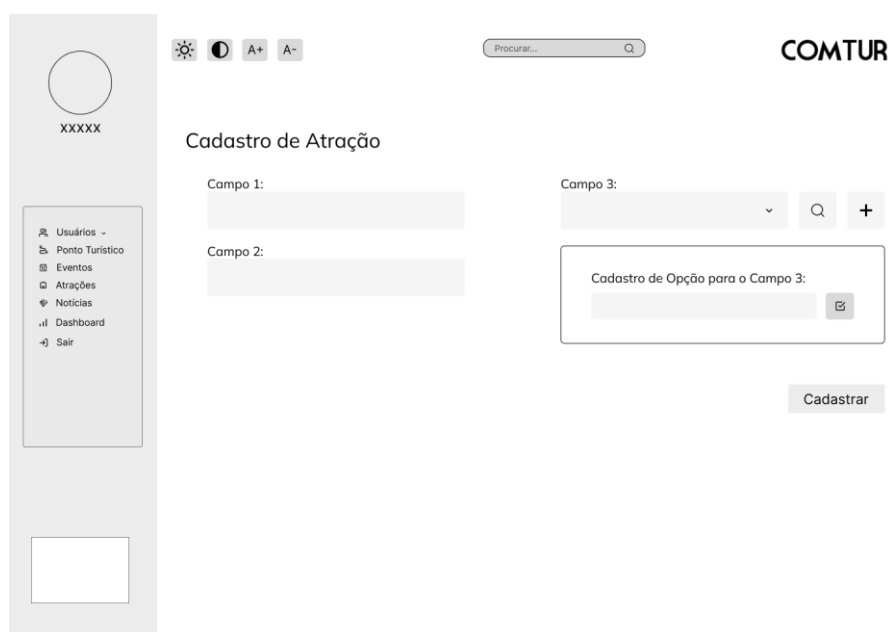
Figura 33 — Imagem do Wireframe da Tela de Cadastro de Notícia



Fonte: Elaborado pelas autoras.

O wireframe para a tela de cadastro de notícias como o da figura acima, no Comtur mantém a consistência com o Navbar e Sidebar, proporcionando uma navegação intuitiva. O Navbar superior continua oferecendo acesso rápido a diferentes seções, enquanto o Sidebar à esquerda organiza links específicos para o cadastro de notícias. O corpo principal da tela destaca áreas para preenchimento dos detalhes da notícia, mantendo um design claro e simplificado para facilitar a entrada de dados pelo administrador. Essa abordagem assegura uma experiência coesa e eficiente durante o processo de cadastro.

Figura 34 — Imagem do Wireframe da Tela de Cadastro de Atrações



Fonte: Elaborada pelas autoras

O wireframe para a tela de cadastro de atrações no Comtur segue a mesma abordagem intuitiva e funcional, mantendo consistência com o Navbar e Sidebar para otimizar a navegação. O Navbar superior continua a proporcionar acesso rápido a diferentes seções do sistema, assegurando uma experiência coesa para o administrador.

Na parte esquerda da tela, o Sidebar permanece como uma ferramenta de navegação prática, expandindo-se para incluir links específicos relacionados ao cadastro de atrações. Essa organização lógica e acessibilidade aprimorada facilita a interação durante o processo de inserção de informações.

O corpo principal da tela destaca áreas de preenchimento para os detalhes da atração, como nome, descrição e localização. A disposição clara e simplificada dos campos visa facilitar a compreensão e inserção de dados por parte do administrador.

Essa continuidade de design entre o Navbar, Sidebar e o layout central proporciona uma experiência de usuário coesa e familiar, reforçando a usabilidade e eficiência durante o cadastro de atrações no Comtur.

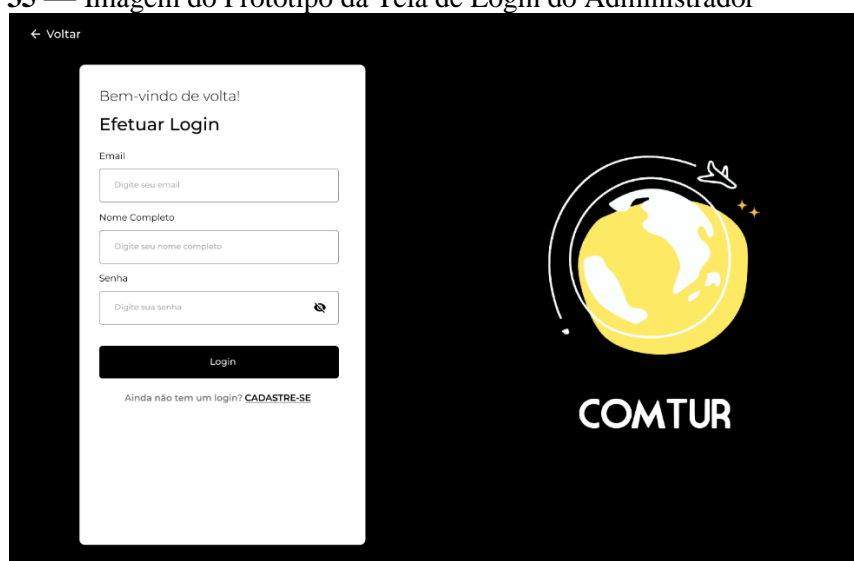
Em síntese, os wireframes desenvolvidos para as telas de cadastro de notícias e atrações no Comtur refletem uma abordagem consistente e intuitiva, mantendo a usabilidade como prioridade. A estrutura organizada, aliada ao Navbar e Sidebar, proporciona uma experiência de navegação clara e eficiente para os administradores. A continuidade no design, seja durante o cadastro de notícias ou atrações, contribui para a familiaridade do usuário, simplificando os processos de entrada de dados. Assim, os Wire frames reforçam a importância da consistência e usabilidade no desenvolvimento de interfaces, promovendo uma experiência de usuário coesa e eficaz no Comtur.

4.4 PROTÓTIPOS DE TELA

Os protótipos de tela, conforme destacado por Bill Buxton, um renomado pesquisador em interação humano-computador, são "representações tangíveis e interativas da proposta de design, permitindo a exploração, comunicação e refinamento de ideias". No contexto do Comtur, essas representações desempenham um papel essencial ao fornecer uma visão visual antecipada das interfaces planejadas. Como afirma Don Norman, autor de "The Design of Everyday Things", "protótipos são uma ferramenta fundamental para explorar possíveis soluções de design antes da implementação completa".

A importância dos protótipos no Comtur transcende a mera visualização, pois, como ressaltado por Jakob Nielsen, "a iteração rápida e o feedback contínuo dos protótipos são cruciais para a criação de uma interface de usuário eficaz". Dessa forma, os protótipos de tela não apenas facilitam a comunicação entre as equipes de design e desenvolvimento, mas também desempenham um papel fundamental na identificação precoce de ajustes necessários, contribuindo para economia de tempo e recursos. No âmbito do Comtur, os protótipos de tela emergem como ferramentas indispensáveis para a validação de conceitos, o refinamento de funcionalidades e a garantia de uma experiência de usuário otimizada desde as fases iniciais do desenvolvimento.

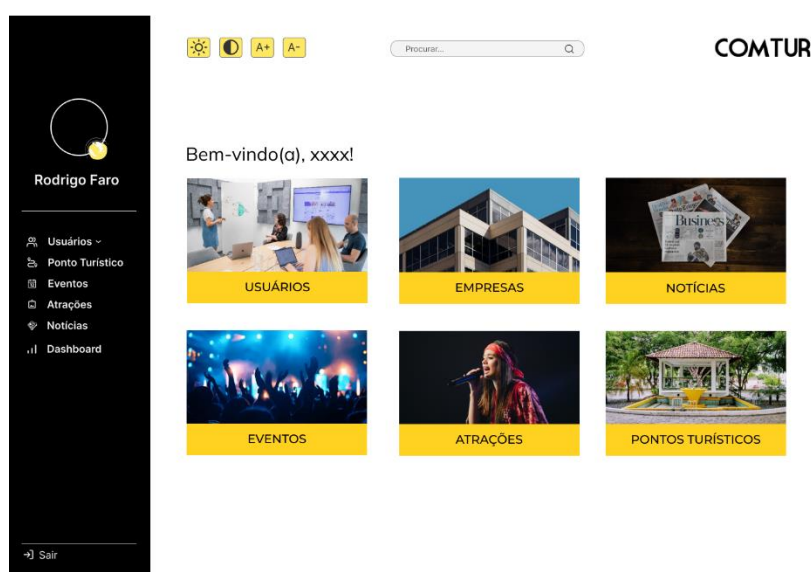
Figura 35 — Imagem do Protótipo da Tela de Login do Administrador



Fonte: Elaborada pelas autoras.

O protótipo de tela para o Login de Administrador no Comtur é concebido para proporcionar uma representação visual e interativa do processo autenticação. Inspirado pelas diretrizes de design de Don Norman, o protótipo busca oferecer uma experiência de usuário eficaz e intuitiva. O layout destaca claramente os campos essenciais, como nome de usuário e senha, seguindo uma abordagem minimalista e de fácil compreensão. A inclusão de elementos como botões de "Entrar" e "Esqueceu a Senha" contribui para uma experiência de login acessível e orientada pelo usuário, refletindo o compromisso do Comtur com a usabilidade.

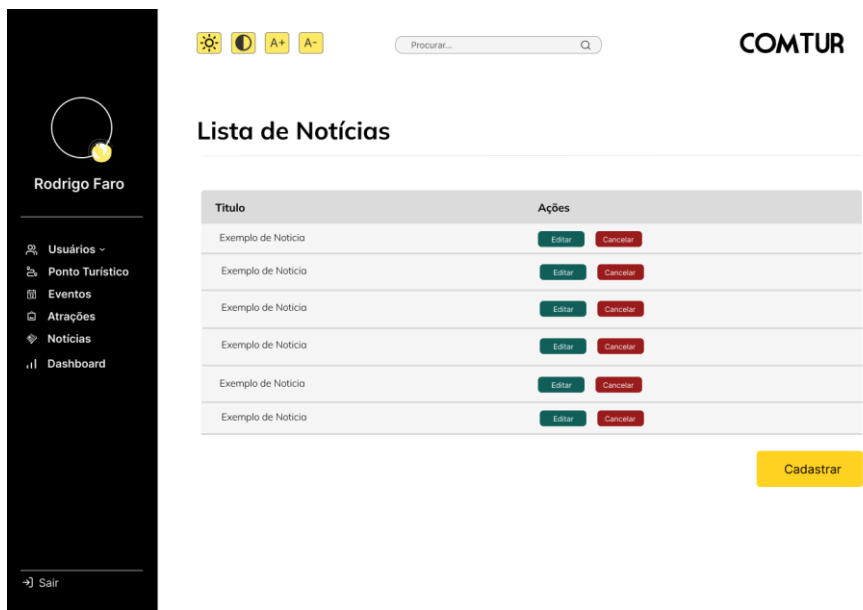
Figura 36 — Imagem do Prototipo da Tela Inicial do Administrador



Fonte: Elaborada pelas autoras.

O protótipo de tela para a Tela Inicial do Administrador no Comtur visa oferecer uma visão consolidada e funcional. Inspirado por conceitos de design de Ralph Johnson, a tela destaca um Navbar superior para navegação rápida entre seções e um Sidebar à esquerda para acesso direto a funcionalidades específicas. Seis cards no corpo principal oferecem atalhos visuais para áreas importantes. Essa abordagem proporciona uma experiência coesa, refletindo a prioridade do Comtur em oferecer uma interface administrativa eficiente e intuitiva.

Figura 37 — Imagem do Protótipo da Tela de Cadastro de Notícias



Fonte: Elaborada pelas autoras.

O protótipo de tela para o Cadastro de Notícia no Comtur visa simplificar o processo de inserção de informações relevantes. Inspirado por Steve Krug, o design destaca áreas claras para título, conteúdo e data de publicação, facilitando a compreensão e entrada de dados. Botões de ação, como "Salvar" e "Cancelar", são estrategicamente posicionados para garantir uma interação eficiente. Essa abordagem busca otimizar a usabilidade durante o cadastro de notícias, proporcionando uma experiência intuitiva e centrada no usuário.

Figura 38 — Imagem do Protótipo da Tela de Cadastro de Atrações

COMTUR

Procurar...

Cadastro

Atração

*Nome:

*Tipo:

*Descrição:

Tipo Atração:

Rodrigo Faro

- Usuários
- Ponto Turístico
- Eventos
- Atrações
- Notícias
- Dashboard

Sair

Fonte: Elaborada pelas autoras.

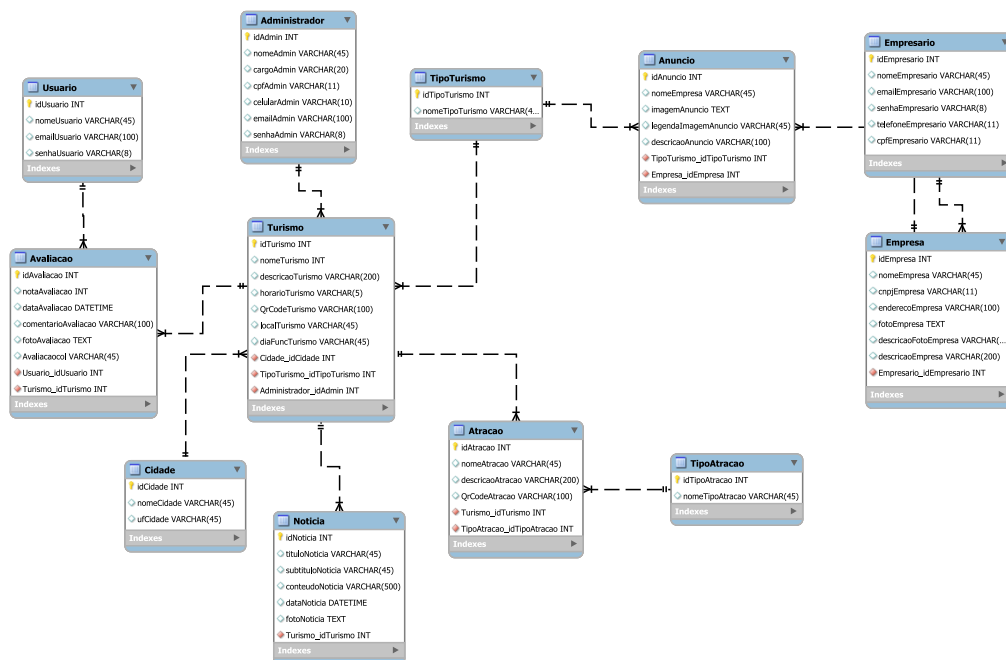
O protótipo de tela para o Cadastro de Atrações acima, no Comtur se configura como uma representação visual e interativa destinada a ilustrar a futura interface de usuário dedicada a esse processo. Inspirado pela abordagem de Bill Buxton, pesquisador em interação humano-computador, o protótipo busca proporcionar uma experiência antecipada e tangível do design proposto. Este protótipo apresenta áreas de preenchimento intuitivas, como nome, descrição e localização, visando uma fácil compreensão e inserção de dados pelos administradores do Comtur. A iteração rápida e o feedback contínuo provenientes desse protótipo, seguindo os princípios destacados por Jakob Nielsen, são cruciais para refinar a interface, garantindo uma experiência de usuário eficaz no momento do cadastro de atrações. Este processo iterativo e visual, alinhado às práticas de design centrado no usuário, contribui para a identificação precoce de melhorias necessárias, promovendo a eficiência no desenvolvimento do Comtur.

5 BANCO DE DADOS

5.1 MODELO ENTIDADE RELACIONAMENTO

O Modelo Entidade-Relacionamento (MER), segundo o livro “Banco de dados: projeto e implementação” oferece uma representação visual das entidades (representação de objetos do mundo real), atributos (características de uma entidade) e relacionamentos (conexão entre diferentes entidades) que constituem o sistema de armazenamento de dados. É uma ferramenta que auxilia na captura de relações complexas e na estruturação dos dados de maneira compreensível.

Figura 39 — Mapeamento do Objeto Relacional



Fonte: Elaborado pelos autores.

5.2 SCRIPT DAS TABELAS

Script das tabelas, conforme abordado no livro “Banco de dados: Projeto e Implementação”, são as instruções ou comandos escritos em uma linguagem de consulta estruturada (SQL) para criar, modificar ou excluir tabelas em um banco de dados. Esses scripts são utilizados para definir a estrutura de um banco de dados relacional, incluindo as tabelas, campo e restrições.

Quadro 22 — Script SQL – Tabela TipoTurismo

```
CREATE TABLE tipoturismo (
    tipoturismoid integer GENERATED BY DEFAULT AS IDENTITY,
    nome character varying(50) NOT NULL,
    CONSTRAINT "PK_tipoturismo" PRIMARY KEY (tipoturismoid)
);
```

Fonte: Elaborado pelos autores.

Quadro 23 — Script SQL – Tabela TipoAtracao

```
CREATE TABLE tipoatracao (
    tipoatracaoid integer GENERATED BY DEFAULT AS IDENTITY,
    nome character varying(50) NOT NULL,
    CONSTRAINT "PK_tipoatracao" PRIMARY KEY (tipoatracaoid)
);
```

Fonte: Elaborado pelos autores.

Quadro 24 — Script SQL – Tabela Administrador

```
CREATE TABLE administrador (
    administradorid integer GENERATED BY DEFAULT AS IDENTITY,
    nome character varying(50) NOT NULL,
    cargoadministrador text NOT NULL,
    cpfadministrador text NOT NULL,
    telefoneadministrador text NOT NULL,
    emailadministrador text NOT NULL,
    senhaadministrador text NOT NULL,
    CONSTRAINT "PK_administrador" PRIMARY KEY (administradorid)
);
```

Fonte: Elaborados pelos autores.

Quadro 25 — Script SQL – Tabela Noticia

```
CREATE TABLE noticia (
    noticiaid integer GENERATED BY DEFAULT AS IDENTITY,
    titulo character varying(50) NOT NULL,
    subtitulo text NOT NULL,
    conteudo text NOT NULL,
    datapublicacao date NOT NULL,
```

```

        "horaPublicacao" text NOT NULL,
        CONSTRAINT "PK_noticia" PRIMARY KEY (noticiaid)
    );

```

Fonte: Elaborado pelos autores.

Quadro 26 — Script SQL – Tabela Empresario

```

CREATE TABLE empresario (
    empresarioid integer GENERATED BY DEFAULT AS IDENTITY,
    nome character varying(50) NOT NULL,
    emailempresario text NOT NULL,
    senhaempresario text NOT NULL,
    telefoneempresario text NOT NULL,
    CONSTRAINT "PK_empresario" PRIMARY KEY (empresarioid)
);

```

Fonte: Elaborado pelos autores.

Quadro 27 — Script SQL – Tabela Usuario

```

CREATE TABLE usuario (
    usuarioid integer GENERATED BY DEFAULT AS IDENTITY,
    nome character varying(50) NOT NULL,
    email character varying(50) NOT NULL,
    senha character varying(50) NOT NULL,
    CONSTRAINT "PK_usuario" PRIMARY KEY (usuarioid)
);

```

Fonte: Elaborado pelos autores.

Quadro 28 — Script SQL – Tabela Cidade

```

CREATE TABLE cidade (
    cidadeid integer GENERATED BY DEFAULT AS IDENTITY,
    nome character varying(50) NOT NULL,
    uf character varying(2) NOT NULL,
    CONSTRAINT "PK_cidade" PRIMARY KEY (cidadeid)
);

```

Fonte: Elaborado pelos autores.

Quadro 29 — Script SQL – Tabela Avaliação

```

CREATE TABLE avaliacao (
    avaliacaoid integer GENERATED BY DEFAULT AS IDENTITY,
    nota character varying(50) NOT NULL,
    dataavaliacao date NOT NULL,
    comentario character varying(100) NOT NULL,
    foto text NOT NULL,
    avaliacao character varying(50) NOT NULL,
    nomeusuario character varying(50) NOT NULL,
    turismoavaliacao character varying(50) NOT NULL,
    CONSTRAINT "PK_avaliacao" PRIMARY KEY (avaliacaoid),
    CONSTRAINT "FK_avaliacao_usuario" FOREIGN KEY ("NomeUsuario")
REFERENCES usuario (usuarioid)
);

```

Fonte: Elaborado pelos autores.

Quadro 30 — Script SQL – Tabela Turismo

```

CREATE TABLE turismo (
    turismoid integer GENERATED BY DEFAULT AS IDENTITY,
    nome character varying(50) NOT NULL,
    descricao character varying(100) NOT NULL,
    horario character varying(5) NOT NULL,
    qrcode character varying(100) NOT NULL,
    localturismo character varying(50) NOT NULL,
    diafuncionamento character varying(10) NOT NULL,
    cidadeturismo character varying(50) NOT NULL,
    tipoturismo character varying(50) NOT NULL,
    administradorturismo character varying(50) NOT NULL,
    CONSTRAINT "PK_turismo" PRIMARY KEY (turismoid),
    CONSTRAINT "FK_cidade_turismo" FOREIGN KEY ("CidadeTurismo")
REFERENCES cidade (cidadeid),
    CONSTRAINT "FK_tipoturismo_turismo" FOREIGN KEY ("TipoTurismo")
REFERENCES tipoturismo (tipoturismoid),
    CONSTRAINT "FK_administrador_turismo" FOREIGN KEY
("Administrador") REFERENCES administrador (administradorid)
);

```

Fonte: Elaborado pelos autores.

Quadro 31 — Script SQL – Tabela Atração

```

CREATE TABLE atracao (
    atracaoid integer GENERATED BY DEFAULT AS IDENTITY,
    nome character varying(50) NOT NULL,
    descricao character varying(100) NOT NULL,
    qrcode character varying(100) NOT NULL,
    turismo character varying(50) NOT NULL,
    tipoatracao character varying(50) NOT NULL,
    CONSTRAINT "PK_atracao" PRIMARY KEY (atracaoid),
    CONSTRAINT "FK_turismo_atracao" FOREIGN KEY ("Turismo")
REFERENCES turismo (turismoid),
    CONSTRAINT "FK_tipoatracao_turismo" FOREIGN KEY ("TipoAtracao")
REFERENCES tipoatracao (tipoatracaoid)
);

```

Fonte: Elaborado pelos autores.

Quadro 32 — Script SQL – Tabela Unúncio

```

CREATE TABLE anuncio (
    anuncioid integer GENERATED BY DEFAULT AS IDENTITY,
    nome character varying(50) NOT NULL,
    imagem text NOT NULL,
    legenda character varying(50) NOT NULL,
    descricao character varying(100) NOT NULL,
    tipoturismo character varying(50) NOT NULL,
    empresa character varying(50) NOT NULL,
    CONSTRAINT "PK_anuncio" PRIMARY KEY (anuncioid),
    CONSTRAINT "FK_tipoturismo_anuncio" FOREIGN KEY ("TipoTurismo")
REFERENCES tipoturismo (tipoturismoid),
    CONSTRAINT "FK_empresa_anuncio" FOREIGN KEY ("Empresa")
REFERENCES empresa (empresaid)
);

```

Fonte: Elaborado pelos autores.

Quadro 33 — Script SQL – Tabela Empresa

```

CREATE TABLE empresa (
    empresaid integer GENERATED BY DEFAULT AS IDENTITY,
    nome character varying(50) NOT NULL,

```

```
cnpjta character varying(50) NOT NULL,  
endereco character varying(100) NOT NULL,  
foto text NOT NULL,  
descricaofoto character varying(50) NOT NULL,  
descricaoempresa character varying(100) NOT NULL,  
empresario character varying(50) NOT NULL,  
CONSTRAINT "PK_empresa" PRIMARY KEY (empresaid),  
CONSTRAINT "FK_empresario_empresa" FOREIGN KEY ("Empresario")  
REFERENCES empresario (empresarioid)  
);
```

Fonte: Elaborado pelos autores.

6 ARQUITETURA DE SOFTWARE

A arquitetura de software, conforme delineada por Grady Booch, renomado engenheiro de software, constitui a estrutura essencial que orienta o desenvolvimento da aplicação, traduzindo requisitos do sistema em uma representação tangível. No âmbito do projeto Comtur, a arquitetura de software representa a espinha dorsal do sistema, proporcionando uma visão organizada e coesa de como os componentes interagem entre si. Como ressaltado por Ralph Johnson, um dos autores do livro "Design Patterns: Elements of Reusable Object-Oriented Software", a arquitetura de software é o "esqueleto" que sustenta a aplicação, delineando as relações entre os diversos módulos e garantindo uma estrutura robusta.

A necessidade de uma arquitetura sólida no Comtur transcende a mera organização do código. Conforme preconiza Martin Fowler, especialista em arquitetura de software, "a arquitetura é importante para lidar com a complexidade inerente ao desenvolvimento de software". No contexto do Comtur, a complexidade do sistema exige uma arquitetura bem definida para facilitar não apenas a implementação atual, mas também a manutenção futura e a expansão do projeto.

Assim, a arquitetura de software no Comtur não é apenas uma formalidade técnica, mas uma estratégia fundamental para assegurar a clareza na concepção, a facilidade na evolução e a capacidade de adaptação às demandas emergentes. Nesse sentido, ela se revela como um guia orientador essencial, seguindo as palavras de Fred Brooks, autor de "The Mythical Man-Month": "Arquitetura é sobre as partes do sistema e as suas interações. Essa é a tarefa central de qualquer arquitetura."

6.1 ARQUITETURA DE DESENVOLVIMENTO UTILIZADA

O desenvolvimento do projeto Comtur incorporou uma arquitetura em camadas, inspirada em diretrizes promovidas por especialistas em engenharia de software renomados. A implementação da API em C# reflete a escolha de uma linguagem respaldada por práticas sólidas, com influência de líderes da indústria como Anders Hejlsberg, arquiteto-chefe da linguagem C# na Microsoft. A opção pelo React para a interface do usuário segue a recomendação de especialistas como Jordan Walke, um dos desenvolvedores principais do React, destacando-se pela capacidade de criar componentes reutilizáveis e uma experiência de usuário dinâmica. A utilização do framework Tailwind CSS, reconhecido por sua abordagem prática no design e estilização, alinha-se às práticas sugeridas por Adam Wathan, criador do Tailwind CSS, e reflete uma tendência moderna na comunidade de desenvolvimento.

O fluxo de interação do usuário no sistema inicia-se com uma requisição, seguindo as práticas recomendadas por especialistas em segurança de dados como Troy Hunt e sua ênfase na validação rigorosa dos dados na camada de Controllers. Validados, esses dados são encaminhados para a camada de Repositories, adotando conceitos disseminados por especialistas em arquitetura de software como Martin Fowler, que propõe uma abordagem eficiente para a persistência de dados.

Na camada de Repositories, a criação do método de post como uma interface para a persistência de dados segue as recomendações de Robert C. Martin, conhecido como "Uncle Bob", para design de código limpo e coeso. Esta camada age como intermediária entre a lógica de negócios e a manipulação de dados no banco, implementando princípios sugeridos por especialistas em arquitetura de software como Eric Evans, autor do livro "Domain-Driven Design".

A camada de Services desempenha um papel central na execução da lógica de negócios, adotando práticas propostas por Michael Feathers, defensor da escrita de código testável e de fácil manutenção. A integração eficaz entre a camada de Services e a DbContext segue as melhores práticas de desenvolvimento sugeridas por especialistas como Scott Guthrie, Vice-Presidente Executivo da Microsoft Cloud + AI, garantindo uma comunicação suave e eficiente entre a aplicação e o banco de dados do Comtur.

Os dados são, então, persistidos no banco de dados, consolidando a integridade e confiabilidade do sistema, em conformidade com as práticas recomendadas por especialistas em persistência de dados como Joe Celko. Essa abordagem em camadas, influenciada por princípios e práticas recomendadas de especialistas renomados, não apenas facilita a manutenção do código e a adição de novos recursos, mas também estabelece uma clara separação de responsabilidades entre as diferentes partes da aplicação, promovendo uma arquitetura flexível e escalável.

As ferramentas empregadas no âmbito do projeto Comtur abrangem softwares, linguagens de programação, ambientes de desenvolvimento e outras aplicações, cada uma desempenhando funções específicas durante o ciclo de vida do sistema. Conforme ressalta Anders Hejlsberg, arquiteto-chefe da linguagem C# na Microsoft, "a escolha da linguagem de programação é crucial para o sucesso do projeto, influenciando diretamente na clareza e eficiência da lógica de negócios implementada."

Essas ferramentas são imperativas, uma vez que, conforme destaca Martin Fowler, especialista em padrões de arquitetura de software, elas proporcionam funcionalidades especializadas que simplificam tarefas complexas e aumentam a eficiência global do

desenvolvimento. Do ponto de vista de Scott Guthrie, Vice-Presidente Executivo da Microsoft Cloud + AI, "a seleção cuidadosa de ambientes de desenvolvimento, como o Visual Studio, contribui para uma codificação eficiente e para a manutenção da qualidade do software."

A escolha cuidadosa de tecnologias, respaldada por especialistas e líderes de renome, como C# e React, reflete a preferência por ferramentas estabelecidas e amplamente reconhecidas no desenvolvimento de software. O uso do Tailwind CSS, também endossado por especialistas em design de interfaces como Adam Wathan, reforça o compromisso com uma experiência de usuário intuitiva e atraente. Essa arquitetura estruturada, moldada por princípios e práticas recomendadas de especialistas, convergem para assegurar eficiência, flexibilidade e desempenho no Comtur, fornecendo não apenas uma experiência sólida para os usuários atuais, mas também uma base robusta para futuras iterações e aprimoramentos do sistema.

6.1.1 LINGUAGEM DE PROGRAMAÇÃO C#

Figura 40 — Imagem da Logo do C#



Fonte: Medium (2019).

Acima temos a logo da linguagem implementada no projeto Comtur, o C# desempenha um papel crucial na construção de uma API robusta e eficiente. Como destacado por Anders Hejlsberg, arquiteto-chefe da linguagem C# na Microsoft, "C# é projetado para ser uma linguagem de programação simples, moderna e orientada a objetos". Essa escolha linguística alinha-se à necessidade de uma sintaxe clara e estruturada, facilitando o desenvolvimento de lógica de negócios coesa e de fácil manutenção.

Martin Fowler, especialista em padrões de arquitetura de software, ressalta que "C# é particularmente adequado para a implementação de arquiteturas em camadas". No contexto do Comtur, que adota uma arquitetura em camadas para promover coesão e flexibilidade, a escolha do C# contribui para uma estrutura organizacional do código-fonte que facilita a compreensão e a manutenção.

Além disso, Scott Guthrie, Vice-Presidente Executivo da Microsoft Cloud + AI, destaca a integração do C# com o ecossistema .NET, fornecendo acesso a uma ampla gama de bibliotecas e ferramentas. Essa integração eficaz é essencial para garantir a comunicação suave entre as diferentes camadas da aplicação e o banco de dados do Comtur, proporcionando uma base sólida para o desenvolvimento sustentável da API. Em suma, a escolha do C# no Comtur não apenas se baseia em sua elegância e simplicidade, mas também em sua adaptabilidade e integração eficaz, atendendo aos requisitos específicos de uma aplicação centrada em dados e funcionalidades avançadas.

6.1.2 VISUAL STUDIO

Figura 41 — Logo do Visual Studio



Fonte: 1000 Logos (2023).

Acima temos a logo do Visual Studio que foi escolhido como ambiente de desenvolvimento para a implementação da API em C# no projeto Comtur, respaldada por sua reputação consolidada e recursos abrangentes. Como enfatiza Anders Hejlsberg, arquiteto-chefe da linguagem C# na Microsoft, o Visual Studio é reconhecido por sua robustez e conjunto de ferramentas integradas, proporcionando um ambiente de desenvolvimento completo para a criação de aplicativos empresariais de alta qualidade.

Martin Fowler, especialista em padrões de arquitetura de software, destaca que o Visual Studio oferece funcionalidades avançadas para a implementação de arquiteturas em camadas, facilitando a estruturação e organização do código-fonte. Essa capacidade é crucial para o projeto Comtur, que adota uma arquitetura em camadas para garantir coesão e flexibilidade no desenvolvimento.

A integração do Visual Studio com o ecossistema .NET é outro ponto destacado por especialistas como Scott Guthrie, Vice-Presidente Executivo da Microsoft Cloud + AI. Essa integração eficiente simplifica o processo de desenvolvimento em C#, permitindo uma comunicação fluida entre as diferentes camadas da aplicação e o banco de dados do Comtur.

Portanto, ao optar pelo Visual Studio como ambiente principal para o desenvolvimento da API em C#, a equipe do projeto Comtur busca não apenas uma ferramenta consolidada e poderosa, mas também um ambiente que facilite a implementação de boas práticas de arquitetura e proporcione um fluxo de trabalho eficaz para a criação de uma aplicação escalável e de alta qualidade.

6.1.3 LINGUAGEM DE PROGRAMAÇÃO REACT

Figura 42 — Logo do React



Fonte: Logos World (2023).

Na figura acima, podemos ver a logo do React, uma biblioteca de JavaScript desenvolvida pelo Facebook, projetada para facilitar a criação de interfaces de usuário dinâmicas e eficientes. Baseando-se no conceito de componentes reutilizáveis, React simplifica a construção de interfaces interativas, tornando-as mais escaláveis e fáceis de manter. No contexto do projeto Comtur, React foi adotado para a implementação da interface do usuário, proporcionando uma experiência de usuário responsiva e atraente. Com a capacidade de criar componentes independentes, React permite a modularidade do código, facilitando a manutenção e a adição de novos recursos ao longo do tempo. Além disso, seu modelo de atualização eficiente do DOM (Document Object Model) contribui para um desempenho ágil, crucial para uma aplicação que visa oferecer uma experiência de usuário fluida ao explorar pontos turísticos, notícias e eventos da cidade no Comtur. Ao utilizar React no projeto, há uma

vantagem significativa na construção de uma interface intuitiva e responsiva, alinhada com as expectativas modernas de experiência do usuário na web.

A implementação da interface do usuário, utilizando React, foi conduzida no ambiente de desenvolvimento Visual Studio Code. Reconhecido por sua leveza, extensibilidade e ampla aceitação na comunidade de desenvolvimento, o Visual Studio Code proporcionou um ambiente ágil e eficiente para a criação de componentes reutilizáveis e a construção dinâmica da interface do usuário. Sua integração com ferramentas modernas de desenvolvimento web e extensões especializadas para React contribuíram para uma experiência de codificação produtiva e centrada no desenvolvedor.

6.1.4 VISUAL STUDIO CODE

Figura 43 — Logo do Visual Studio Code



Fonte: Logos World (2021).

A escolha do Visual Studio Code como ambiente de desenvolvimento para o projeto Comtur reflete a preferência por uma ferramenta versátil e amplamente adotada na comunidade de desenvolvimento. Como destacado por Martin Fowler, renomado autor e defensor de práticas ágeis, o Visual Studio Code é reconhecido por sua leveza, simplicidade e extensibilidade, proporcionando um ambiente de codificação eficiente e centrado no desenvolvedor. Através de sua interface intuitiva e suporte robusto para diversas linguagens de programação, o Visual Studio Code se destaca como uma escolha versátil, como observa Chris Dias, gerente de programa da Microsoft, afirmando que "o VS Code foi projetado para ser uma ferramenta leve, mas poderosa, que atende às necessidades de desenvolvedores modernos".

Além disso, o Visual Studio Code se integra perfeitamente ao ecossistema do React, conforme mencionado por Jordan Walke, um dos desenvolvedores principais do React. Sua extensibilidade permite a incorporação de plug-ins e extensões especializadas para React,

facilitando o desenvolvimento de componentes reutilizáveis e a construção dinâmica da interface do usuário. Dessa forma, a escolha do Visual Studio Code para o desenvolvimento com React no projeto Comtur está alinhada com a busca por uma ferramenta que não apenas atenda às exigências técnicas, mas também proporcione uma experiência de codificação ágil e eficaz para a equipe de desenvolvimento.

6.1.5 POSTGRESQL

Figura 11 — Logo do PostgreSQL



Fonte: Logos World (2022).

A figura acima, contém a logo do PostgreSQL, que desempenha um papel crucial na gestão eficiente do banco de dados e na garantia da integridade dos dados. O PostgreSQL, como ressaltado por Dave Page, diretor do projeto PostgreSQL, é uma poderosa base de dados relacional de código aberto, reconhecida por sua confiabilidade e recursos avançados. "O PostgreSQL oferece um robusto sistema de gerenciamento de dados, suportando transações ACID e proporcionando flexibilidade para atender a uma variedade de necessidades de aplicativos", destaca Page.

A escolha do PGAdmin como ferramenta de administração do PostgreSQL é respaldada pela sua interface intuitiva e recursos abrangentes. Conforme afirmado por Joe Conway, membro sênior da comunidade PostgreSQL, "o PGAdmin é uma ferramenta essencial para administrar o PostgreSQL, permitindo visualizar e manipular dados, criar consultas complexas e gerenciar esquemas de banco de dados de maneira eficaz."

A integração do PGAdmin com o PostgreSQL facilita tarefas críticas, desde a criação e modificação de tabelas até a execução de consultas complexas. Scott Guthrie, Vice-Presidente Executivo da Microsoft Cloud + AI, destaca a importância de ferramentas de gerenciamento eficientes, afirmando que "o uso do PGAdmin agrega valor à administração do PostgreSQL,

proporcionando uma experiência simplificada e eficaz para os desenvolvedores e administradores de banco de dados."

A escolha do PostgreSQL e do PGAdmin no Comtur é embasada na necessidade de um sistema robusto de gerenciamento de banco de dados e de uma ferramenta intuitiva para administrar dados de maneira eficiente. Essa combinação, respaldada por especialistas renomados, assegura a confiabilidade e a eficácia na gestão do banco de dados, contribuindo para a estabilidade e integridade do sistema como um todo.

Em síntese, a arquitetura de software no projeto Comtur emerge como a espinha dorsal que sustenta a concepção, desenvolvimento e evolução contínua da aplicação. Inspirada pelas palavras de Grady Booch, que a define como a estrutura fundamental que traduz os requisitos em uma representação tangível, e reforçada por especialistas como Martin Fowler e Ralph Johnson, a arquitetura no Comtur não é apenas uma questão de organização do código, mas sim uma estratégia indispensável para lidar com a complexidade inerente ao desenvolvimento de software.

Ao adotar uma abordagem coesa e bem definida na escolha de linguagens, ferramentas, e na implementação de arquiteturas em camadas, o Comtur busca não apenas atender às exigências técnicas, mas também garantir clareza, manutenibilidade e adaptabilidade futura. A combinação de tecnologias como C# e React, apoiada por ambientes de desenvolvimento robustos como Visual Studio e Visual Studio Code, representa um compromisso com a excelência na construção da lógica de negócios e da interface do usuário.

A gestão eficiente do banco de dados, conduzida pelo PostgreSQL e administrada pelo PGAdmin, destaca a importância de escolhas estratégicas em todas as fases do desenvolvimento. Ao seguir os princípios de arquitetura delineados por especialistas renomados, o Comtur assegura uma base sólida para a expansão futura, a manutenção eficaz e a sustentabilidade do sistema. Nesse contexto, a arquitetura de software não é apenas uma camada técnica, mas sim um alicerce vital que guia o projeto em direção à eficiência, escalabilidade e sucesso duradouro.

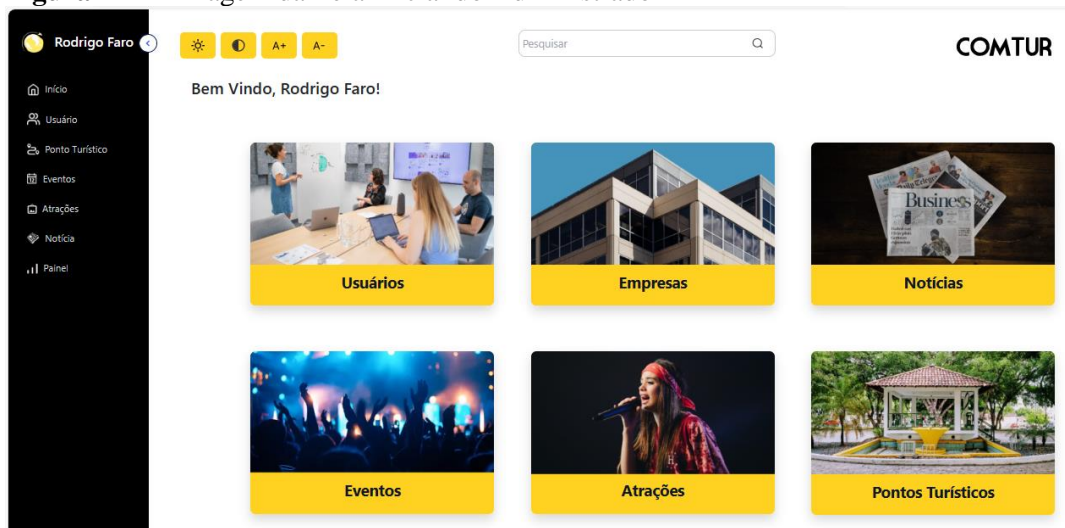
6.2 TELAS DO SISTEMA

As telas de um sistema representam as interfaces visuais através das quais os usuários interagem com as funcionalidades e informações disponíveis. Como salienta Don Norman, um pioneiro em design centrado no usuário, "a interface é o produto". Nesse contexto, as telas desempenham um papel crucial no Comtur, sendo a porta de entrada para usuários e administradores interagirem com as diversas funcionalidades do sistema.

A importância das telas no Comtur é evidenciada pelo fato de que constituem a primeira impressão do usuário em relação à aplicação, conforme ressaltado por Jakob Nielsen ao afirmar que "a primeira impressão é a mais importante". Telas intuitivas, claras e eficientes contribuem para uma experiência de usuário positiva, enquanto interfaces confusas ou desorganizadas podem resultar em frustração e dificuldades no uso do sistema.

Ao adotar práticas de design centrado no usuário, o Comtur busca garantir que suas telas sejam não apenas esteticamente agradáveis, mas também funcionais e adaptadas às necessidades dos usuários. Isso alinha-se com os princípios destacados por Steve Krug, autor de "Don't Make Me Think", que enfatiza a importância de interfaces que permitam aos usuários realizar tarefas de forma intuitiva, sem a necessidade de esforço cognitivo excessivo.

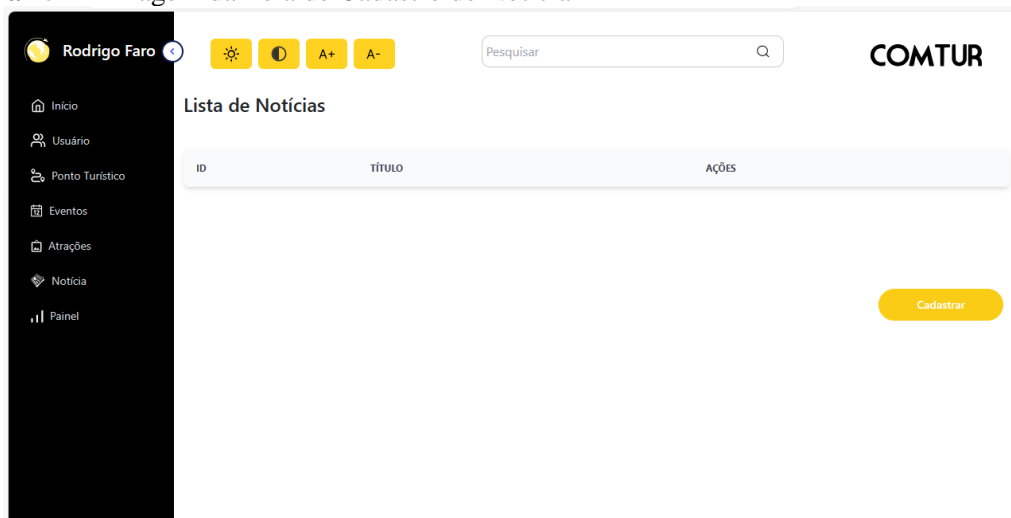
Figura 44 — Imagem da Tela Inicial do Administrador



Fonte: Elaborado pelas autoras.

A tela inicial do administrador no sistema Comtur é projetada para fornecer uma visão consolidada e funcional das principais áreas e funcionalidades disponíveis. Inspirada por conceitos de design centrado no usuário, a tela destaca um Navbar superior para navegação rápida entre diferentes seções e um Sidebar à esquerda que oferece acesso direto a funcionalidades específicas. Seis cards dispostos no corpo principal fornecem atalhos visuais para áreas importantes, proporcionando uma experiência coesa e intuitiva para os administradores, refletindo a prioridade do Comtur em oferecer uma interface administrativa eficiente.

Figura 45 — Imagem da Tela de Cadastro de Noticia



Fonte: Elaborado pelas autoras.

A tela de cadastro de notícias no sistema Comtur é concebida para simplificar o processo de inserção de informações relevantes. Inspirada por conceitos de usabilidade de Steve Krug, o design destaca áreas claras para título, conteúdo e data de publicação, facilitando a compreensão e inserção de dados. Botões de ação, como "Salvar" e "Cancelar", são estrategicamente posicionados para garantir uma interação eficiente. Essa abordagem busca otimizar a usabilidade durante o cadastro de notícias, proporcionando uma experiência intuitiva e centrada no usuário.

Figura 46 — Imagem da Tela de Cadastro de Turismo e Tipo de Turismo

Fonte: Elaborado pelas autoras.

A tela de cadastro de turismo no sistema Comtur é desenvolvida para facilitar a inclusão e gestão de informações sobre pontos turísticos. Inspirada pelos princípios de Don Norman, o design destaca campos intuitivos para detalhes como nome, descrição e atrações associadas. Botões de ação, como "Salvar" e "Cancelar", são posicionados estrategicamente

para promover uma interação eficiente. Essa abordagem visa otimizar a usabilidade durante o cadastro de informações turísticas, garantindo uma experiência de usuário eficaz e alinhada às melhores práticas de design centrado no usuário.

Figura 47 — Imagem da Tela de Cadastro de Atrações e Tipo de Atrações

Cadastro

Atração

*Nome:

*Descrição:

*Tipo:

Id	Nome
----	------

Fonte: Elaborado pelas autoras.

A tela de cadastro de atrações no Comtur é projetada para oferecer uma representação visual e interativa do processo de inserção de novas atrações turísticas. Inspirada pelos princípios de design de Bill Buxton, o layout destaca áreas de preenchimento para informações cruciais, como nome, descrição e localização, visando uma fácil compreensão e entrada de dados pelos administradores. A abordagem interativa do protótipo busca fornecer uma visão tangível do design proposto, permitindo iterações rápidas e feedback contínuo.

7 CONCLUSÃO

À medida que a internet se consolida como uma ferramenta crucial no cenário do turismo, é imperativo que as cidades abracem estratégias inovadoras para promover seus destinos. Em meio a esse panorama, o projeto Comtur surge como uma resposta visionária para impulsionar o turismo em Jales, preenchendo lacunas identificadas e proporcionando uma experiência rica em conteúdo e interatividade.

A análise detalhada dos desafios atuais revelou a falta de uma presença online exclusiva para o turismo na cidade. A concentração do turismo na área de saúde, a falta de divulgação de locais turísticos menos conhecidos e a ausência de um sistema eficaz para conectar pontos culturais foram identificados como obstáculos significativos. O Comtur propõe não apenas superar esses desafios, mas transformá-los em oportunidades de crescimento e desenvolvimento.

Os requisitos funcionais cuidadosamente delineados refletem uma abordagem abrangente, desde a inclusão de botões de ação para atualização de registros até um sofisticado sistema de pesquisa. Isso não apenas promete destacar pontos turísticos, mas também proporcionar uma experiência envolvente para os usuários, conectando-os de maneira significativa à riqueza cultural de Jales.

Além disso, os requisitos não funcionais solidificam a visão do Comtur como um projeto de alta qualidade. Compatibilidade com diversos navegadores e dispositivos, um banco de dados eficiente e tecnologia de desenvolvimento robusta são elementos-chave que garantem a acessibilidade, confiabilidade e desempenho eficiente do sistema.

O Comtur não se limita a ser um mero guia turístico online; é uma iniciativa que visa integrar-se à comunidade local. A ênfase na segurança, revisão de conteúdo e a inclusão de um leitor de QR Code destacam a preocupação com a criação de um ambiente online confiável e interativo.

O potencial de impacto do Comtur transcende o setor do turismo. Ao gerar empregos, criar oportunidades de negócios e melhorar a qualidade de vida dos residentes e visitantes, o projeto se posiciona como um catalisador para o desenvolvimento sustentável em Jales.

Em suma, o desenvolvimento do software não apenas se mostra viável, mas promissor para transformar a experiência turística em Jales. Ao abraçar a inovação, conectar a comunidade e elevar os padrões de qualidade, o Comtur emerge como uma jornada promissora, pronta para deixar uma marca duradoura no turismo e no desenvolvimento da cidade.

8 REFERÊNCIAS

1000 LOGOS (2023). **Logo Visual Studio**. <<https://1000logos.net/visual-studio-logo/>>. Acesso em: 20 nov.2023

BOOCH, G. (1991). "**Object-Oriented Analysis and Design with Applications**." Addison-Wesley.

BROOKS, F. P. (1995). "**The Mythical Man-Month: Essays on Software Engineering**." Addison-Wesley.

DELIBERATO, P. C. P. **Fisioterapia preventiva: fundamentos e aplicações**. São Paulo: Manole, 2002.

DIAS, E. C.; MELO, E. M. de. Políticas públicas em saúde e segurança no trabalho. In: MENDES, R. (Org.). **Patologia do trabalho**. 2. ed. atual. e ampl. São Paulo: Atheneu, 2007. v. 2. p. 1683-1720.

FOWLER, M. (2003). "**Patterns of Enterprise Application Architecture**." Addison-Wesley.

GUEDES, Gilleanes T. A. **UML 2 Uma Abordagem prática**, São Paulo: Novatec, 2009.

GUEDES, Gilleanes T. A. **UML Uma Abordagem prática**, 3 ed. São Paulo: Novatec, 2008.

JOHNSON, R., GAMMA, E., HELM, R., & VLISSIDES, J. (1994). "**Design Patterns: Elements of Reusable Object-Oriented Software**." Addison-Wesley.

KRUG, S. (2014). "**Don't Make Me Think**." New Riders.

LOGOS WORLD (2021). **Logo Visual Studio Code**. <<https://logos-world.net/wp-content/uploads/2023/08/React-Symbol.png>>. Acesso em: 20 nov.2023

LOGOS WORLD (2023). **Logo React**. <<https://logos-world.net/react-logo/>>. Acesso em: 20 nov.2023

MACHADO, F. N. R. **Banco de dados: projeto e implementação**. 4. Ed. São Paulo: Érica, 2020.

NIELSEN, J. (2006). "**Ten Usability Heuristics**."

NORMAN, D. A. (2002). "**The Design of Everyday Things**." Basic Books.

PÁDUA, P. F. W. **Engenharia de Software**. 6. ed. Porto Alegre: AMGH, 2010.

PRESSMAN, R. S. **Engenharia de Software: fundamentos, métodos e padrões**. 3. ed. Rio de Janeiro: LTC, 2015.

SOMMERVILLE, I. **Engenharia de Software**. 10. ed. São Paulo: Pearson Education do Brasil, 2018.

UNGER, Russ; CHANDLER, Carolyn. **O guia para projetar UX**. 1. Ed. São Paulo: Alta Books, 2009.

WATRALL, E; SIARTO, J. Use A Cabeça! **Web Design**. Alta Books, 2009.