INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO NORTE

JOÁS JORDSON KELWYN PINHEIRO DE CARVALHO

E-NARRATIVAS: IMPLEMENTAÇÃO BACK-END DE GERENCIAMENTO DE DADOS E MODULARIZAÇÃO DE TELAS

Canguaretama, RN – 2025

JOÁS JORDSON KELWYN PINHEIRO DE CARVALHO

E-NARRATIVAS: IMPLEMENTAÇÃO BACK-END DE GERENCIAMENTO DE DADOS E MODULARIZAÇÃO DE TELAS

(Ficha Catalográfica no verso, Biblioteca responsável pela elaboração)

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Sistemas para Internet do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Dr. Bruno Gomes de Araujo

CANGUARETAMA/RN

2025

JOÁS JORDSON KELWYN PINHEIRO DE CARVALHO

MODULARIZAÇÃO DO LAYOUT E IMPLEMENTAÇÃO DO CRUD NARRATIVAS PARA O REPOSITÓRIO E-NARRATIVAS

Trabalho de Conclusão de Curso apresentado ao Curso Superior de Tecnologia em Sistemas para Internet do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, em cumprimento às exigências legais como requisito parcial à obtenção do título de Tecnólogo em Sistemas para Internet.

Aprovado em: ___/___/____

Banca Examinadora

Documento assinado digitalmente
gov.br
BRUNO GOMES DE ARAUJO
Data: 03/09/2026 22:11:32-0300
Verifique em https://validar.iti.gov.br

_____ _____

Dr. Bruno Gomes de Araujo

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Documento assinado digitalmente
gov.br
TATIANA AMARAL SORRENTINO
Data: 04/09/2026 16:11:42-0300
Verifique em https://validar.iti.gov.br

_____ _____

Dr. Tatiana Amaral Sorrentino - Examinadora

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

Documento assinado digitalmente
gov.br
LEANDRO SILVA COSTA
Data: 04/09/2026 15:57:34-0300
Verifique em https://validar.iti.gov.br

_____ _____

Dr. Leandro Silva Costa - Examinador

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte

RESUMO

As narrativas são cruciais para a sociedade, pois permitem a transmissão de ideias e a preservação de tradições e memórias. Contudo, a ausência de um local centralizado para armazená-las e consultá-las dificulta o acesso ao conhecimento. Diante disso, surge a necessidade de uma solução que facilite o acesso a informações fundamentais para o crescimento intelectual sobre o passado de um povo, lugar e país, permitindo também o aprendizado sobre outras culturas. Assim, iniciou-se a criação de um sistema web, funcionando como um portfólio online, para centralizar o registro dessas narrativas e recuperar dados culturais para o meio comum. Este trabalho apresentará o processo de elaboração e desenvolvimento das operações básicas de banco de dados (CRUD - Criar, Ler, Atualizar, Excluir) para este repositório. A pesquisa aborda a modularização do layout das páginas de tela inicial e cadastro de narrativas para otimizar a manutenção e reutilização de código, bem como as correções e implementações necessárias no banco de dados.

Palavras-chaves: Narrativas; Sistema Web; Repositório on-line; Banco de Dados, CRUD.

LISTA DE ILUSTRAÇÕES

Figura 1. Views e URLs no Django: Fundamentos Para Gerenciar Rotas. 13

Figura 2. Parte da tela Inicial do site com área que se repete marcada em vermelho. 18

Figura 3. Parte da tela Inicial do site com área que se repete marcada em vermelho. 19

Figura 4. Parte da tela de cadastro das narrativas com área que se repete marcada em vermelho. 19

Figura 5. Pasta templates do projeto Django. 20

Figura 6. Código do banco alterado. 22

Figura 7. Tela de Listagem Narrativas. 23

Figura 8. Tela de cadastro das Narrativas. 24

Figura 9. Tela de cadastro das Narrativas. 24

Figura 10. Tela de Listagem das Narrativas. 25

Figura 11. Tela de Listagem dos Tipos de Narrativas. 26

Figura 12. Tela de Cadastro de Tipo de Narrativa. 27

Figura 13. Tela de Listagem de Estilo. 28

Figura 14. Tela de Cadastro de Estilo. 29

SUMÁRIO

SUMÁRIO

1. **INTRODUÇÃO 12**
   - 1.1. OBJETIVO GERAL 13
   - 1.2. OBJETIVOS ESPECÍFICOS 13
2. **REFERENCIAL TEÓRICO 14**
   - 2.1. Narrativas Brasileiras 14
   - 2.2. Desenvolvimento Web 15
   - 2.2.1. HTML 15
   - 2.2.2. Python 16
   - 2.2.3. Django 16
   - 2.3. Versionamento de Código 17
   - 2.4. Scrum 18
3. **METODOLOGIA 19**
4. **RESULTADOS 19**
   - 4.1. Modularização de Layout 20
   - 4.2. Banco de dados 21
   - 4.3 Desenvolvimento 22
   - 4.3.1 Narrativa 22
   - 4.3.2 Layout Tipo_Narrativa 25
   - 4.3.3 Layout Estilo_Narrativa 26
5. **CONCLUSÃO 28**

REFERÊNCIAS 29

ANEXOS 30

1. INTRODUÇÃO

As narrativas, frequentemente permeadas de mistério e fantasia, são cruciais para a transmissão de valores sociais. Além de ensinar e preservar memórias coletivas, elas podem reinterpretar eventos passados, ajustando a forma como são percebidos e entendidos (RABELO, 2011).

Ela já é uma prática universal entre os seres humanos. Em muitas sociedades conhecidas e em diversos períodos históricos, as pessoas contam lendas, superstições, curiosidades, histórias e estórias, alguns integrados a rituais religiosos, outros usados para instrução e muitos simplesmente para entretenimento (CORREIA, 2012).

Entretanto, essas narrativas não possuem um método de registro oficial, esses registros ocorrem de maneira isolada em cada município, pois, na maioria das localidades, não há um meio eficiente de acesso a essas informações. Isso dificulta o conhecimento tanto aos moradores quanto aos visitantes dessas cidades.

Neste cenário, faz-se necessário a criação de um sistema web, que funcionará como repositório de narrativas, e assim também disponibilizar para toda a população. Através dos projetos de pesquisa intitulados “Repositório on-line de narrativas dos municípios brasileiros: estudo de caso na cidade de Arês-RN”, “e-Narrativas: Repositório on-line de narrativas dos municípios brasileiros (Etapa 2024)” e “Desenvolvimento do e-Narrativas: repositório on-line de narrativas dos municípios brasileiros - Etapa 2025”, aprovados respectivamente nos editais nº 01/2023 – PROPI/RE/IFRN, nº 01/2024 – PROPI/RE/IFRN e nº 01/2025 – PROPI/RE/IFRN, iniciou o desenvolvimento do repositório e-Narrativas.

O autor do presente trabalho ficou responsável pela modularização do layout do repositório, correções no banco de dados e funcionamento das operações básicas para gerenciamento de narrativas.

O restante do trabalho está estruturado da seguinte forma: o Capítulo 2 apresenta o referencial teórico sobre narrativas e tecnologias utilizadas; o Capítulo 3 descreve a metodologia utilizada; o Capítulo 4 apresenta os resultados; e o Capítulo 5 discute as considerações finais.

1.1. OBJETIVO GERAL

O objetivo principal do presente trabalho consiste na modularização das páginas de tela inicial e cadastro de narrativas para otimizar a manutenção e reutilização de código no repositório on-line de narrativas.

1.2. OBJETIVOS ESPECÍFICOS

O presente trabalho teve como objetivos específicos:

- Realizar uma pesquisa bibliográfica sobre desenvolvimento web, Python, Django, HTML, CSS, versionamento de código e metodologias ágeis.
- Revisar o layout das páginas de tela inicial e cadastro de narrativas do repositório.
- Modularizar o layout das páginas de tela inicial e cadastro de narrativas no projeto Django.
- Implementar correções necessárias no banco de dados do repositório online de narrativas.
- Garantir o funcionamento das operações básicas (CRUD) para o gerenciamento de narrativas.

2. REFERENCIAL TEÓRICO

O referencial teórico estabelece a fundamentação teórica que serve de base para o desenvolvimento do projeto e-Narrativas. Serão apresentadas as temáticas essenciais que guiaram a concepção e implementação do sistema, detalhando as competências técnicas e os conceitos fundamentais que sustentam sua arquitetura e funcionalidades. A discussão abordará desde as narrativas brasileiras e sua relevância cultural, até as tecnologias de desenvolvimento web empregadas, incluindo HTML, Python e o framework Django, além de ferramentas de versionamento de código como Git e GitHub.

2.1. Narrativas Brasileiras

As narrativas que se desenvolveram no Brasil formam um cenário literário de grande interesse. Suas origens históricas podem ser encontradas no Período Colonial, que se estendeu de 15000 a 1822. Essa época foi marcada pelos primeiros registros escritos, os quais foram trazidos pelos colonizadores portugueses. É importante notar que esses textos primários consistem principalmente em relatos de natureza religiosa e informações sobre a terra. Eles revelavam, de maneira marcante, a perspectiva dos europeus em relação ao novo território e às populações nativas. (Castello,1999).

Algumas literaturas têm o poder de cultivar e enriquecer a sensibilidade de um homem, sem que ele precise buscar experiências fora delas.(Candido, 1964). A própria imersão nessas literaturas já oferece um aprendizado cultural e um desenvolvimento da capacidade de sentir e compreender o mundo de forma mais apurada. A experiência da leitura se basta para enriquecer o ser humano nesses aspectos, portanto, essas histórias do folclore oferecem uma janela para a complexidade e a diversidade do universo imaginário do povo brasileiro.

A expressão "narrativas" designa o modo pelo qual as histórias são relatadas, seja através da fala ou da escrita. Essa forma de expressão possui a capacidade intrínseca de despertar o interesse do ouvinte ou leitor, de gerar uma conexão emocional e de tornar mais acessível o entendimento dos acontecimentos que são narrados (Bruner, 1986). Entre as diversas formas narrativas, destacam-se os

contos, romances, lendas e mitos, cada um com suas convenções e propósitos específicos.

2.2. Desenvolvimento Web

A internet é uma imensa e complexa rede de comunicação mundial que conecta milhares de dispositivos espalhados pelo mundo, possibilitando a troca de dados e informações de forma rápida e eficiente. Além disso, a internet oferece uma imensidade de serviços essenciais, por exemplo, a navegação em sites (páginas web), transferência de arquivos e a troca de mensagens pelo correio eletrônico (e-mail). Esses serviços são fundamentais no nosso cotidiano, facilitando a comunicação, o aprendizado e o acesso à informação em tempo real. (COSTA, 2007).

O Protocolo de Transferência de Hipertexto, conhecido pela sigla HTTP (Hypertext Transfer Protocol), desempenha um papel crucial no funcionamento da Web. Em outras palavras, ele constitui o conjunto de regras que viabilizam a transmissão de páginas da internet. As solicitações HTTP são estruturadas através de diferentes métodos, a exemplo de GET, POST, PUT e DELETE, os quais comunicam ao servidor a operação específica que se deseja executar. Por sua vez, as respostas enviadas pelo servidor HTTP incluem um código de status, que informa sobre o sucesso ou a falha da requisição, podendo também conter informações relevantes, como o conteúdo da página que foi solicitada. (ROSS e KUROSE, 2013).

2.2.1. HTML

O HTML, sigla em inglês para *HyperText Markup Language*, que em portugues significa linguagem para marcação de hipertexto. Trata-se de uma linguagem de marcação empregada para instruir o navegador sobre a maneira de organizar e apresentar o conteúdo de uma página web. Sua estrutura se baseia em um conjunto de elementos, utilizados para delimitar, envolver ou identificar distintas seções do conteúdo, possibilitando que estas sejam formatadas ou comportem-se de maneiras específicas. Essas marcações são realizadas por meio de *tags*, que consistem em palavras reservadas indicando o início e, frequentemente, o fim de uma marcação. Tais *tags* são delimitadas pelos sinais de meno (<) e maior (>) e

possuem a capacidade de transformar o conteúdo inserido dentro de um elemento, por exemplo, em um hiperlink, que direciona para outra página da web ou de aplicar estilos como o itálico ao texto, entre outras funcionalidades. (Silva, 2008).

De acordo com as especificações do W3C (*World Wide Web Consortium*) que é um consórcio internacional no qual toda sua equipe trabalha a favor do desenvolvimento de padrões web (W3C, 2022). A linguagem HTML evoluiu ao longo do tempo, sendo a versão 5 a mais recente. Essa atualização significativa introduziu diversas inovações e ampliou o suporte ao CSS3 (*Cascading Style Sheets* nível 3).

O CSS (Cascading Style Sheets, ou Folhas de Estilo em Cascata) é uma linguagem de estilo utilizada para descrever a apresentação de um documento escrito em uma linguagem de marcação, como HTML. Ele permite separar o conteúdo (HTML) da apresentação visual (CSS), tornando o desenvolvimento e a manutenção de páginas web muito mais eficientes.

Segundo Jacobs (2014), a utilização do CSS proporciona um design profissional às páginas web, aprimorando a experiência de navegação dos usuários. A principal vantagem do CSS é a sua capacidade de aplicar estilos de forma consistente em múltiplas páginas. Em vez de definir o estilo para cada elemento HTML individualmente em cada página, é possível criar um arquivo CSS externo e vinculá-lo a todas as páginas do site. Assim, qualquer alteração de estilo pode ser feita em um único lugar e refletir em todo o site.

2.2.2. Python

Desenvolvida na década de 1980 por Guido van Rossum no Instituto de Pesquisa Nacional para Matemática e Ciência da Computação (CWI - Centrum Wiskunde & Informatica), na Holanda, *Python* é definida como uma linguagem de programação de alto nível, caracterizada por ser interpretada, de tipagem forte e dinâmica. Seu desenvolvimento foi motivado pela intenção de simplificar o processo de programação, abstraindo as complexidades do hardware para o programador. Essa filosofia resultou em uma linguagem notavelmente simples, porém de grande poder, que encontrou vasta aplicação em campos como Ciência de Dados, Análise de Dados, Inteligência Artificial, Aprendizado Profundo, Desenvolvimento Web e no desenvolvimento de aplicações de propósito geral. A vasta gama de bibliotecas

disponíveis em *Python* facilita a reutilização de código e funcionalidades no desenvolvimento de software. (SANTIAGO, 2022).

Python se destaca como uma linguagem de programação multiplataforma, atributo que assegura a sua portabilidade e compatibilidade com uma variedade de sistemas operacionais. Adicionalmente, por ser de código aberto, sua codificação é acessível e passível de modificação por qualquer indivíduo, o que possibilita a criação de bibliotecas personalizadas pela comunidade. A linguagem também oferece suporte à interoperabilidade com outras linguagens de programação, como C/C++, Java, C#, .NET e PHP, entre outras. Essa característica permite aos desenvolvedores integrar e utilizar ferramentas ou funcionalidades que não estão implementadas nativamente no *Python*. (SANTIAGO, 2022).

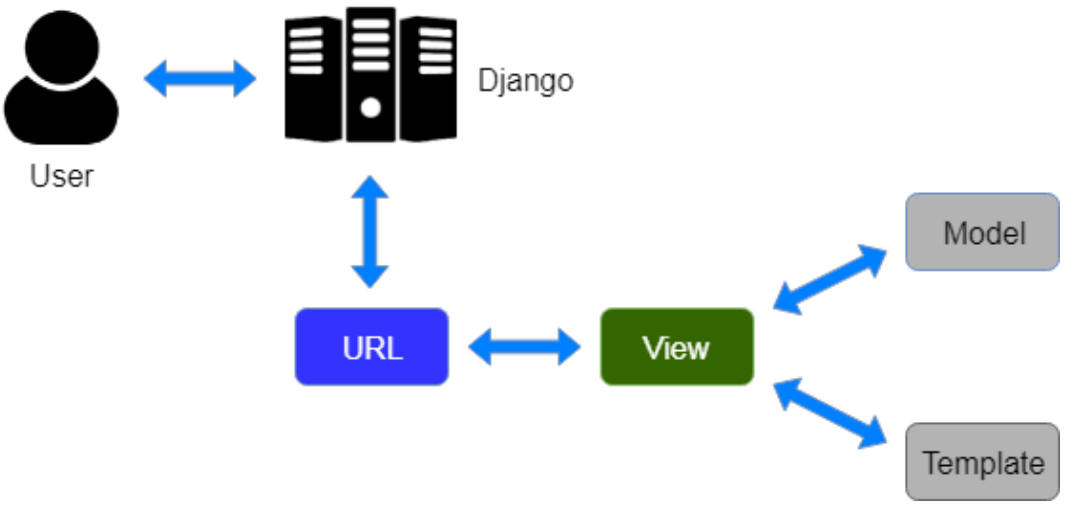
2.2.3. Django

Um *framework* de código aberto para o desenvolvimento de aplicações Web escrito em *Python*, foi criado em 2005 por um grupo de programadores do Lawrence Journal-World. O objetivo principal de sua criação foi acelerar o processo de desenvolvimento de aplicações para a Web. Este *framework* ganhou reconhecimento por oferecer soluções integradas para muitos desafios comuns no desenvolvimento web, disponibilizando diversas funcionalidades prontas para uso, como autenticação de usuários, administração de conteúdo e geração de mapas de sites. (SANTIAGO, 2022).

Django auxilia na prevenção de vulnerabilidades de segurança frequentes e proporciona escalabilidade aos sistemas, permitindo a expansão da capacidade de um sistema sem comprometer seu desempenho, a exemplo do que ocorre com plataformas como Mozilla Firefox, Pinterest e Instagram. A ampla versatilidade do Django o torna uma escolha frequente para diversas empresas, sendo empregado na construção de sistemas que abrangem desde o gerenciamento de conteúdo até plataformas de cunho científico. (TEIXEIRA, 2018).

O Django utiliza a estrutura MTV (*Model-Template-View*), representado na Figura 1 onde o framework gerencia a maior parte da comunicação entre requisições HTTP.

Figura 1 Views e URLs no Django: Fundamentos Para Gerenciar Rotas



Fonte: (ELIAS, 2025).

Quando é criado um projeto em Django temos um diretório inicial de pastas e arquivos, onde estão os arquivos models.py, views.py e urls.py, responsáveis pelo fluxo MTV.

- *Model:* Esta camada lida com a gestão e o acesso aos dados persistidos no banco de dados. Sua função primordial é assegurar a integridade dessas informações, além de prover os mecanismos necessários para sua manipulação e recuperação.
- *Template:* Responsável pela apresentação visual da aplicação web, esta camada define como os dados fornecidos pela camada View são exibidos ao usuário, seguindo um conjunto de diretrizes preestabelecidas. Essencialmente, ela determina a interface com a qual o usuário interage.
- *View:* Incorpora a lógica da aplicação. Sua responsabilidade reside em processar as solicitações dos usuários e gerar a resposta correspondente. Além disso, a *View* determina o curso de ação a ser tomado em função da requisição recebida e interage com a camada Model para obter ou persistir dados conforme a necessidade.

2.3. Versionamento de Código

Git é uma ferramenta de controle de versão operada via linha de comando. Sua principal função é registrar o progresso de um projeto, possibilitando comparar versões, retornar a estados anteriores e examinar as alterações feitas em cada etapa (DEODATO, 2019). Como um sistema de versionamento de código distribuído, o Git permite que cada desenvolvedor possua uma cópia completa do repositório em sua máquina, incluindo todos os arquivos e o histórico de alterações. Essa característica elimina a dependência de um servidor central, possibilitando que todos trabalhem de forma independente, mesmo offline, e realizem operações com rapidez, já que são processadas localmente. Esse modelo facilita o acesso aos arquivos do projeto, o acompanhamento de sua evolução, além da possibilidade de sincronizar com versões mais recentes e propor mudanças.

O GitHub, por sua vez, atua como a plataforma de hospedagem que armazena esses projetos. Ele aprimora o Git ao oferecer uma interface de usuário intuitiva e funcionar como um mecanismo de distribuição para os repositórios. Além disso, o GitHub proporciona controle administrativo, permitindo definir quem tem permissão para visualizar, alterar ou gerenciar um projeto. O GitHub se consolidou como o epicentro do desenvolvimento de software moderno e uma robusta rede profissional. Embora seja frequentemente descrito como uma "rede social para programadores", sua função principal é ser uma plataforma de hospedagem baseada na nuvem para o Git, que é o verdadeiro pilar tecnológico subjacente (DEODATO, 2019).

2.4. Scrum

O gerenciamento de projetos envolve um conjunto de processos, fases e atividades planejadas e executadas para alcançar objetivos específicos dentro de um prazo e orçamento determinados. Tradicionalmente, muitos projetos eram conduzidos por metodologias em cascata, que seguiam um fluxo linear e sequencial. No entanto, com a crescente complexidade e a necessidade de maior adaptabilidade em diversos setores, surgiram as metodologias ágeis.

Metodologias ágeis são abordagens iterativas e incrementais para o gerenciamento de projetos, especialmente em desenvolvimento de software, que enfatizam a flexibilidade, a colaboração do cliente e a capacidade de resposta a mudanças. Elas se contrapõem às metodologias tradicionais por priorizarem a entrega contínua de valor e a adaptação rápida a novos requisitos (PEREIRA 2007).

Nesse contexto, destaca-se o Scrum, um framework para gerenciamento de projetos que segue os princípios das metodologias ágeis. O Scrum é caracterizado por ciclos de desenvolvimento curtos e interativos, chamados sprints, que geralmente duram de uma a quatro semanas. Em cada sprint, uma equipe multidisciplinar trabalha para entregar um incremento funcional do produto. O Scrum define papéis claros (Product Owner, Scrum Master e Time de Desenvolvimento), eventos específicos (Sprint Planning, Daily Scrum, Sprint Review e Sprint Retrospective) e artefatos (Product Backlog, Sprint Backlog e Incremento), que juntos promovem a transparência, inspeção e adaptação contínua do projeto (PEREIRA 2007). Essa estrutura permite que as equipes respondam rapidamente a *feedback* e mudanças, garantindo que o produto final atenda às necessidades do cliente de forma eficaz.

3. METODOLOGIA

O presente trabalho foi desenvolvido com base em uma Pesquisa Aplicada, que busca gerar conhecimentos para aplicação prática na solução de problemas específicos. Essa abordagem se alinha perfeitamente ao projeto, pois o objetivo é desenvolver um sistema web funcional para modularizar páginas e gerenciar narrativas, impactando diretamente a otimização e a manutenção do repositório online.

Para a elaboração deste estudo, foi realizada uma pesquisa bibliográfica abrangente. As principais fontes consultadas para a construção do referencial teórico e para o suporte ao desenvolvimento foram artigos científicos e documentações técnicas disponíveis em sites como o Google Scholar, repositórios de instituições de ensino (como o IFRN) e documentações oficiais de tecnologias como Django, Python, HTML e CSS.

Toda a elaboração e o desenvolvimento do projeto seguiram o processo Scrum. As atividades foram divididas em sprints, com os primeiros dedicados à documentação e planejamento do sistema. Nos sprints subsequentes, o repositório online foi desenvolvido de forma incremental, com foco na implementação das funcionalidades. Esta metodologia ágil permitiu flexibilidade na adaptação às mudanças e entregas contínuas, garantindo que o projeto evoluísse de acordo com as necessidades identificadas.

4. RESULTADOS

O e-Narrativas, desenvolvido em Python, foi criado para facilitar o registro de narrativas dos municípios brasileiros. A proposta é elaborar um sistema de cadastro acessível via página web, permitindo que qualquer pessoa consulte as informações. O projeto é uma colaboração de alunos do IFRN - Campus Canguaretama, abrangendo os cursos Superior de Tecnologia em Sistemas para Internet e Técnico Integrado de Nível Médio em Informática.

O presente trabalho tem foco na contribuição do autor para o projeto, que incluiu a modularização das páginas de tela inicial e cadastro de narrativas que estão dentro dos templates, além de parte da implementação das operações CRUD, que é a sigla em inglês para operações básicas de persistência de dados em sistema: Create (Criar), Read (Ler), Update (Atualizar), e Delete (Excluir). As responsabilidades dos demais pesquisadores abrangeram o layout do sistema, a implementação das operações básicas de listagem, busca e filtragem de narrativas, e o georreferenciamento das mesmas.

Essas operações são onipresentes em aplicações de banco de dados e em uma vasta gama de sistemas. A implementação dessas funcionalidades CRUD em um projeto permite a listagem, o registro, a edição e a remoção de cada item.

4.1. Modularização de Layout

A identidade visual do Portfólio online foi concebida por outro pesquisador do projeto. Partes do site, como menu, cabeçalho e rodapé, são elementos recorrentes em todas as páginas, e essa repetição é um ponto central no processo de modularização do site.

A modularização envolve a separação do código que se repete, permitindo que ele seja aproveitado em todo o projeto. Isso otimiza a manutenção, pois uma alteração no menu, por exemplo, exige modificação em apenas um arquivo, em vez de todas as páginas. Adicionalmente, esse método alinha-se a um dos princípios da programação, que é a prevenção da repetição de código.

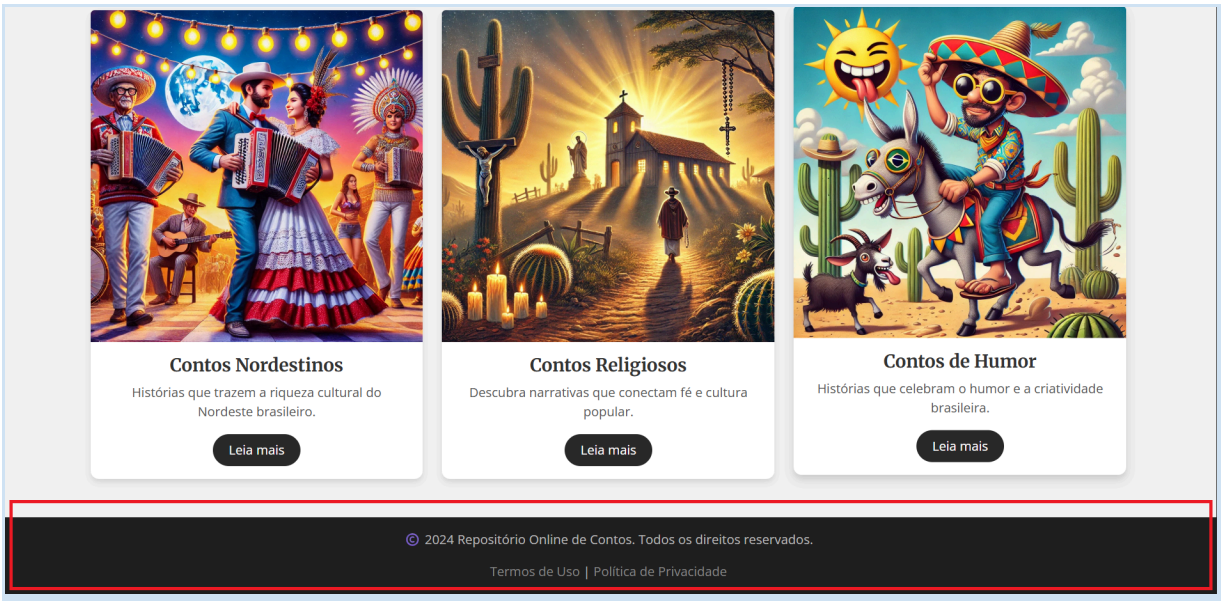
A primeira seção identificada com repetição em todas as páginas é o cabeçalho e o menu, conforme assinalado em vermelho na Figura 2.

Figura 2. Parte da tela Inicial do site com área que se repete marcada em vermelho

Fonte: Do próprio Autor (2025)

A outra área detectada que se repete nas páginas é o rodapé, também marcada de vermelho na Figura 3.

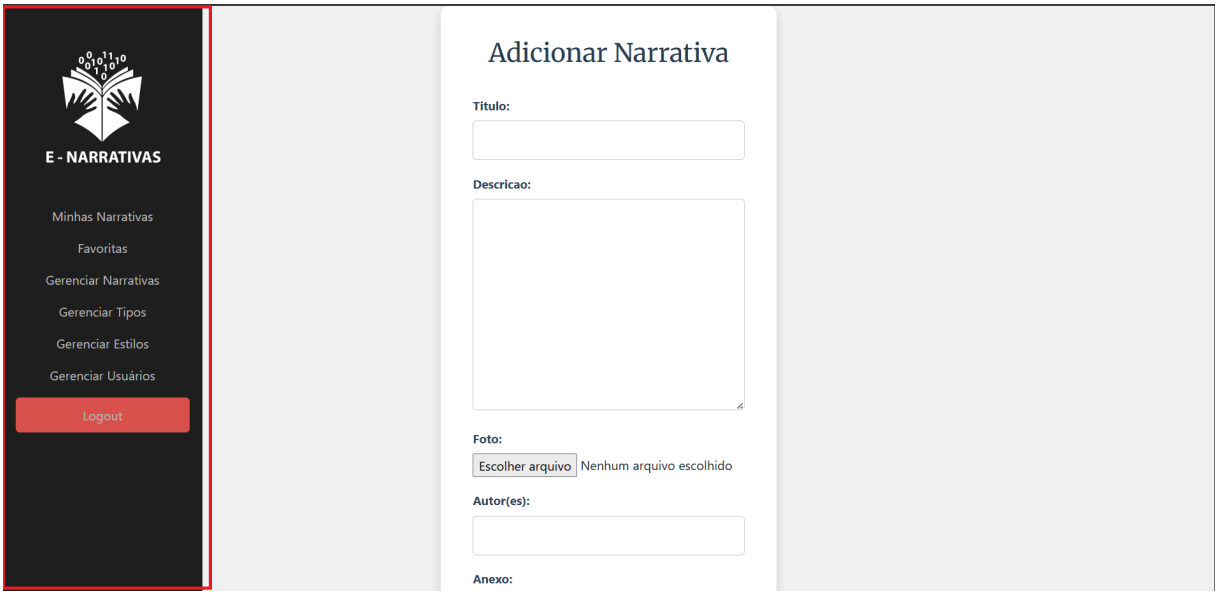
Figura 3. Parte da tela Inicial do site com área que se repete marcada em vermelho



Fonte: Do próprio Autor (2025)

Outra área recorrente, destacada em vermelho na Figura 4, é o menu lateral presente nas páginas de cadastro.

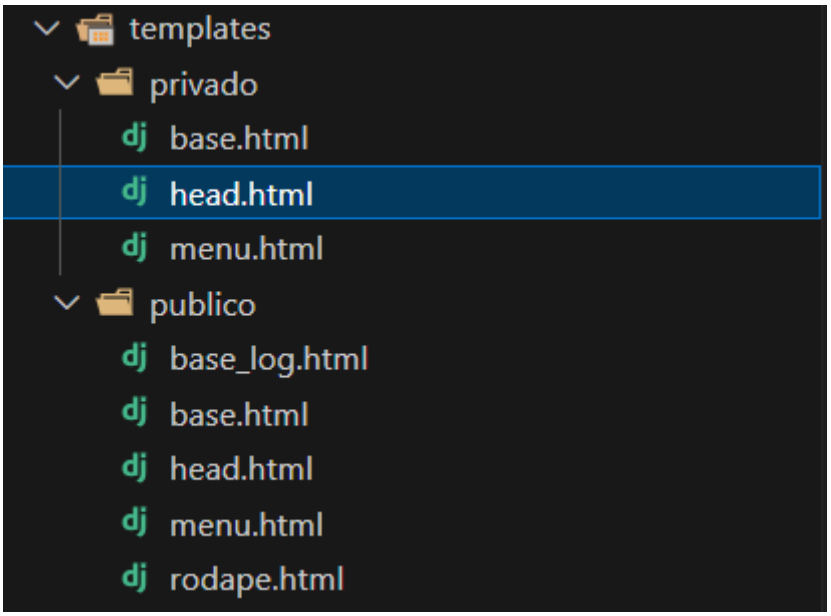
Figura 4. Parte da tela de cadastro das narrativas com área que se repete marcada em vermelho.



Fonte: Do próprio Autor (2025)

Para alcançar a modularização do projeto, adotou-se a abordagem recomendada pelo próprio framework Django. Uma pasta dedicada, nomeada templates, foi estabelecida na raiz do diretório do projeto como mostra na Figura 5.

Figura 5. Pasta templates do projeto Django



Fonte: Do próprio Autor (2025)

A pasta templates serve como o principal repositório para a organização e gerenciamento dos arquivos essenciais à modularização, permitindo uma estrutura de projeto mais limpa, escalável e de fácil manutenção. Esta metodologia garante que componentes distintos possam ser desenvolvidos e geridos de forma independente, otimizando o fluxo de trabalho e a colaboração entre desenvolvedores.

Dentro do diretório de templates, encontram-se duas subpastas cruciais para a estrutura e funcionalidade do projeto. A primeira subpasta, nomeada "privado", é dedicada a conter os códigos que se repetem especificamente nas páginas de perfil e gerenciamento dos dados do site, que são acessadas apenas por usuários autenticados. Esta distinção é fundamental para organizar o código de forma lógica e segura.

Em contraste, a segunda, denominada "publico", abriga todos os arquivos que contêm os códigos reutilizáveis e elementos de design que se repetem em diversas páginas acessíveis ao público geral do projeto. Isso inclui, por exemplo, cabeçalhos,

rodapés, menus de navegação, estilos CSS globais e scripts JavaScript comuns, garantindo uma consistência visual e funcional em toda a interface do usuário.

4.2. Banco de dados

O ORM (Object-Relational Mapper - Mapeamento Objeto-Relacional) do Django é essencial para desenvolvedores, permitindo interagir com bancos de dados usando o próprio código Python em vez de SQL. Ele simplifica a manipulação de dados e acelera o desenvolvimento. O ORM é integrado aos Models do Django, onde cada tabela do banco de dados é uma classe Python e seus atributos correspondem às colunas. Isso permite operações como criar, listar, atualizar e excluir registros de forma intuitiva, além de consultas complexas. O ORM melhora a produtividade e a portabilidade do código, sendo compatível com diversos sistemas de banco de dados sem exigir alterações no código da aplicação. Em resumo, o ORM do Django é uma ferramenta que abstrai a complexidade do banco de dados, focando na lógica de negócio.

O banco de dados do Repositório online foi elaborado e desenvolvido por outro colaborador do projeto. Contudo, foi necessário algumas alterações. O campo de imagem estava como obrigatório, mas é um recurso opcional e flexível que permite ao usuário decidir se deseja carregar uma imagem. Se nenhuma imagem for fornecida, o sistema preencherá automaticamente o campo com uma imagem padrão para manter a consistência visual e a integridade da interface. Essa lógica foi alterada e é demonstrada na Figura 6, linhas 30 e 31 do código.

Figura 6. Código do banco alterado

```
16  def atualizar_nome_foto(instance, filename):
17      path = "narrativas/"+instance.slug+"/anexo_"
18      format = instance.slug + '.' + filename.split('.')[-1]
19      return os.path.join(path, format)
20
21  def atualizar_nome_anexo(instance, filename):
22      path = "narrativas/"+instance.slug+"/anexo_"
23      format = 'anexo_' + instance.slug + '.' + filename.split('.')[-1]
24      return os.path.join(path, format)
25
26
27  class Narrativa(models.Model):
28      titulo = models.CharField('Titulo', max_length=300)
29      descricao = models.TextField('Descricao')
30      foto = models.ImageField('Foto', upload_to=atualizar_nome_foto,
31                               default='narrativas/default/default.png', null=True, blank=True)
32      autor = models.CharField('Autor(es)', max_length=200)
33      anexo = models.FileField('Anexo', upload_to=atualizar_nome_anexo, null=True, blank=True)
34      link = models.CharField('Link', max_length=500, null=True, blank=True)
35      slug = models.CharField(max_length=15, unique=True)
36      tipo_narrativa = models.ForeignKey(Tipo_Narrativa, on_delete=models.PROTECT)
37      estilo_narrativa = models.ManyToManyField(Estilo_Narrativa)
38      def __str__(self):
39          return self.titulo
```

Fonte: Do próprio Autor (2025)

Foi acrescentado o campo slug conforme mostra na Figura 6, linha 35 do código que é uma String que identifica exclusivamente um objeto, geralmente usado em URLs para torná-las mais amigáveis ao usuário e aos mecanismos de busca, um recurso específico, geralmente um artigo, uma página, um produto, etc.

É criado a partir de um título (por exemplo, "Mula sem cabeça"), que é convertido para letras minúsculas, com espaços e caracteres especiais substituídos por hífens (resultando em "mula-sem-cabeça"). Um exemplo de sua aplicação pode ser visto em: https://enarrativas.nic.ifrn.edu.br/narrativas/mula-sem-cabeca, onde "mula-sem-cabeca" representa o slug.

As linhas 16 e 21 da Figura 6, são funções que foram implementadas para gerenciar o nome dos arquivos ao realizar o upload, especificamente para determinar onde um arquivo será salvo no sistema de arquivos do servidor e qual será o nome desse arquivo. Por exemplo, uma imagem seria 'mula-sem-cabeça.jpg' e um arquivo seria 'mula-sem-cabeça.pdf'. Seu objetivo principal é gerar um caminho completo e um nome de arquivo único e organizado para a foto que está sendo carregada.

4.3 Desenvolvimento

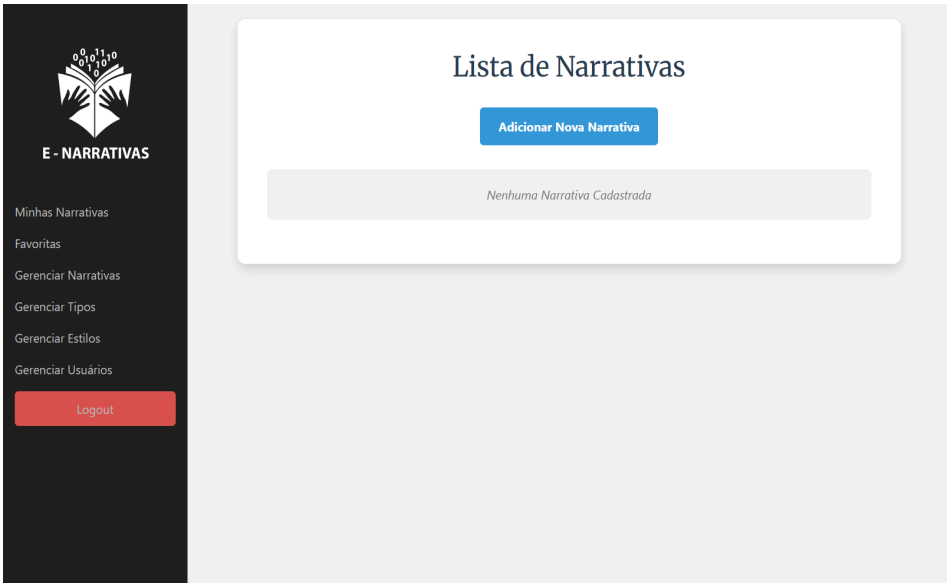
No desenvolvimento deste trabalho, fui responsável pela modularização do layout do repositório, pela implementação das correções no banco de dados e pela garantia do funcionamento das operações básicas para o gerenciamento de narrativas.

Foi feita a modularização de duas telas, a tela inicial e a tela de cadastro das narrativas, no qual também fui o responsável pelo CRUD da mesma.

4.3.1 Narrativa

O CRUD implementado foi correspondente a tabela do banco de dados Narrativa. A primeira página que o administrador do sistema acessa será a de listagem, mostrada na Figura 7.

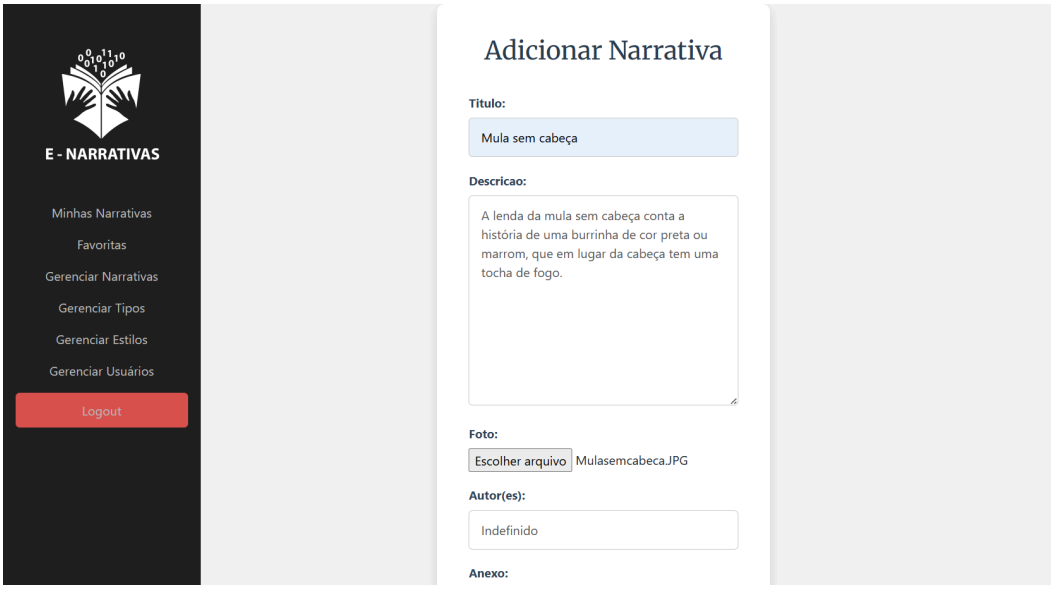
Figura 7. Tela de Listagem Narrativas.



Fonte: Do próprio Autor (2025)

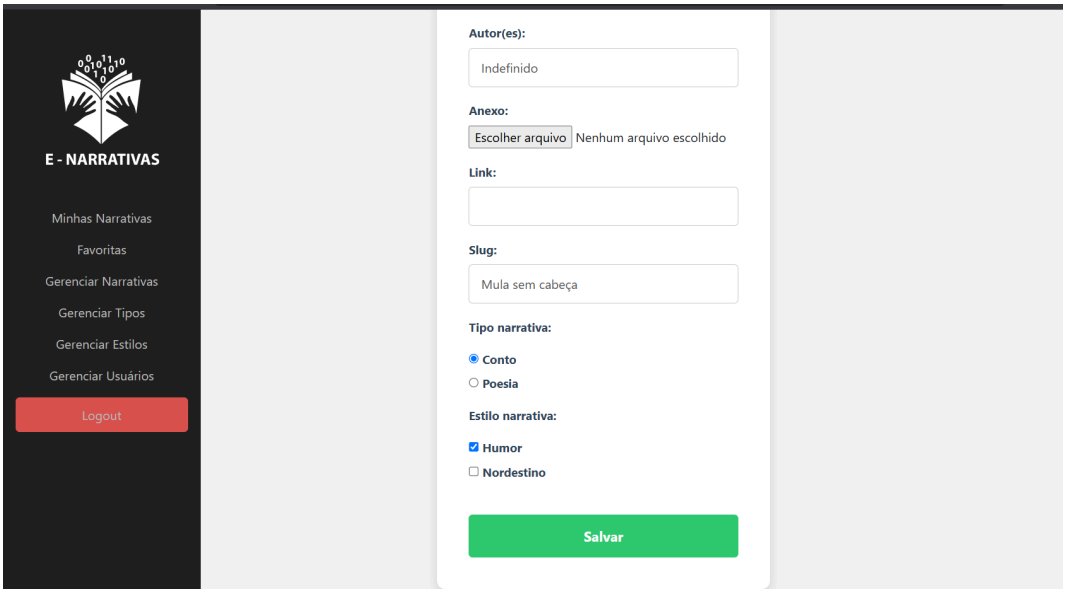
Na Figura 7 as narrativas serão listadas e também permite o cadastro através do botão Adicionar Nova Narrativa. Ao clicar no botão Adicionar Nova Narrativa, o usuário será redirecionado para a página da Figura 8.

Figura 8. Tela de cadastro das Narrativas, parte 1.



Fonte: Do próprio Autor (2025)

Figura 9. Tela de cadastro das Narrativas, parte 2.



Fonte: Do próprio Autor (2025)

Na tela de cadastro das Narrativas (Figura 8 e 9) é possível inserir o título da narrativa, descrição, foto, autor, anexo, link, slug, tipo e estilo da narrativa. O tipo e estilo da narrativa foram CRUDS realizador por outro autor que contribuiu no projeto e que podem ser cadastrados através do menu lateral antes de cadastrar a narrativa.

No exemplo em questão está realizando o cadastro da narrativa Mula sem cabeça. Após clicar no botão Salvar, a informação digitada será salva no banco de dados e o usuário será redirecionado para a tela de listagem novamente, conforme mostra a Figura 10.

Figura 10. Tela de Listagem das Narrativas.

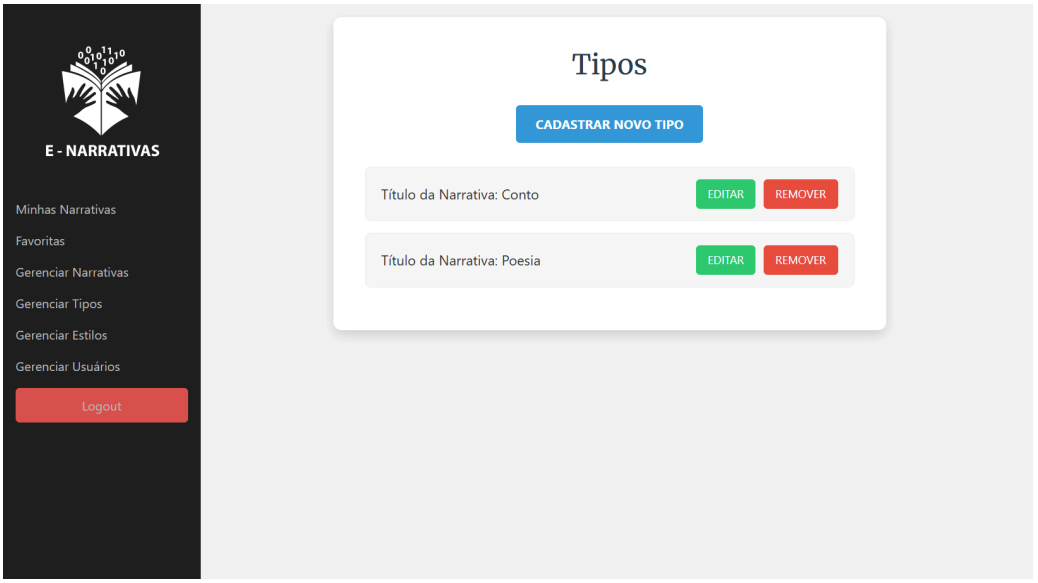
Fonte: Do próprio Autor (2025)

Na tela de listagem das Narrativas (Figura 10), agora é possível visualizar que Mula sem cabeça foi cadastrada. Além disso, é possível editar a narrativa clicando no botão Editar ou apagar do banco de dados clicando no botão Remover.

4.3.2 Layout Tipo_Narrativa

A funcionalidade CRUD para os tipos de narrativa foi modularizada, resultando na implementação de um layout, como demonstrado na Figura 11. Anteriormente, essa seção exibia apenas uma tela em branco com os tipos de narrativa.

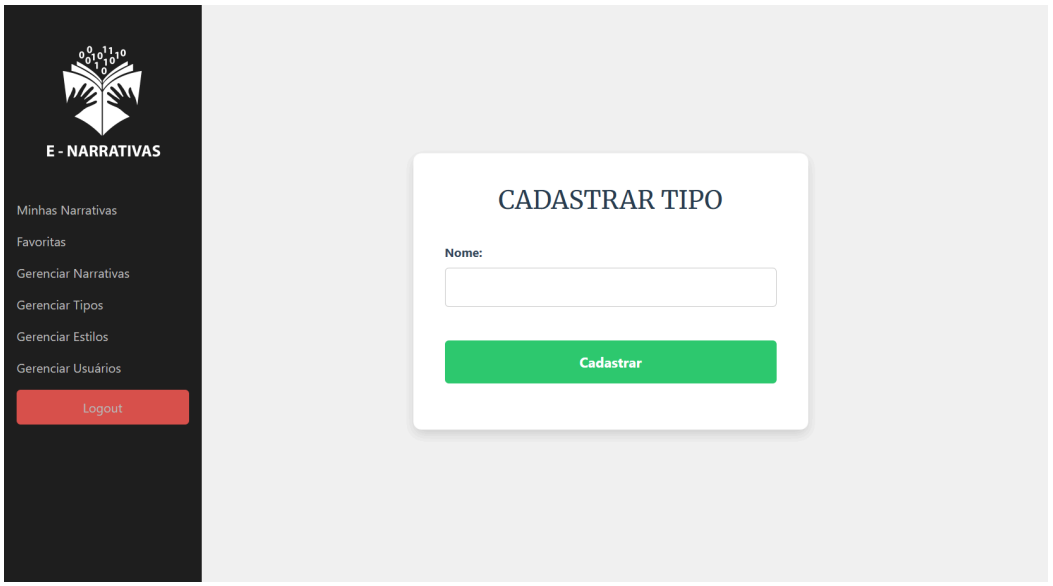
Figura 11. Tela de Listagem dos Tipos de Narrativas.



Fonte: Do próprio Autor (2025)

Na figura 11 exibe os Tipos de narrativas que já estão cadastrados e também permite o cadastro de um novo Tipo ao clicar no botão CADASTRAR NOVO TIPO.

Figura 12. Tela de Cadastro de Tipo de Narrativa.



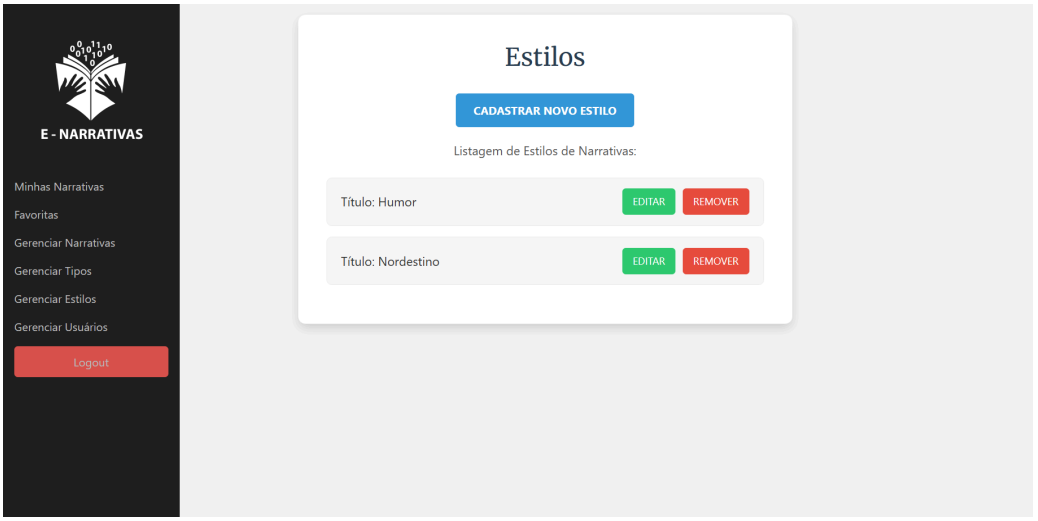
Fonte: Do próprio Autor (2025)

A Tela de Cadastro de Tipo de Narrativa (Figura 12) permite a inserção do nome do Tipo. Após o clique no botão "Cadastrar", a informação digitada é salva no banco de dados e o usuário é redirecionado para a tela de listagem.

4.3.3 Layout Estilo_Narrativa

O layout para a funcionalidade CRUD dos estilos de narrativa foi implementado e modularizado. Anteriormente, essa seção exibia apenas uma tela em branco com os estilos.

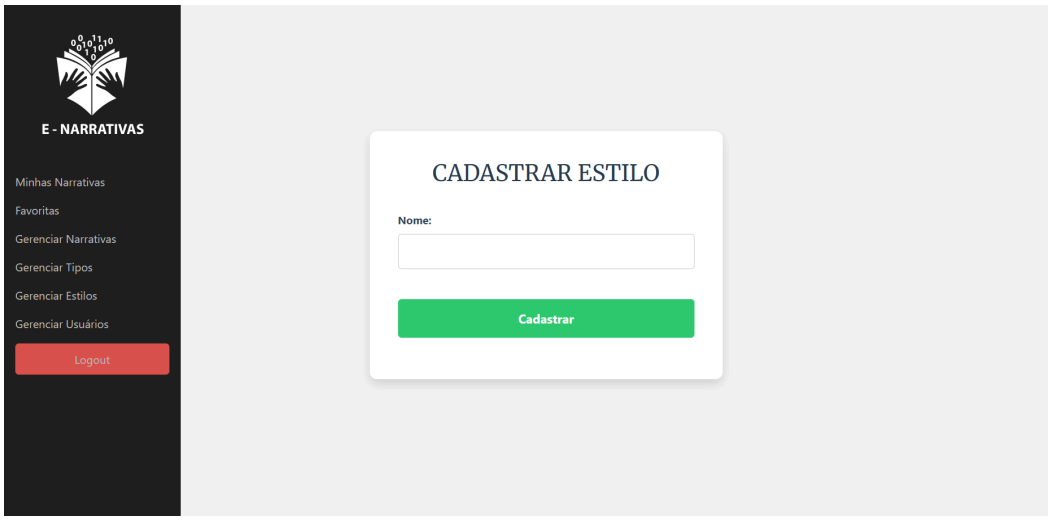
Figura 13. Tela de Listagem de Estilo.



Fonte: Do próprio Autor (2025)

A Figura 13 mostra os estilos de narrativa já registrados e oferece a opção de cadastrar um novo estilo clicando no botão "CADASTRAR NOVO ESTILO".

Figura 14. Tela de Cadastro de Estilo.



Fonte: Do próprio Autor (2025)

Ao clicar no botão "CADASTRAR NOVO ESTILO" para inserir um novo estilo, o usuário será redirecionado para a tela de cadastro (Figura 14), permitindo inserir o nome do estilo desejado e, posteriormente, inserir no banco. Em seguida, após cadastrado, o usuário será redirecionado novamente para a página de listagem

5. CONCLUSÃO

Este trabalho alcançou seu objetivo geral de modularizar as páginas de tela inicial e cadastro de narrativas, otimizando a manutenção e reutilização do código no repositório on-line de narrativas. As modificações no banco de dados, como a correção do campo de imagem e a adição do campo slug, juntamente com a implementação das operações CRUD, garantiram a funcionalidade básica e a eficiência do sistema.

Para o desenvolvimento deste projeto, diversas tecnologias foram cruciais. Python, com sua sintaxe clara e vastas bibliotecas, facilitou a implementação da lógica do sistema. O framework Django, por sua vez, agilizou o desenvolvimento web ao fornecer uma estrutura robusta e funcionalidades prontas para uso, como o ORM, que simplificou a interação com o banco de dados. HTML e CSS foram fundamentais para a estruturação e estilização das páginas, respectivamente, permitindo a criação de uma interface intuitiva e responsiva. O uso de Git e GitHub foi essencial para o controle de versão, permitindo a colaboração eficiente e o rastreamento das alterações no código. A metodologia Scrum guiou o processo de desenvolvimento, proporcionando flexibilidade e entregas incrementais.

Como trabalhos futuros, sugere-se expandir as funcionalidades do sistema e-Narrativas, como a implementação de um sistema de busca mais avançado com filtros adicionais, aprimorar a interface do usuário com elementos mais interativos e dinâmicos, e explorar a possibilidade de integrar o repositório com outras plataformas, como o SUAP (Sistema Unificado de Administração Pública) para compartilhamento de dados ou APIs (em inglês, Application Programming Interface) de georreferenciamento mais robustas para visualização das narrativas em mapas interativos.

REFERÊNCIAS

ANDRADE, Francisco Adam et al. Versionamento de Projeto na Prática com Git e GitHub: Um Relato de Experiência do Curso Ofertado pelo Projeto LearningLab No Interior Cearense. In: Workshop sobre Educação em Computação (WEI). SBC, 2024. p. 183-194.

CANDIDO, Antonio. Formação da literatura brasileira. São Paulo: Martins, 1964.

CASTELLO, José Aderaldo. A literatura brasileira: origens e unidade (1500-1960). Edusp, 1999.

CORREIA, Jacinta. Lendas e Contos Tradicionais. 2012.

COSTA, Carlos J. Desenvolvimento para web. ITML press/Lusocredito, 2007.

DEODATO, Felipe Augusto de Oliveira; BARIONI, Gabriel Paz da Silva. Indexação e recuperação de recursos educacionais abertos em repositórios suportados por licenças creative commons e versionamento de código. 2019.

ELIAS. Views e URLs no Django: Fundamentos Para Gerenciar Rotas. Disponível em: <https://prosimples.com/views-e-urls-no-Django-fundamentos-para-gerenciar-rotas/>. Acesso em: 1 jun. 2025.

KUROSE, James; ROSS, Keith. Redes de computadores e Internet. São Paulo: Person, 2006.

PEREIRA, Paulo; TORREÃO, Paula; MARÇAL, Ana Sofia. Entendendo Scrum para gerenciar projetos de forma ágil. Mundo PM, v. 1, n. 14, p. 64-71, 2007.

RABELO, Amanda Oliveira. A importância da investigação narrativa na educação. **Educação & Sociedade**, v. 32, p. 171-188, 2011.

SANTIAGO, Cynthia Pinheiro et al. Desenvolvimento de sistemas Web orientado a reuso com Python, Django e Bootstrap. Sociedade Brasileira de Computação, v. 25, 2020.

SILVA, Maurício Samy. Criando sites com HTML: sites de alta qualidade com HTML e CSS. Novatec Editora, 2008.

TEIXEIRA, Adriano Dias. Aplicações web seguras em Django. 2018. Tese de Doutorado. Disponível em:<https://repositorium.sdum.uminho.pt/handle/1822/79825 > Acesso em: 01 de jun. de 2025.

ANEXOS

ANEXO 1 - Link para o repositório GITHUB do projeto:

https://github.com/brunogomesifrn/pp23narrativas