

CENTRO UNIVERSITÁRIO DE BRASÍLIA (CEUB)
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
DISCIPLINA DE PROJETO INTEGRADOR II

22309255 Victor Albuquerque
22301685 Gabriel Paiva

**FERRAMENTA PARA O COMBATE DE FAKE NEWS NA POLÍTICA
BRASILEIRA**

Brasília, Setembro 2025

SUMÁRIO

| | |
|--------------------------------------------------------------------------------|-----------|
| 1. Introdução | 3 |
| 1.1 Objetivo do Projeto | 4 |
| 1.2 Escopo | 4 |
| 1.3 Público-Alvo | 4 |
| 1.4 Stakeholders | 5 |
| 2. Requisitos do Sistema | 6 |
| 2.1 Requisitos Funcionais | 6 |
| 2.2 Requisitos Não Funcionais | 7 |
| 3. Arquitetura e Tecnologias | 8 |
| 3.1 Diagrama de Arquitetura | 8 |
| 3.2 Tecnologias Utilizadas | 8 |
| 4. Telas e Design | 9 |
| 4.1 Protótipo de Telas | 9 |
| 4.2 Padrões de Design | 11 |
| 4.2.1 Visibilidade do status do sistema | 11 |
| 4.2.2 Correspondência entre o sistema e o mundo real | 11 |
| 4.2.3 Liberdade de controle para o usuário | 12 |
| 4.2.4 Consistência e padrões | 12 |
| 4.2.5 Prevenção de erros | 12 |
| 4.2.6 Reconhecimento em vez de memorização | 13 |
| 4.2.7 Flexibilidade e eficiência de uso | 13 |
| 4.2.8 Estética e design minimalista | 13 |
| 4.2.9 Ajudar usuários a reconhecerem, diagnosticarem e recuperarem-se de erros | 13 |
| 4.2.10 Ajuda e documentação | 14 |
| 5. Modelagem do Sistema | 15 |
| 5.1 Diagrama de Casos de Uso | 15 |
| 5.2 Diagrama de Classes | 16 |
| 5.3 Diagrama de Sequência | 17 |
| 5.4 Diagrama Entidade-Relacionamento | 18 |
| 6. Desenvolvimento e Metodologia | 19 |
| 6.1 Metodologia de Desenvolvimento | 19 |
| 6.2 Repositório de Código-Fonte | 19 |
| 6.3 Estrutura do Código | 19 |
| 7. Gestão de Tarefas e Kanban | 21 |
| 7.1 Uso do Kanban | 21 |
| 7.2 Quadro Kanban | 21 |

| | | |
|------------|--------------------------------------|-----------|
| 7.2.1 | 21 de agosto | 21 |
| 7.2.2 | 17 de setembro | 21 |
| 8. | Testes e Validação | 22 |
| 8.1 | Estratégia de Testes | 22 |
| 8.2 | Ferramentas de Teste | 22 |
| 9. | Implantação e Manutenção | 23 |
| 9.1 | Processo de Implantação | 23 |
| 9.2 | Plano de Manutenção | 23 |
| 9.3 | Versão Executável ou Deploy na Nuvem | 24 |
| 9.4 | Documentação Técnica | 24 |
| 9.5 | Manual do Usuário/Treinamento | 24 |
| 10. | Conclusão | 25 |
| 11. | Anexos | 26 |
| 12. | Referências | 27 |

1. Introdução

O amplo acesso à tecnologia, junto às redes sociais, fez com que notícias falsas fossem disseminadas com baixo custo e ampla distribuição (Barcelos et al., 2021). Durante as eleições presidenciais dos Estados Unidos em 2016, foram registradas 156 notícias falsas, as quais foram compartilhadas um total de 37 milhões de vezes, influenciando no resultado da eleição (Allcott, Hunt, and Matthew Gentzkow. 2017). Em 2018, o Brasil já estava entre os países com maior circulação de fake news no mundo. Durante a pandemia esse fenômeno se intensificou (Barcelos et al., 2021). Foram identificadas 329 notícias falsas sobre a COVID-19, sendo 20% relacionados à política, dificultando campanhas de vacinação e tornando uma preocupação a saúde pública (Barcelos et al., 2021). O conceito de *fake news* ainda não é bem definido, mas um dos mais amplamente utilizados em estudos é que *fake news* são conteúdos falsos produzidos e compartilhados como verdadeiros (Shu K et al., 2017).

Com o crescimento exponencial da tecnologia, surgiram as inteligências artificiais generativas, como o *ChatGPT* e *DeepSeek*. Assim como ocorre com as redes sociais, qualquer pessoa com um celular e acesso à internet agora pode utilizar esses produtos. As inteligências artificiais generativas utilizam o aprendizado de máquina para analisar grandes volumes de dados e gerar conteúdos como textos, imagens e áudios (Jovanovic, M., & Campbell, M., 2022). A partir disso, conteúdos falsos são mais fáceis de serem produzidos. No entanto, é possível utilizar-se do aprendizado de máquina para detectar as fake news (Alghamdi, J., Luo, S. & Lin, Y., 2023).

Uma pesquisa da Avaaz (2020) com 2001 brasileiros entre 18 e 65 anos evidenciou a relevância social da problemática abordada neste estudo. Os resultados indicaram que 80% dos respondentes manifestaram interesse em receber correções provenientes de verificadores de fatos quando expostos a conteúdos potencialmente falsos. Esta expressiva proporção sugere uma demanda social por ferramentas de verificação e constitui um indicador relevante da percepção pública quanto à necessidade de mecanismos eficazes para o enfrentamento da desinformação no contexto nacional. A identificação desta necessidade corrobora a pertinência da presente investigação, que visa contribuir para a identificação das melhores soluções tecnológicas aplicáveis ao problema identificado.

Dessa forma, a proposta deste projeto justifica-se pela crescente necessidade de mecanismos tecnológicos capazes de auxiliar na identificação e no combate à desinformação em larga escala. A utilização de técnicas de inteligência artificial para a detecção de fake news apresenta-se como uma solução inovadora e eficiente, pois combina análise automatizada, precisão estatística e capacidade de aprendizado

contínuo. Assim, espera-se que a aplicação desenvolvida contribua não apenas para a redução da propagação de conteúdos enganosos, mas também para o fortalecimento da confiabilidade das informações disponíveis no meio digital.

1.1 Objetivo do Projeto

Solução para combater disseminação de fake news políticas. Baseado em inteligência artificial servirá como uma ferramenta de verificação e análise das informações que circulam no ambiente digital. Seu propósito central é criar uma ferramenta digital contra a disseminação de conteúdos falsos ou enganosos que podem influenciar negativamente a percepção pública sobre temas relevantes.

1.2 Escopo

O sistema de detecção de fake news consiste em uma aplicação web, que permite aos usuários enviar textos para análise automatizada de veracidade. O projeto implementa uma solução baseada em inteligência artificial para examinar e classificar o texto submetido, fornecendo uma avaliação quanto à probabilidade de o conteúdo constituir uma notícia falsa ou enganosa, também faz uma busca na internet por fontes confiáveis sobre a notícia e retorna ao usuário. As funcionalidades principais são a autenticação de usuários, campo de envio do texto, processamento do conteúdo, relatório do processamento do conteúdo para o usuário e histórico de análises anteriores pedida pelo usuário.

Além disso o sistema possuirá algumas limitações, sendo elas o formato de entrada que inicialmente serão apenas textos, o idioma suportado será o português brasileiro, existirá um limite de caracteres por análise e um limite diário de análises a fim de garantir a performance do sistema.

1.3 Público-Alvo

O sistema de detecção de fake news é direcionado a diversos perfis de usuários que compartilham a necessidade de validar a autenticidade de informações. Este grupo inclui cidadãos comuns preocupados com a veracidade das notícias que consomem diariamente; profissionais de comunicação como jornalistas, editores e assessores de imprensa que necessitam verificar conteúdos antes de sua divulgação; educadores e estudantes engajados em pesquisas e trabalhos acadêmicos que necessitam de fontes confiáveis; profissionais de marketing e relações públicas que precisam monitorar menções a suas marcas ou clientes.

Todos estes usuários compartilham das mesmas necessidades: obter análises rápidas e precisas sobre a confiabilidade de conteúdos noticiosos, receber orientações claras sobre potenciais sinais de desinformação, acessar uma ferramenta intuitiva que não exija conhecimentos técnicos avançados e contar com um recurso que os auxilie nessa detecção de notícias falsas.

1.4 Stakeholders

Para garantir que a aplicação de detecção de fake news seja não apenas precisa, mas também relevante e utilizável, foi definido um grupo estratégico de stakeholders. Este grupo representa três pilares essenciais para o sucesso do projeto: a supervisão acadêmica e técnica, a validação por especialistas da área de comunicação e a avaliação da experiência do usuário final.

O primeiro pilar será representado pela Professora Kadidja Valeria, [orientadora do projeto/especialista na área da ciência da informação]. Sua participação é fundamental para validar a metodologia aplicada, a robustez da implementação técnica e o alinhamento dos resultados com os objetivos acadêmicos e científicos propostos.

O segundo pilar é composto pelos Professores de Jornalismo do Centro Universitário de Brasília (CEUB). Como especialistas no domínio do problema, eles avaliarão a eficácia da aplicação sob a ótica profissional. A sua validação focará na precisão dos critérios utilizados para a detecção, na capacidade da ferramenta em identificar diferentes nuances de desinformação e na relevância dos resultados para o combate prático às fake news. O feedback deste grupo garantirá que a ferramenta tenha credibilidade e aplicabilidade no cenário jornalístico atual.

Por fim, o terceiro pilar de validação será formado pelos alunos do curso de Jornalismo do CEUB. Eles representam o público-alvo e os futuros profissionais que poderão utilizar a ferramenta. A validação com os estudantes terá como foco principal a usabilidade, a interface (UI) e a experiência do usuário (UX). Serão coletadas percepções sobre a facilidade de uso, a clareza das informações apresentadas e a utilidade da aplicação no fluxo de trabalho de verificação de fatos e consumo de notícias. O envolvimento dos alunos é crucial para refinar a aplicação e assegurar sua adoção prática.

2. Requisitos do Sistema

2.1 Requisitos Funcionais

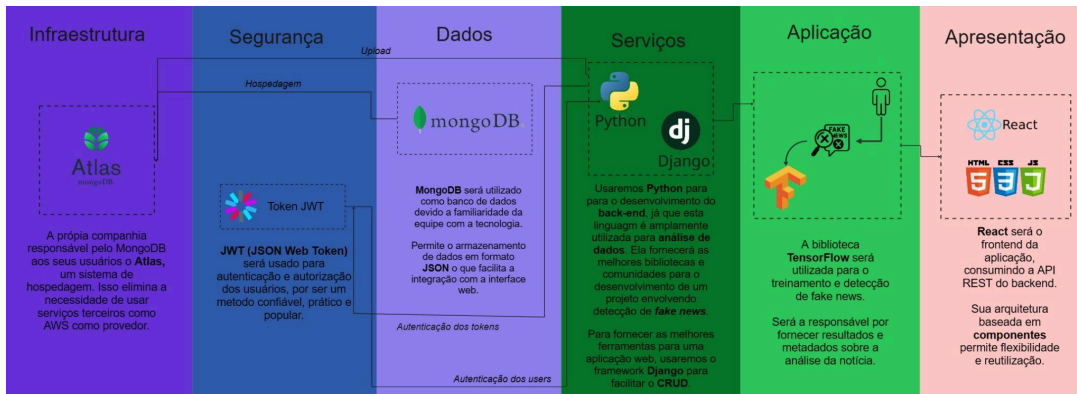
- **Login:** Permite que o usuário acesse o sistema por meio de suas credenciais (e-mail e senha), garantindo segurança no acesso às funcionalidades do sistema. Utiliza autenticação JWT para proteger os dados.
- **Logout:** Encerra a sessão ativa do usuário, garantindo a segurança das informações e impedindo o uso não autorizado da conta.
- **Cadastrar Usuário:** O usuário pode criar uma conta na plataforma, inserindo informações básicas como nome, e-mail e senha para autenticação futura.
- **Enviar Texto para Análise:** Após login, o usuário pode submeter textos para análise de veracidade. O sistema processa a entrada com os modelos de IA para determinar se o conteúdo é fake news.
- **Receber Relatório da Análise:** Após o processamento, o sistema apresenta ao usuário um relatório detalhado contendo a probabilidade de o texto ser fake news, além de grau de confiança e informações adicionais da análise.
- **Consultar Histórico de Análises:** O sistema permite que o usuário visualize as análises realizadas anteriormente, facilitando o acompanhamento de conteúdos já verificados.
- **Visualizar Metadados da Análise:** Fornece dados adicionais sobre a análise realizada, como tempo de processamento, fontes de dados consultadas e justificativa da classificação, garantindo transparência.
- **Gerenciar Conta do Usuário:** O usuário pode acessar e modificar suas informações pessoais cadastradas, como e-mail e senha.
- **Cancelar Análise em Andamento:** Caso o usuário deseje, ele pode interromper uma análise que esteja em execução, oferecendo maior controle sobre as operações realizadas.
- **Receber Mensagens de Erro e Ajuda:** O sistema é projetado para fornecer orientações claras quando ocorrem erros, ajudando o usuário a diagnosticar e resolver o problema.
- **Anexar Arquivo para Análise:** Além da inserção direta do texto, o usuário também pode anexar arquivos que contenham o conteúdo para ser analisado pelo sistema.

2.2 Requisitos Não Funcionais

O sistema deve apresentar alto desempenho, processando análises em tempo real com baixa latência. Em termos de segurança, deve-se implementar criptografia de dados. Quanto à compatibilidade, o sistema precisa funcionar perfeitamente em diferentes navegadores (Chrome, Firefox, Safari, Edge) e dispositivos (desktop, mobile, tablet), com design responsivo. Será necessário também salvar os dados dos usuários e seus chats. A usabilidade deve ser intuitiva, permitindo que usuários sem conhecimento técnico em machine learning compreendam facilmente os resultados da análise junto com explicações claras sobre o funcionamento algorítmico, garantindo transparência no processo de detecção. O sistema deve buscar por fontes confiáveis sobre a notícia enviada pelo usuário e se encontrar deve retornar ao usuário.

3. Arquitetura e Tecnologias

3.1 Diagrama de Arquitetura



3.2 Tecnologias Utilizadas

- **Python:** Linguagem principal do back-end, escolhida pela sua robustez e forte comunidade em projetos de análise de dados. Permite integração eficiente com bibliotecas de machine learning, como TensorFlow.
- **Django:** Framework web de alto nível para Python, usado para estruturar o back-end da aplicação. Facilita a criação de APIs RESTful e operações de CRUD (criação, leitura, atualização e exclusão de dados).
- **HTML / CSS / JS:** Tecnologias base para o desenvolvimento da interface web. HTML estrutura o conteúdo, CSS estiliza a página e JavaScript adiciona interatividade.
- **React:** Biblioteca JavaScript para construção do front-end. Sua arquitetura baseada em componentes permite flexibilidade, reutilização de código e criação de interfaces rápidas e responsivas.
- **JWT (JSON Web Token):** Utilizado para autenticação e autorização dos usuários de forma segura, prática e confiável. É uma solução amplamente adotada para aplicações web modernas.
- **TensorFlow:** Biblioteca de machine learning que será usada para treinar modelos de detecção de fake news. Também será responsável por fornecer análises e resultados sobre as notícias processadas.
- **MongoDB:** Banco de dados NoSQL, escolhido por armazenar dados em formato JSON, o que facilita a integração com aplicações web.
- **MongoDB Atlas:** Plataforma de hospedagem gerenciada para MongoDB, que elimina a necessidade de servidores próprios, oferecendo escalabilidade e segurança.

4. Telas e Design

4.1 Protótipo de Telas

A seguir serão apresentados protótipos das principais telas do sistema, começando pela página de criação de conta:



Protótipo da tela de criação de conta. O formulário contém os seguintes elementos:

- Ícone de uma lupa com o texto "FAKE NEWS" e um "X" vermelho.
- Título: **Crie sua conta.**
- Campos de entrada: Nome de usuário, E-mail, Senha, Confirmar senha.
- Checkbox: ☐ Aceito os **termos e condições**
- Botão: Criar conta
- Seletor: ou
- Link: Já tem uma conta? **Faça o Login**

Agora a página de login:



Protótipo da tela de login. O formulário contém os seguintes elementos:

- Ícone de uma lupa com o texto "FAKE NEWS" e um "X" vermelho.
- Título: **Bem-vindo de volta!**
- Campos de entrada: E-mail, Senha.
- Checkbox: ☐ Lembrar de mim
- Botão: Entrar
- Seletor: ou
- Link: **Criar nova conta**

Agora a página de envio de notícia para análise, o usuário pode optar por enviar a notícia para análise por texto, ou anexar um arquivo. Além disso, à esquerda ele pode escolher um chat antigo para visualização, ou selecionar criar um novo chat. E no canto superior direito é possível acessar a página de usuário:


Agora a página de um chat existente, após uma notícia ter sido analisada. Nela o usuário pode visualizar o resultado da análise, a notícia que ele enviou, e requisitar metadados sobre a análise. Mantendo a opção de visualização e criação de chats antigos da tela anterior, e a opção de acessar Usuário no canto superior direito:

Ler Mais'. Below this, it states '89% de chance de ser Fake News.' in red. At the bottom, there are four buttons: 'Mostrar resultados mais detalhados da análise', 'Explicações baseadas em palavras', 'Versão do modelo', and 'Tempo de processamento'. The top right shows 'Nome do usuário' with a profile icon."/>

4.2 Padrões de Design

4.2.1 Visibilidade do status do sistema

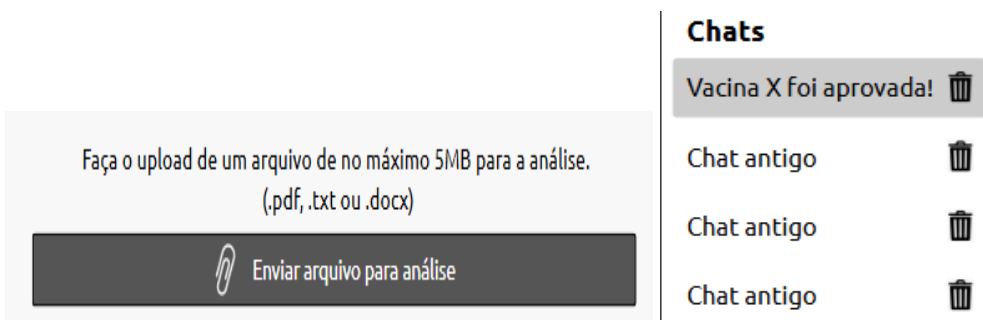
Para facilitar o entendimento das funcionalidades do sistema para o usuário serão utilizados botões que fornecem *feedback* ao usuário. Por exemplo, ao passar o mouse por cima de um botão de confirmação sua cor fica verde. Imagem ilustrativa a seguir:



The mockup shows a registration form titled "Crie sua conta." with a "FAKE NEWS" icon. It includes input fields for name (Gabriel), email (gabriel.npaiva@sempreceub.com), and password (masked with dots). A checkbox labeled "Aceito os termos e condições" is checked. A green button labeled "Criar conta" is highlighted with a mouse cursor, indicating a visual feedback state. Below the button is a link "Já tem uma conta? Faça o Login".

4.2.2 Correspondência entre o sistema e o mundo real

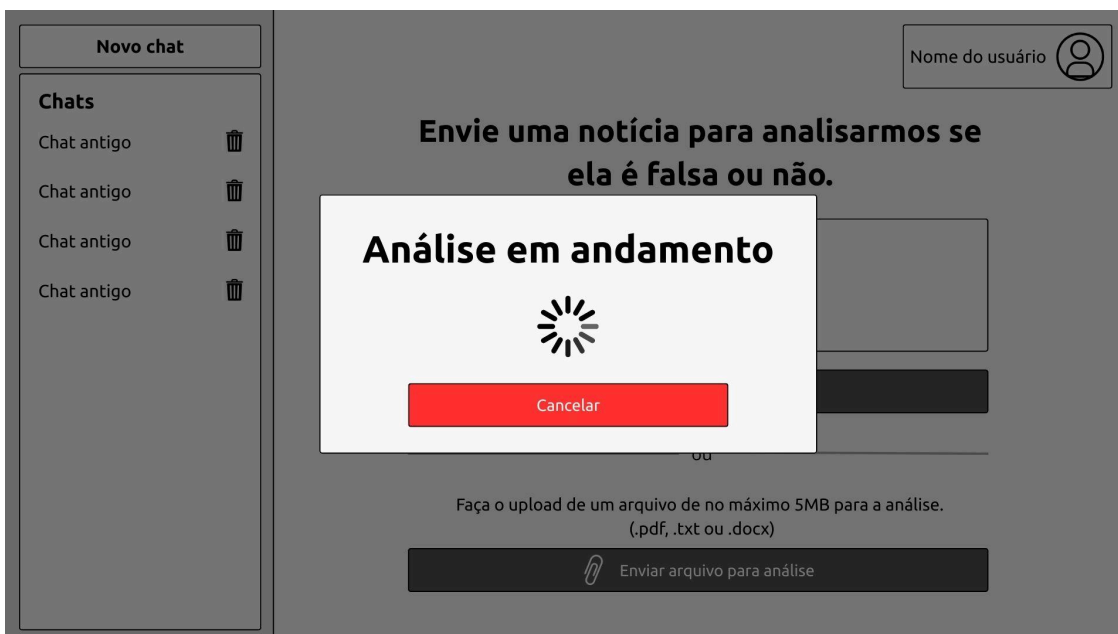
Para que exista uma correlação entre mundo real e o sistema, utilizamos ícones verossímeis e intuitivos. Como um clipe para anexo de arquivo, e uma lixeira para deletar um chat antigo:



The mockup consists of two parts. On the left, a file upload area with the text "Faça o upload de um arquivo de no máximo 5MB para a análise. (.pdf, .txt ou .docx)" and a button with a paperclip icon labeled "Enviar arquivo para análise". On the right, a "Chats" sidebar with a list of chat items. The first item is "Vacina X foi aprovada!" with a trash icon. The following three items are "Chat antigo", each also with a trash icon.

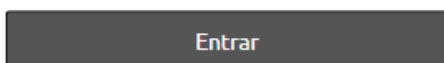
4.2.3 Liberdade de controle para o usuário

O sistema oferece ao usuário controle total sobre as operações. Um exemplo disso é que, enquanto a notícia está sendo analisada, o usuário pode cancelar a análise:



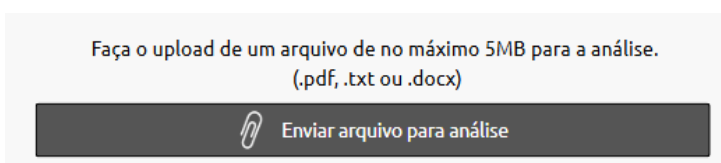
4.2.4 Consistência e padrões

Em todas as páginas mantemos os mesmos padrões de cores e formatos para elementos semelhantes. Isso facilita o usuário final a se acostumar a utilizar o site. Um bom exemplo disso é a padronização da aparência de um botão de confirmação:



4.2.5 Prevenção de erros

Para evitar que erros sejam cometidos, o funcionamento do site está sempre detalhado em legendas para orientar o usuário:



4.2.6 Reconhecimento em vez de memorização

A fim de evitar que o usuário repita as suas credenciais frequentemente, utilizamos um sistema de salvamento de login e senha:

☐ Lembrar de mim

4.2.7 Flexibilidade e eficiência de uso

Para que o uso do site se torne mais eficiente e prático, o sistema será configurado para navegar entre chats antigos com as setas do teclado, além disso, ao invés de clicar em botões de confirmação o usuário pode apertar o atalho “Enter”.

4.2.8 Estética e design minimalista

Para que informações desnecessárias sejam evitadas, e para que somente o conteúdo importante do site se destaca, optamos por um design minimalista e eficiente.

4.2.9 Ajudar usuários a reconhecerem, diagnosticarem e recuperarem-se de erros

Sempre que o usuário cometer um erro, ele será informado para que isso não impeça a utilização do sistema:



4.2.10 Ajuda e documentação

Foram criadas instruções para caso o usuário tenha dúvida de como funciona o sistema:

Caso você queira enviar uma notícia digitada para análise você deve:

- 1- Digitar no campo a notícia que você deseja enviar.
- 2- Clicar no botão de analisar a notícia digitada.

Chat antigo

Chat antigo

1 Digite a notícia aqui...

2 Analisar a notícia digitada

ou

Faça o upload de um arquivo de no máximo 5MB para a análise.
(.pdf, .txt ou .docx)

Enviar arquivo para análise

Caso você queira enviar uma notícia por arquivo para análise você deve:

Clicar no botão de enviar um arquivo para análise.

Chat antigo

Chat antigo

Digite a notícia aqui...

Analisar a notícia digitada

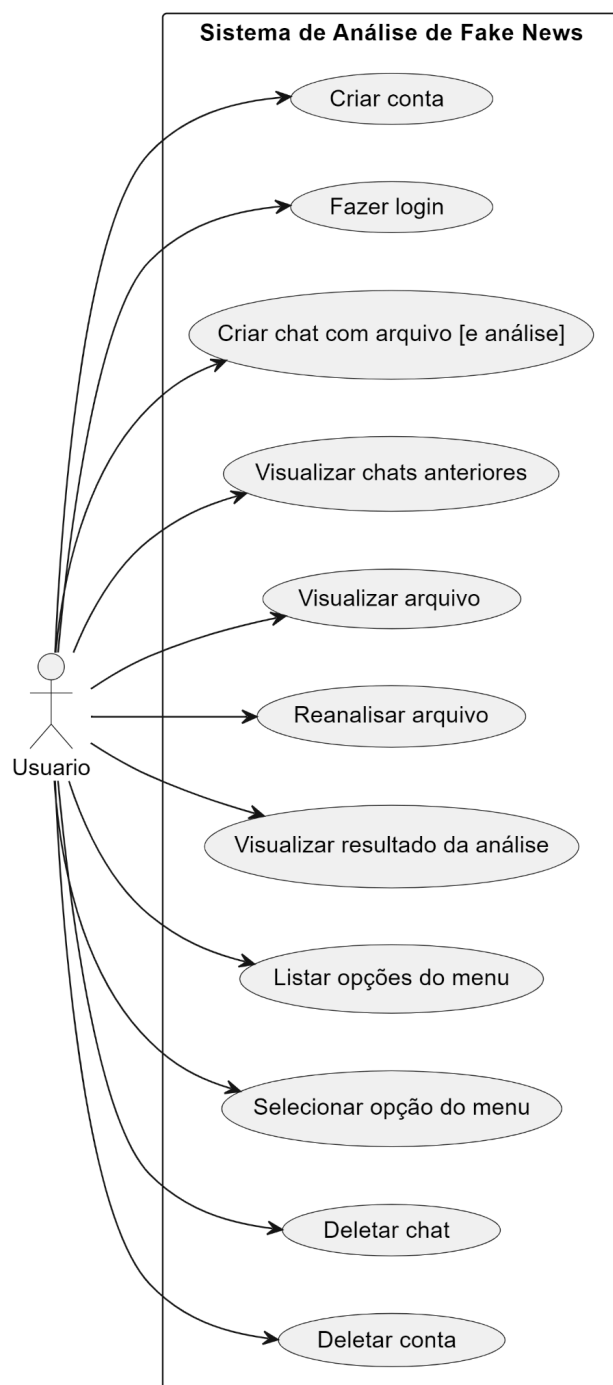
ou

Faça o upload de um arquivo de no máximo 5MB para a análise.
(.pdf, .txt ou .docx)

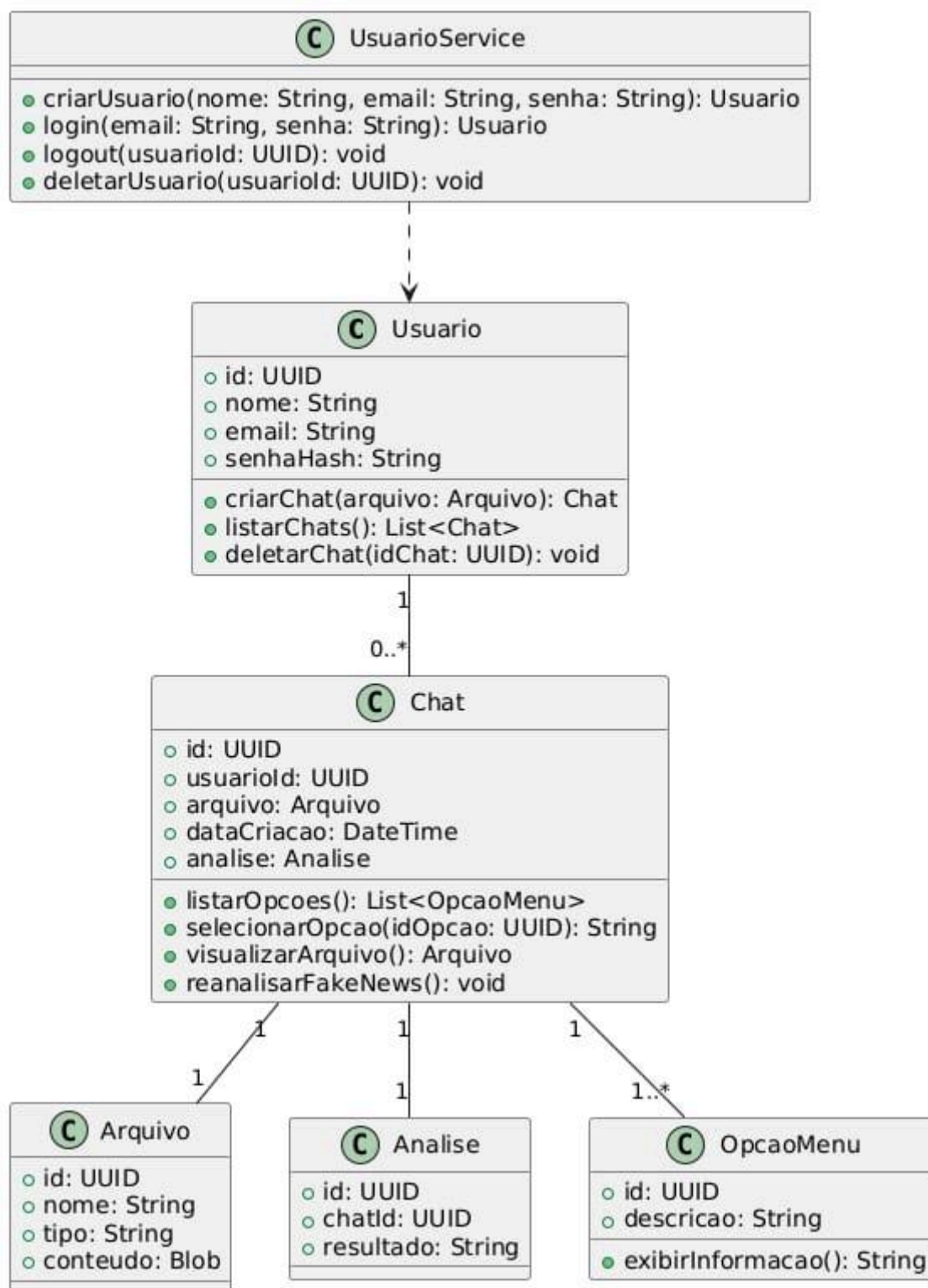
Enviar arquivo para análise

5. Modelagem do Sistema

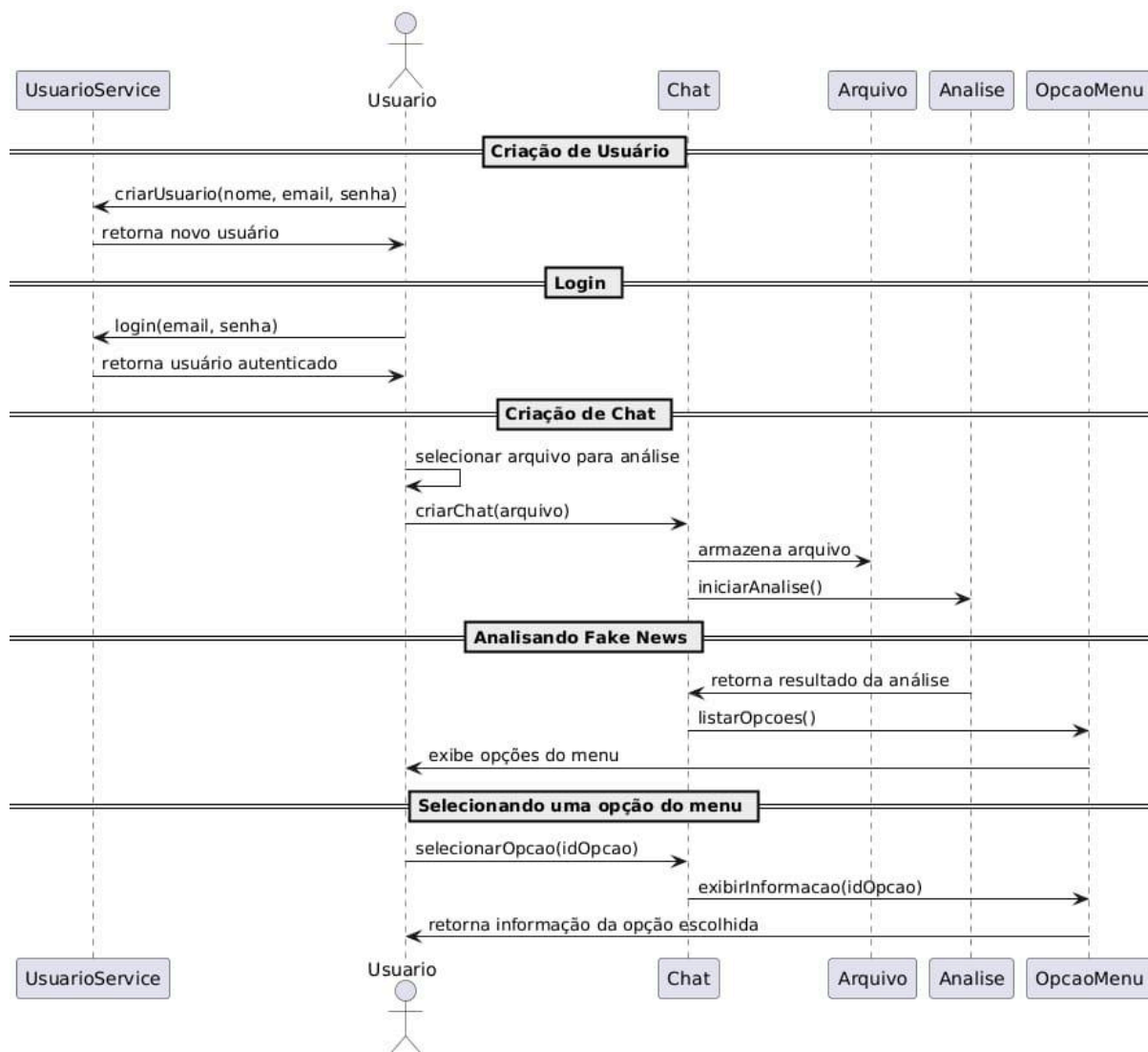
5.1 Diagrama de Casos de Uso



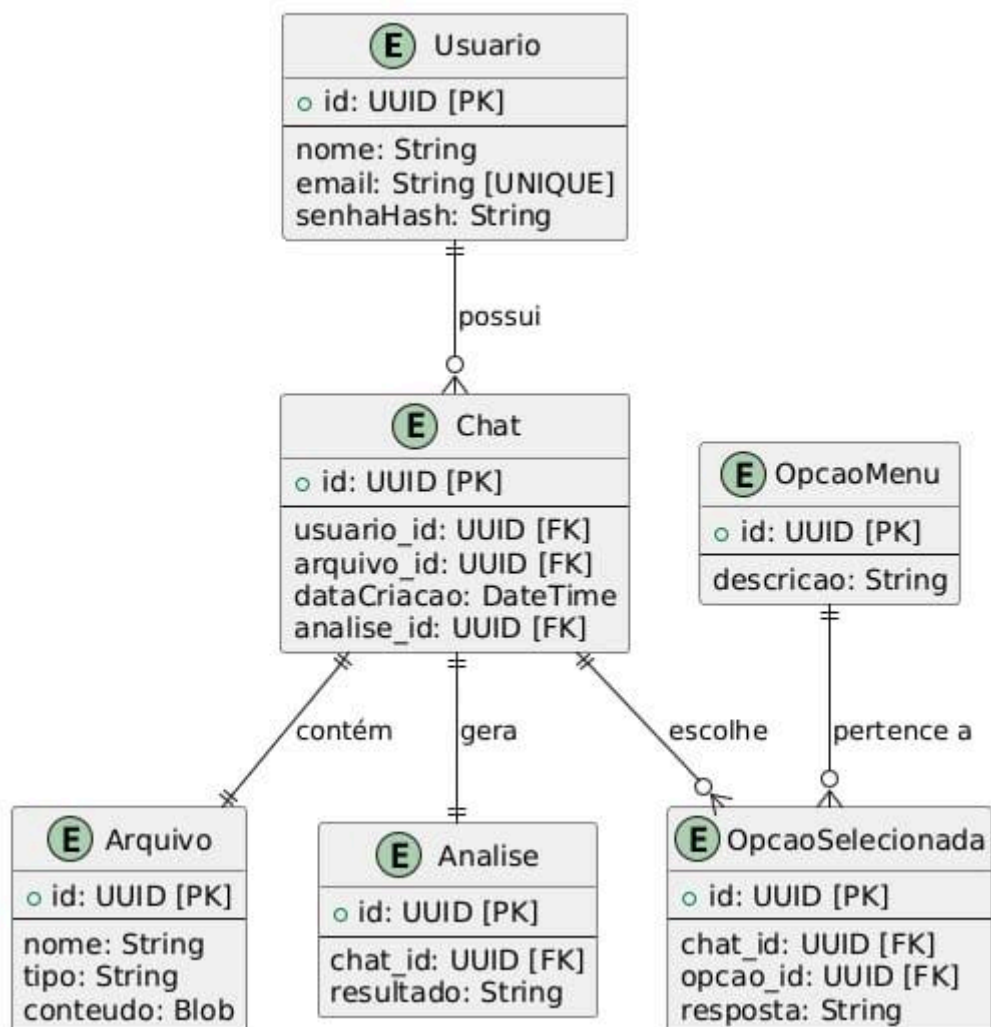
5.2 Diagrama de Classes



5.3 Diagrama de Sequência



5.4 Diagrama Entidade-Relacionamento

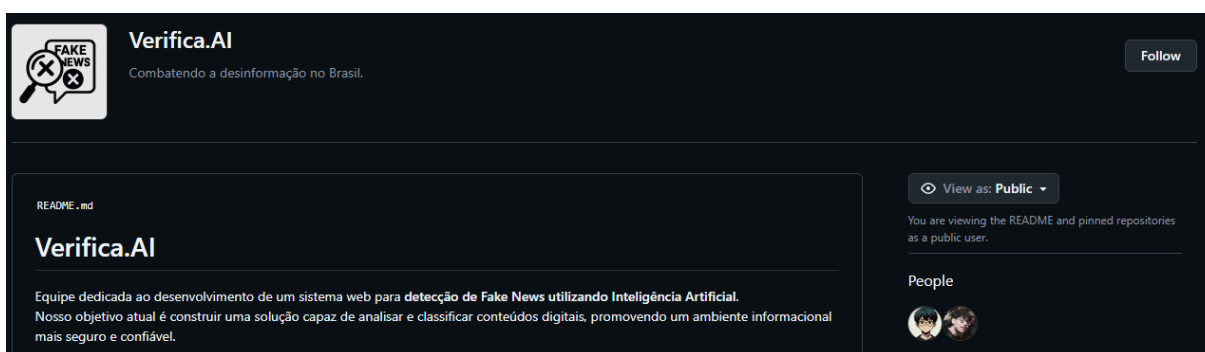


6. Desenvolvimento e Metodologia

6.1 Metodologia de Desenvolvimento

A metodologia utilizada será a kanban, muito utilizada em equipes de desenvolvimento ágil. Ela se baseia no uso de um quadro dividido em colunas que representam as etapas de um processo, permitindo que as tarefas sejam movidas entre essas colunas conforme avançam. O objetivo é melhorar a eficiência, limitar o trabalho em progresso e promover a melhoria contínua, oferecendo uma visão clara do status do trabalho e facilitando a identificação de gargalos.

6.2 Repositório de Código-Fonte



<https://github.com/Verifica-AI>

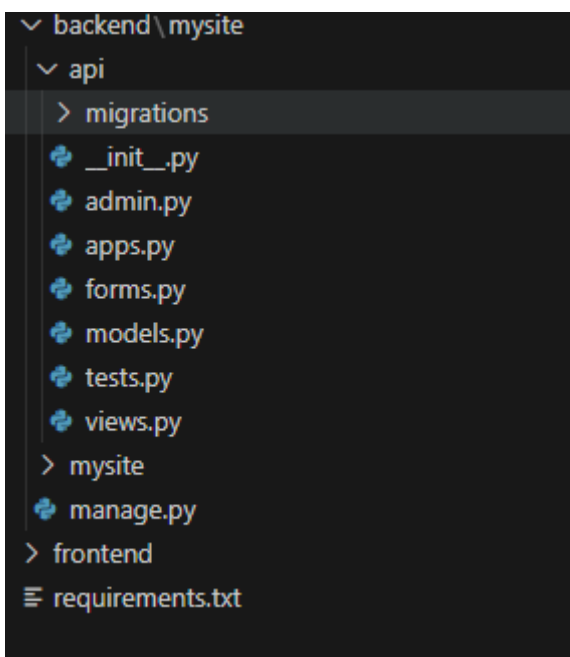
6.3 Estrutura do Código

Este projeto adota uma estrutura de diretórios separada para o frontend e o backend, visando facilitar a manutenção, escalabilidade e organização do código. O diretório backend/ contém a aplicação Django, organizada de forma modular por meio de apps localizados na pasta apps/, onde cada app é responsável por uma funcionalidade específica da aplicação. Utiliza-se o Django REST Framework para a criação de uma API RESTful, que será consumida pelo frontend. Já o diretório frontend/ contém a aplicação React, estruturada com subpastas dentro de src/, como components/ para componentes reutilizáveis, pages/ para as páginas da aplicação, e services/ para os serviços responsáveis por interações com a API.

A arquitetura adotada é baseada em um modelo desacoplado, também conhecido como SPA com API RESTful. A camada frontend, implementada como uma single-page application (SPA) em React, é responsável por toda a interface com o usuário, enquanto o backend, desenvolvido com Django e Django REST Framework, lida com as regras de negócio e persistência dos dados. A comunicação entre as camadas é feita via requisições HTTP utilizando endpoints REST. Essa abordagem oferece vantagens como reutilização da API por múltiplos clientes (incluindo aplicações móveis), maior flexibilidade para escalar cada camada separadamente, e a possibilidade de desenvolvimento simultâneo por equipes

distintas, promovendo maior produtividade e organização no ciclo de desenvolvimento.

Estrutura de pastas do projeto:



Exemplo de código comentado:

```

class UsuarioRegistroForm(UserCreationForm):
    """
    Formulário para registro de novos usuários estendendo o UserCreationForm do Django.
    """
    nome = forms.CharField(max_length=100, required=True,
                           widget=forms.TextInput(attrs={'class': 'form-control', 'placeholder': 'Seu nome'}))
    email = forms.EmailField(required=True,
                             widget=forms.EmailInput(attrs={'class': 'form-control', 'placeholder': 'seu@email.com'}))

    class Meta:
        model = Usuario
        fields = ['nome', 'email', 'password1', 'password2']

    def clean_email(self):
        """Validação personalizada para garantir email único"""
        email = self.cleaned_data.get('email')
        if Usuario.objects.filter(email=email).exists():
            raise forms.ValidationError('Este email já está em uso. Por favor, use outro email.')
        return email
  
```

7. Gestão de Tarefas e Kanban

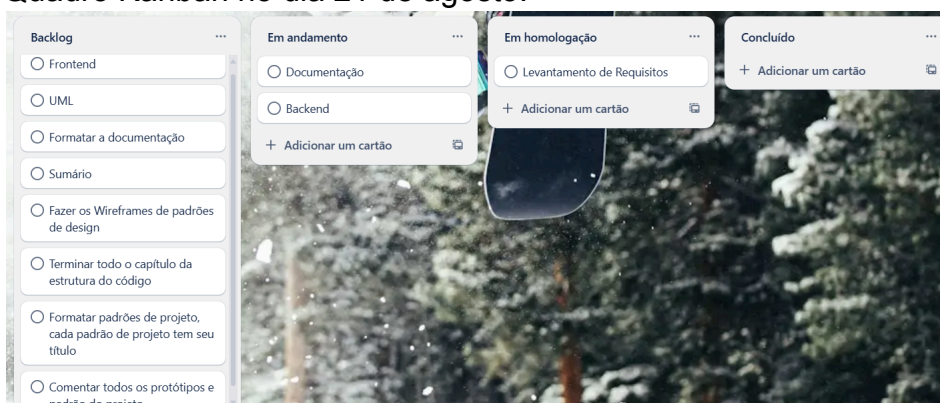
7.1 Uso do Kanban

O projeto utilizará metodologia ágil Kanban como estrutura central de gerenciamento, com a ausência de sprints fixos, o que permite à equipe maior adaptabilidade frente às demandas do projeto integrador. O fluxo de trabalho será organizado em cinco colunas: "Backlog", que mantém o repositório de todas as atividades pendentes priorizadas; "Backlog da Sprint", contendo as tarefas selecionadas para serem as próximas desenvolvidas; "Em Execução", que evidencia as atividades em andamento ativo pela equipe; "Em Homologação", destinada às tarefas que aguardam validação e testes de qualidade; e "Finalizado", que registra as entregas concluídas e aprovadas.

7.2 Quadro Kanban

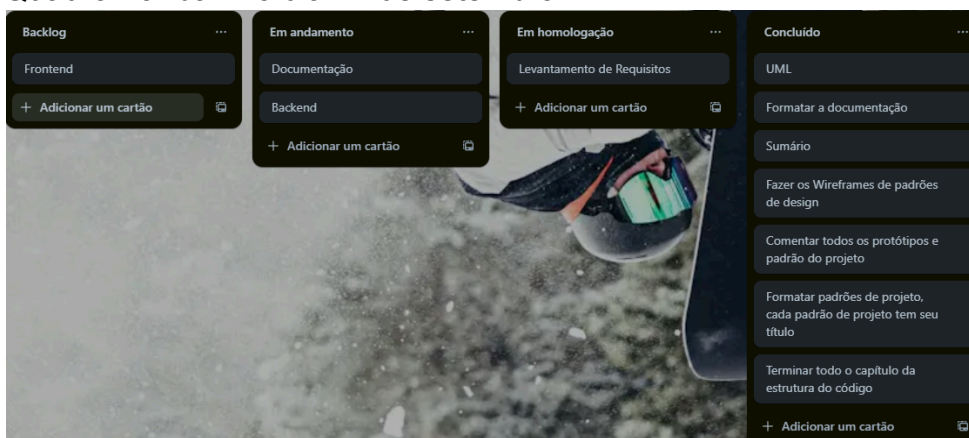
7.2.1 21 de agosto

Quadro Kanban no dia 21 de agosto:



7.2.2 17 de setembro

Quadro Kanban no dia 17 de setembro:



8. Testes e Validação

8.1 Estratégia de Testes

Os testes realizados foram, principalmente, testes unitários, com o objetivo de assegurar que cada componente do sistema esteja funcionando conforme o esperado. Seis testes de integração também foram realizados para verificar o funcionamento da principal funcionalidade do sistema, que é a detecção de fake news, e também a autenticação de usuário.

8.2 Ferramentas de Teste

Para garantir a qualidade do software, adotaremos as seguintes ferramentas e estratégias:

- **Plano de Testes:** Desenvolvemos um plano de testes abrangente que detalhou a estratégia para validar as funcionalidades do sistema. Este plano incluirá os tipos de testes a serem executados (funcionais, de desempenho, de segurança, etc.), os ambientes de teste, os critérios de entrada e saída, e as responsabilidades da equipe.
- **Casos de Teste:** Serão criados casos de teste detalhados para cada funcionalidade, descrevendo os passos para a execução, os dados de entrada, e os resultados esperados. Esses casos de teste servirão como roteiro para a execução dos testes e a verificação do comportamento do software.
- **Relatórios de Teste:** Após a execução dos testes, serão gerados relatórios **de teste** completos. Esses relatórios documentarão os resultados de cada caso de teste, incluindo a identificação de quaisquer defeitos ou não conformidades encontrados. Eles serão essenciais para acompanhar o progresso dos testes, tomar decisões sobre a qualidade do software e garantir a rastreabilidade das correções.
- **Ferramentas de Automação:** Para otimizar e agilizar o processo de testes, utilizaremos Postman para testes de API e Pytest para testes de unidade e integração. O Postman nos permitirá criar e executar coleções de requisições HTTP, validando as respostas das APIs de forma eficiente. O Pytest, por sua vez, será essencial para escrever testes robustos e escaláveis para o código-fonte, garantindo que as menores unidades de código funcionem como esperado.

9. Implantação e Manutenção

9.1 Processo de Implantação

O sistema de detecção de fake news, com seu backend em Django e frontend em React, será disponibilizado para uso através de um processo de deploy na nuvem.

A hospedagem será realizada em uma plataforma de nuvem robusta (como AWS, Google Cloud ou Azure), garantindo alta disponibilidade, escalabilidade e segurança. O processo de instalação envolverá a configuração de ambientes de produção para o backend Django (incluindo o LLM treinado) e o frontend React. O backend será empacotado em containers Docker para facilitar a implantação e garantir consistência entre os ambientes. O frontend será construído e servido por um servidor web otimizado para aplicações React.

As configurações incluirão a parametrização de variáveis de ambiente, chaves de API, credenciais de banco de dados e as configurações específicas para o modelo de linguagem (LLM) treinado para detecção de fake news. Serão utilizados serviços de gerenciamento de segredos na nuvem para proteger informações sensíveis. Um pipeline de CI/CD (Integração Contínua/Entrega Contínua) será implementado para automatizar o processo de deploy, garantindo que novas versões sejam liberadas de forma rápida e confiável.

9.2 Plano de Manutenção

Para garantir a longevidade e a performance do sistema, será implementado um plano de manutenção abrangente, cobrindo tanto correções quanto evoluções.

- **Plano de Suporte e Atualizações:** Este plano definirá a manutenção corretiva, que abordará a correção de bugs, vulnerabilidades de segurança e problemas de desempenho que possam surgir na produção. Também englobará a manutenção evolutiva, que incluirá a adição de novas funcionalidades, aprimoramentos no modelo de detecção de fake news (por exemplo, retreinamento do LLM com novos dados ou atualização para um modelo mais avançado), e otimização de infraestrutura. Será estabelecido um cronograma para atualizações regulares e um processo para lidar com correções urgentes.
- **Monitoramento e Logs:** Utilizaremos ferramentas de monitoramento dedicadas (como Prometheus, Grafana ou soluções nativas da plataforma de nuvem) para acompanhar o desempenho da aplicação, a utilização de recursos do servidor e o comportamento do LLM. Logs detalhados do backend (Django) e do frontend (React), incluindo logs do LLM, serão centralizados e analisados por ferramentas como ELK Stack (Elasticsearch, Logstash, Kibana) ou Splunk, permitindo uma rápida análise de erros e desempenho, identificação de gargalos e investigação de anomalias na detecção de fake news.

9.3 Versão Executável ou Deploy na Nuvem

Após a conclusão do processo de implantação, teremos uma aplicação funcional para uso acessível publicamente através de um domínio configurado. Os usuários poderão interagir com a interface do frontend React para submeter textos e receber o resultado da detecção de fake news processado pelo backend Django e o LLM.

9.4 Documentação Técnica

Será elaborada uma documentação técnica abrangente, essencial para desenvolvedores futuros ou para novas equipes que venham a trabalhar no projeto. Esta documentação incluirá:

- Arquitetura do sistema (diagramas, descrição das camadas e componentes).
- Estrutura do banco de dados (esquemas, relacionamentos).
- APIs do backend Django (endpoints, métodos, payloads de requisição/resposta).
- Estrutura do frontend React (componentes, fluxo de dados).
- Detalhes sobre o treinamento e a integração do LLM.
- Instruções para configuração do ambiente de desenvolvimento.
- Processos de build e deploy.

9.5 Manual do Usuário/Treinamento

Para facilitar a adoção e o uso eficaz do sistema, será fornecido um manual do usuário claro e conciso. Este guia para os usuários explicará como interagir com a interface do frontend, como submeter textos para análise, interpretar os resultados da detecção e solucionar problemas comuns. Adicionalmente, sessões de treinamento poderão ser oferecidas para usuários-chave ou equipes específicas, garantindo que compreendam plenamente as funcionalidades e benefícios da ferramenta de detecção de fake news.

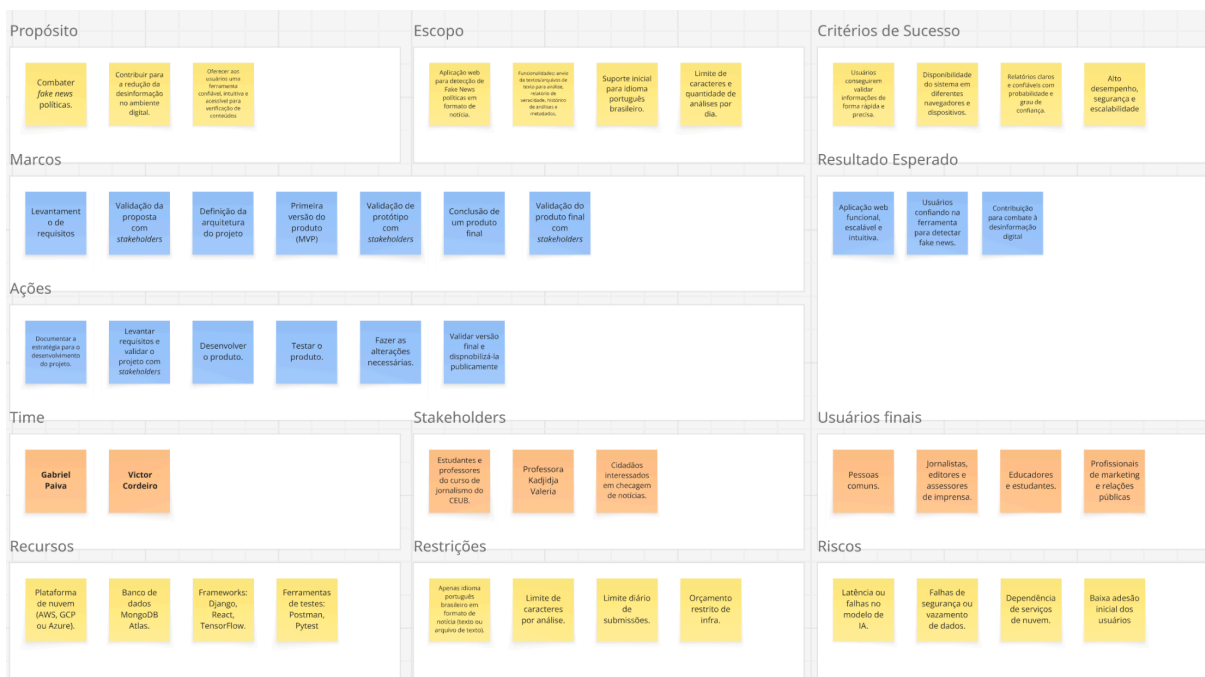
10. Conclusão

Este projeto visa desenvolver uma aplicação web inovadora para a detecção de fake news, utilizando um Modelo de Linguagem de Grande Escala (LLM) ajustado (fine-tuned) especificamente para essa finalidade. A arquitetura robusta, com um backend em Django e frontend em React, garante uma solução eficiente e escalável.

Com este software, esperamos trazer benefícios significativos, como o combate à desinformação, fornecendo aos usuários uma ferramenta confiável para verificar a autenticidade de conteúdos. A capacidade de identificar fake news rapidamente pode impactar positivamente a tomada de decisões, a formação de opinião pública e a disseminação de informações precisas. Em última análise, buscamos contribuir para um ambiente digital mais informado e seguro para todos.

11. Anexos

ANEXO I - Project Model Canvas (PMC)



12. Referências

BARCELOS, T. N. et al. Análise de fake news veiculadas durante a pandemia de COVID-19 no Brasil. *Revista Panamericana de Salud Pública*, v. 45, e65, 2021. Disponível em: <https://www.scielosp.org/pdf/rpsp/2021.v45/e65/pt>. Acesso em: 23 set. 2025.

ALLCOTT, H.; GENTZKOW, M. Social Media and Fake News in the 2016 Election. *Journal of Economic Perspectives*, v. 31, n. 2, p. 211-236, 2017. Disponível em: <https://www.aeaweb.org/articles?id=10.1257/jep.31.2.211>. Acesso em: 23 set. 2025.

SHU, Kai et al. Fake News Detection on Social Media: A Data Mining Perspective. 2017. Preprint. Disponível em: <https://arxiv.org/pdf/1708.01967>. Acesso em: 23 set. 2025.

JOVANOVIC, M.; CAMPBELL, M. Generative Artificial Intelligence: Trends and Prospects. *Computer*, v. 55, n. 10, p. 107-112, 2022. Disponível em: <https://www.computer.org/csdl/magazine/co/2022/10/09903869/1H0G6xvtREk>. Acesso em: 23 set. 2025.

ALGHAMDI, Jawaher; LIN, Yuqing; LUO, Suhuai. Towards COVID-19 fake news detection using transformer-based models. *Knowledge-Based Systems*, v. 274, 110642, 2023. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950705123003921>. Acesso em: 23 set. 2025.

AVAAZ. O Brasil está sofrendo uma infodemia de Covid-19. 4 de maio de 2020. Disponível em: https://avaazimages.avaaz.org/brasil_infodemia_coronavirus.pdf. Acesso em: 23 set. 2025.

