

Especificação de Requisitos de Software

Sistema AlethelA

1. Introdução

1.1. Propósito

O presente documento descreve a Especificação de Requisitos para a plataforma AlethelA, em desenvolvimento pela Equipe 11 da disciplina de Projeto de Software.

A plataforma AlethelA é um assistente inteligente de conversacional projetado com o propósito de combater à desinformação no cenário político brasileiro.

1.2. Escopo do Produto

A rápida disseminação de notícias falsas, especialmente através de redes sociais, representa uma ameaça significativa para a sociedade, influenciando a opinião pública e a estabilidade política. O contexto político brasileiro, em particular, é um dos principais alvos desse tipo de desinformação.

A verificação manual de notícias é um processo complexo, caro e demorado, incapaz de acompanhar o enorme volume de conteúdo gerado diariamente. Embora soluções de aprendizado de máquina existam, muitas dependem de modelos proprietários e pagos, e poucas foram empiricamente validadas para as nuances do cenário brasileiro.

Nesse sentido, a Plataforma AlethelA foi concebida com o intuito de auxiliar, principalmente, a sociedade brasileira no combate à desinformação política. O objetivo é fornecer ao usuário da plataforma uma ferramenta poderosa e transparente para a verificação de fatos, ajudando a mitigar o impacto negativo da disseminação de notícias falsas.

1.3. Público-Alvo

O público alvo da plataforma AletheIA compreende qualquer usuário que deseje verificar a veracidade de notícias, com foco em conteúdos políticos do Brasil.

1.3. Definição, Siglas e Abreviações

LLM (Modelo de Linguagem de Grande Porte): modelo de IA usado para analisar o texto.

RAG (Geração Aumentada por Recuperação): abordagem em que o sistema busca **informações verificadas** em uma base de conhecimento antes de gerar a análise.

Gemma 2: modelo de linguagem open-source do Google utilizado como núcleo do assistente.

1.4. Referências

- GÔLO, Marcos P. S.; MORI, Adriel L. V.; OLIVEIRA, William G.; BARBOSA, Jacson R.; GRACIANO-NETO, Valdemar V.; LIMA, Eliomar A. de; MARCACINI, Ricardo M.. On the use of Large Language Models to Detect Brazilian Politics Fake News. In: ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL (ENIAC), 21. , 2024, Belém/PA. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2024 . p. 1-12, novembro 2024. Disponível em: <https://doi.org/10.5753/eniac.2024.245119>. Acesso em: outubro 2025.
- Documento de Histórias de Usuário. Disponível em: <https://docs.google.com/document/d/1MpH1zcc71cH5qVfYRIGa9ZENTebKMNQ9RI6uz-cCnJ4/edit?tab=t.0>

2. Visão Geral do Produto

2.1 Perspectiva do produto

A plataforma AletheIA é um assistente de IA para combate à desinformação no cenário político brasileiro, que realiza verificação de fatos sobre notícias fornecidas pelo usuário, combinando LLM e RAG para basear a análise em informações atuais e verificadas, oferecendo transparência e veracidade às informações submetidas a plataforma.

2.2 Funções do produto

Entrada: texto de uma notícia.

Recuperação: busca em base com notícias verificadas, portais de notícias confiáveis (ex.: G1, UOL) e agências de checagem.

Análise: cruzamento da notícia com o contexto recuperado por meio do Gemma 2.

Saída:

- Classificação: “Verdadeira”, “Falsa” ou “Indeterminada”
- Score de probabilidade (ex.: 95% de chance de ser fake);
- Justificativa em tópicos, clara e coerente.

2.3 Classes e características dos usuários

Usuários que desejam verificar a veracidade de notícias, com foco em conteúdos políticos do Brasil. Incluindo:

- Cidadãos Comuns;
- Jornalistas Investigativos;
- Cientistas Políticos;
- Políticos;
- Órgãos Públicos; e outros.

2.4 Ambiente de operação

Modelo de linguagem: Google Gemma 2 (via Ollama).

Arquitetura de análise: RAG (Retrieval-Augmented Generation).

Base de conhecimento: banco vetorial (ex.: ChromaDB, FAISS) com fontes confiáveis.

Backend: Python (ex.: LangChain ou LlamaIndex).

Frontend: a definir (ex.: Streamlit, Flask, React).

2.5 Restrições de projeto e implementação

- Uso de LLM open-source (Gemma 2) para flexibilidade e controle, evitando dependência de APIs pagas.
- Análise fundamentada em fontes verificadas, exigindo curadoria contínua da base.

2.6 Suposições e dependências

- Disponibilidade de uma base de conhecimento atualizada e confiável (jornais e agências de checagem).
- Dependência do pipeline RAG mais Gemma 2 e da infraestrutura de banco vetorial e orquestração (LangChain/LlamaIndex).

3. Requisitos

3.1 Requisitos funcionais

RF-001 - O sistema deve permitir que o usuário cadastre-se na plataforma.

RF-002 - O sistema deve permitir a autenticação do usuário (login e logout) na plataforma.

RF-003 - O sistema deve avaliar a veracidade da notícia fornecida pelo usuário.

RF-004 - O sistema deve permitir que o usuário insira o texto de uma notícia para verificação.

RF005 O sistema deve analisar a notícia fornecida pelo usuário, executando:

- **Vetorização/Recuperação de Contexto (RAG):** O sistema converte a notícia em um vetor e busca em nossa base de dados por notícias e fatos similares de fontes confiáveis.
- **Análise Aumentada:** Combinar notícia e contexto recuperado e enviar ao LLM com instruções de checagem. A notícia original, juntamente com o contexto recuperado, é enviada ao Gemma 2 através de um

prompt estruturado. O prompt instrui o modelo a atuar como um chegador de fatos e cruzar as informações.

- **Geração da Resposta:** O Gemma 2 analisa o material e gera uma resposta.

RF-006 - O sistema deve disponibilizar ao usuário o resultado da análise da notícia, indicando: a classificação da notícia em: “verdadeira”, “falsa” ou “indeterminada”.

RF-007 - O sistema deve disponibilizar ao usuário o resultado da análise da notícia, indicando: a o score de probabilidade de veracidade da notícia, em porcentagem.

RF-008 - O sistema deve disponibilizar ao usuário o resultado da análise da notícia, indicando: uma justificativa em tópicos com os motivos da classificação.

RF-009 - O sistema deve permitir ao usuário a exportação do resultado de avaliação da notícia.

RF-010 - O sistema deve disponibilizar o histórico de pesquisa de notícias ao usuário.

3.2 Requisitos não funcionais

RNF-Desempenho:

RNF-Precisão:

RNF-Explicabilidade:

RNF-Confiabilidade:

Usabilidade:

RNF-: Resultado deve apresentar classificação, score e justificativa de forma clara e legível em uma única tela.

RNF-Dependência: Preferência por componentes open-source, evitando APIs pagas.

3.5 Restrições de projeto

RP-01: Arquitetura baseada em RAG e LLM Gemma 2 (Ollama).

RP-02: Backend em Python, usando bibliotecas como LangChain ou LlamaIndex para orquestrar o fluxo RAG).

RP-03: Banco de dados vetorial (ex: ChromaDB, FAISS) populado com notícias de fontes confiáveis.

4. Anexos

4.1 Diagrama de Casos de Uso

