Regeneration Credit AI Assistant

Sistema de Chatbot com RAG para Projeto de Regeneração da Natureza

Data: 17/11/2025
Versão: 1.0.0-beta
Relatório Técnico-Executivo

1. Introdução

O Regeneration Credit é um sistema peer-to-peer de regeneração da natureza baseado em blockchain, que conecta pessoas interessadas em apoiar projetos de impacto ambiental com produtores e regeneradores que executam ações práticas de preservação e restauração.

Para facilitar o entendimento e a adoção do projeto por diferentes públicos, foi desenvolvido o Regeneration Credit AI Assistant, um chatbot inteligente que utiliza Inteligência Artificial avançada para responder dúvidas sobre o projeto em linguagem natural e acessível.

1.1 Objetivo do Chatbot AI

O assistente foi projetado para permitir que qualquer pessoa, independente do nível técnico, possa:

- Entender os conceitos fundamentais do projeto
- Consultar informações sobre tokenomics e distribuição
- Aprender sobre contratos inteligentes e implementação técnica
- Obter orientações sobre uso do aplicativo Core RC
- Esclarecer dúvidas sobre blockchain Sintrop e infraestrutura

1.2 Escopo do Relatório

Este documento apresenta uma visão executiva do sistema desenvolvido, abordando as tecnologias adotadas, a arquitetura da solução, o sistema RAG (Retrieval-Augmented Generation) implementado e os próximos passos planejados para evolução do projeto.

2. Linguagem de Programação

2.1 Linguagem Principal: Python 3.10+

O projeto foi desenvolvido integralmente em Python 3.10+, uma escolha motivada principalmente pela forte aderência desta linguagem a projetos envolvendo Large Language Models (LLMs).

2.2 Justificativa da Escolha

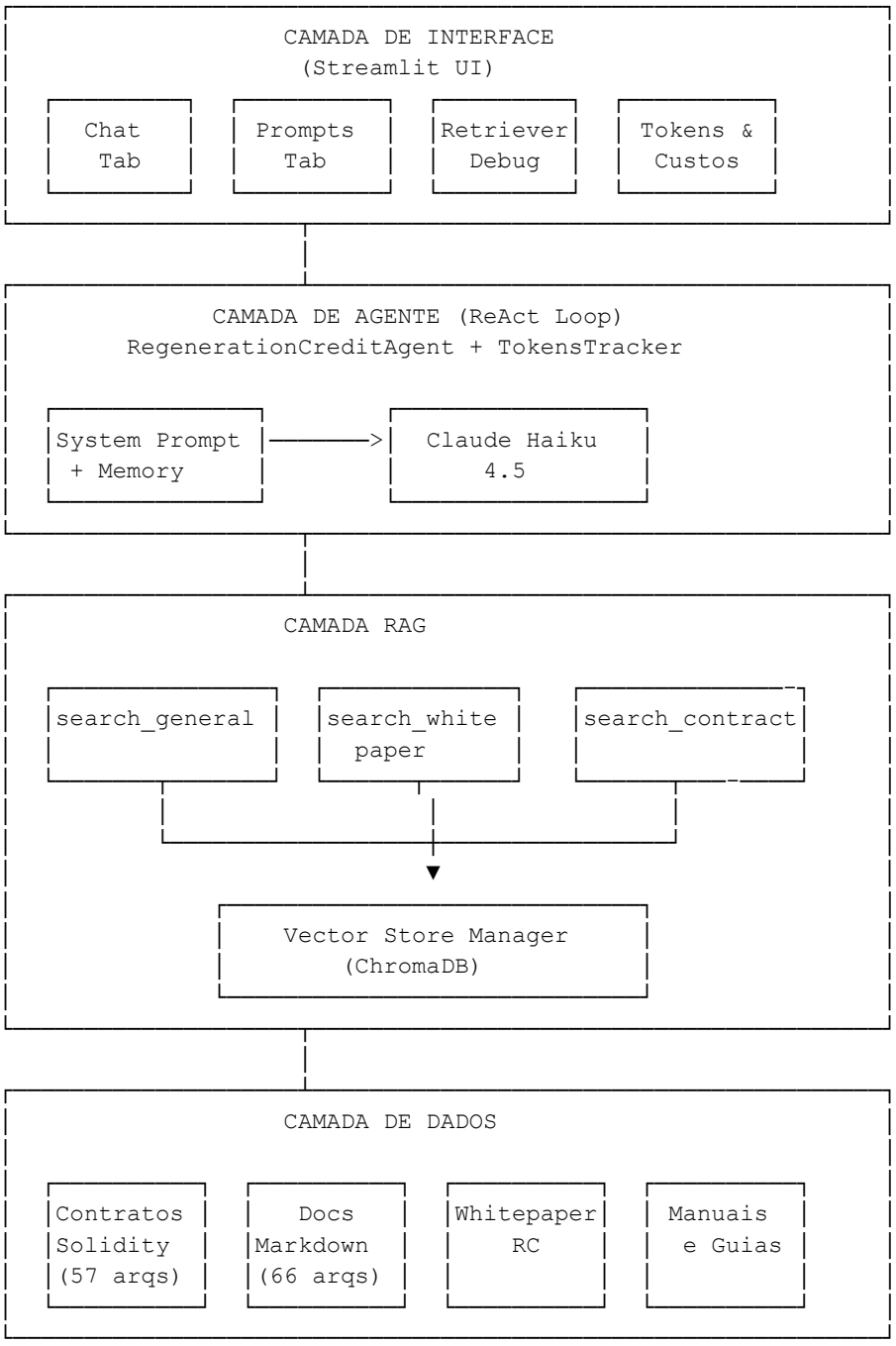
Python se consolidou como a linguagem predominante no ecossistema de IA e Machine Learning, oferecendo vantagens decisivas:

- Ecossistema Rico em IA/ML: Bibliotecas maduras e especializadas (LangChain, Transformers, ChromaDB) com suporte nativo e documentação abrangente
- Integração Facilitada com LLMs: APIs oficiais dos principais provedores (Anthropic, OpenAI) mantêm SDKs Python como prioridade
- Prototipagem Rápida: Sintaxe clara e bibliotecas de alto nível permitem desenvolvimento ágil, ideal para projetos de POC (Proof of Concept)
- Comunidade Ativa: Vasta comunidade focada em IA com recursos, exemplos e suporte contínuo

3. Arquitetura do Sistema

3.1 Visão Geral da Arquitetura

O sistema foi projetado seguindo uma arquitetura modular em camadas, separando responsabilidades e facilitando manutenção e evolução:



3.2 Descrição dos Componentes

Camada de Interface (Streamlit):

Responsável pela interação com o usuário através de múltiplas abas especializadas. Além do chat principal, oferece transparência através de abas de debug (Retriever, Tokens & Custos) e inspeção (Prompts).

Camada de Agente (RegenerationCreditAgent):

Núcleo inteligente do sistema. Implementa um loop ReAct (Reasoning + Acting) manual, onde o agente raciocina sobre a pergunta do usuário, decide quais ferramentas usar, executa buscas e formula respostas. Inclui sistema de memória conversacional para manter contexto entre turnos.

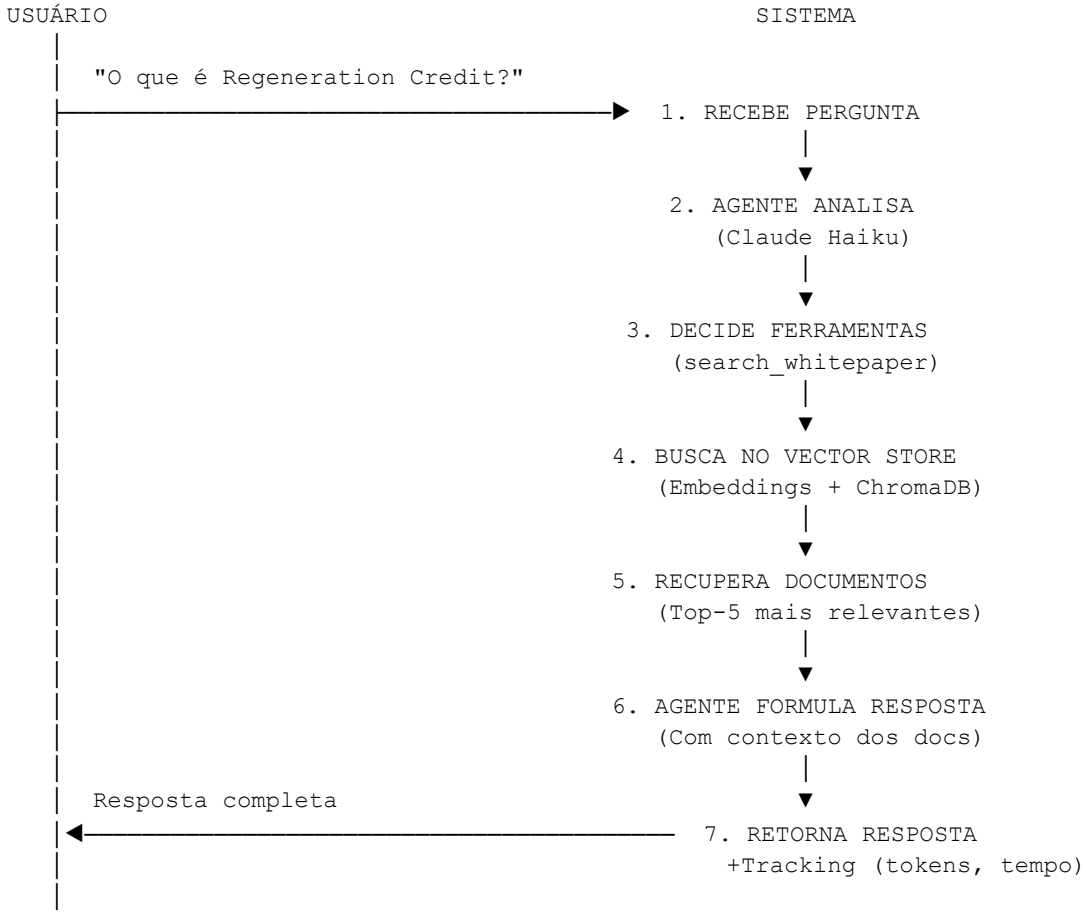
Camada RAG (Retrieval-Augmented Generation):

Gerencia o acesso ao conhecimento através de 4 ferramentas especializadas que buscam informações no vector store. Realiza auditoria completa de cada busca (tempo, scores, metadados) para debugging e otimização.

Camada de Dados:

Repositório de conhecimento com 9 tipos de fontes diferentes, totalizando mais de 120 documentos indexados e pesquisáveis semanticamente.

3.3 Fluxo de Processamento de Pergunta



3.4 Padrões de Design Utilizados

Loop ReAct Manual:

Implementação customizada do padrão ReAct (Reason + Act), permitindo controle granular sobre cada iteração do agente, incluindo tracking detalhado de tokens e custos.

Separação de Responsabilidades:

Cada camada possui responsabilidades bem definidas: interface (UX), agente (lógica), RAG (retrieval), dados (armazenamento). Facilita testes, manutenção e evolução independente.

Modularização:

Componentes independentes e reutilizáveis (TokensTracker, VectorStoreManager, RAGTools) que podem ser testados isoladamente e utilizados em outros contextos.

4. Sistema RAG Estruturado

4.1 Conceito e Funcionamento do RAG

RAG (Retrieval-Augmented Generation) é uma técnica que combina a capacidade de geração de texto de LLMs com a recuperação de informações relevantes de uma base de conhecimento específica.

Ao invés de depender apenas do conhecimento pré-treinado do modelo de IA (que pode estar desatualizado ou não incluir informações específicas do projeto), o sistema:

1. Busca documentos relevantes na base de conhecimento do projeto
2. Fornece esses documentos como contexto para o LLM
3. O LLM gera respostas baseadas nesse contexto específico

Usabilidade para o Projeto:

- Respostas Precisas: Baseadas em documentação oficial do projeto
- Sempre Atualizado: Basta atualizar os documentos e reprocessar
- Rastreável: Cada resposta pode ser vinculada às fontes consultadas
- Econômico: Reduz necessidade de fine-tuning de modelos

4.2 Fontes de Dados (9 tipos, 120+ documentos)

O sistema indexa e processa 9 tipos diferentes de fontes de conhecimento:

1. Contratos Solidity (57 arquivos)

Implementação técnica dos smart contracts: Pools, Rules, interfaces e tipos.

2. Documentação Markdown (66 arquivos)

Documentação técnica gerada automaticamente do código fonte.

3. README.md

Visão geral do projeto principal, arquitetura e instruções.

4. CHANGELOG.md

Histórico completo de mudanças e evolução do projeto.

5. Whitepaper Regeneration Credit

Visão estratégica, tokenomics e regras de negócio do sistema.

6. Manual Core RC

Guia completo do usuário do aplicativo Core (cadastro, níveis, saques, certificados).

7. Tutorial de Carteira MetaMask

Passo a passo para criar e configurar carteira blockchain.

8. Guia de Mineração Sintrop

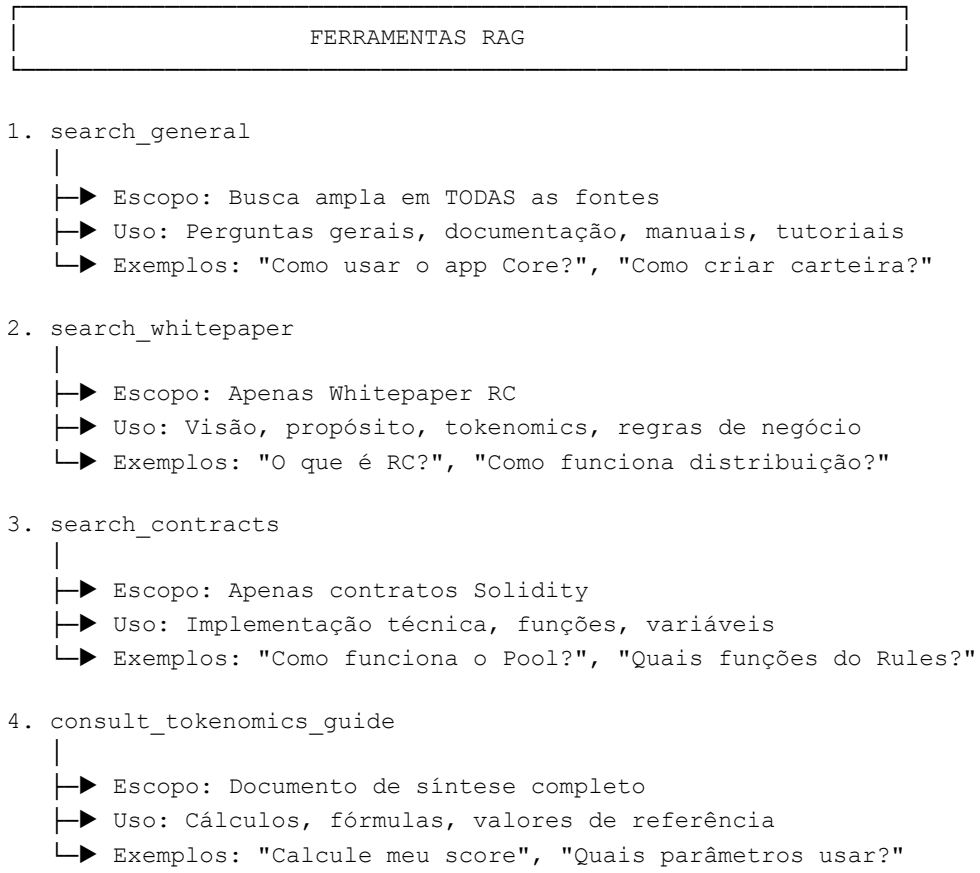
Instruções técnicas para setup de nós e mineração na blockchain.

9. Whitepaper Sintrop Blockchain

Arquitetura da blockchain, consenso Proof-of-Work e especificações técnicas.

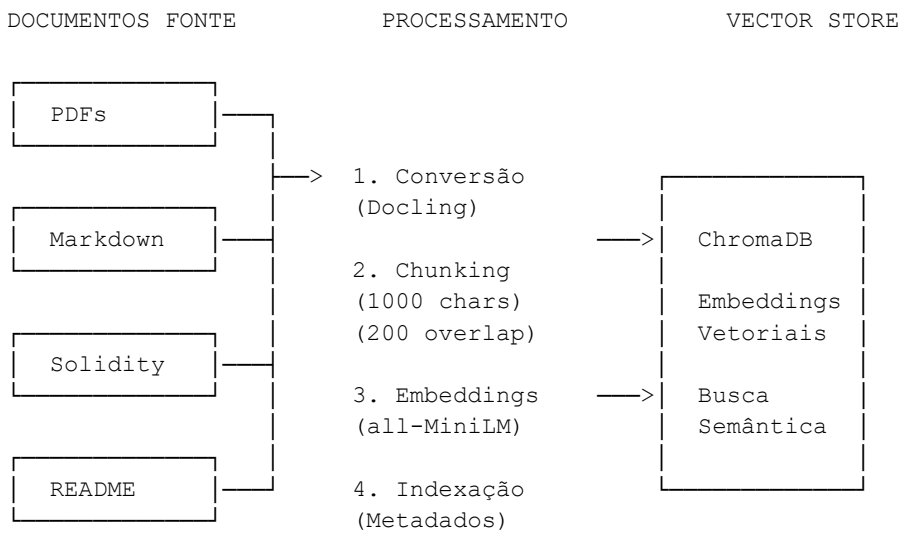
4.3 Ferramentas RAG Especializadas

Para otimizar a busca e reduzir ruído, foram criadas 4 ferramentas especializadas que o agente pode escolher conforme o tipo de pergunta:



4.4 Processamento e Indexação

O pipeline de processamento garante que documentos de diferentes formatos sejam uniformemente indexados:



Estratégia de Chunking:

Documentos são divididos em chunks (pedaços) de 1000 caracteres com sobreposição de 200 caracteres. Isso garante que informações contextuais não sejam perdidas nas divisões e permite recuperação mais precisa.

Metadados Enriquecidos:

Cada chunk é enriquecido com metadados (source_type, source, título) que permitem filtragem e rastreamento da origem das informações.

5. Tecnologias e Ferramentas

5.1 Frameworks e Bibliotecas Core

LangChain 1.0+ (Orquestração de IA):

Framework Python especializado em desenvolvimento de aplicações com LLMs. Fornece abstrações de alto nível para agentes, memória conversacional, ferramentas e integração com vector stores. Escolhido por ser o padrão da indústria e possuir ampla documentação e comunidade ativa.

Claude Haiku 4.5 - Anthropic (LLM):

Modelo de linguagem de última geração da Anthropic, otimizado para velocidade e custo-efetividade. Oferece capacidades avançadas de raciocínio, seguimento preciso de instruções complexas e comportamento mais previsível comparado a modelos concorrentes. Ideal para aplicações de produção que exigem respostas rápidas e consistentes.

ChromaDB (Vector Database):

Banco de dados vetorial open-source, leve e fácil de usar. Projetado especificamente para aplicações de IA, permite armazenamento e busca eficiente de embeddings. Escolhido por sua simplicidade de configuração, adequação para POC e capacidade de escalar para produção.

HuggingFace Sentence Transformers (Embeddings):

Biblioteca para geração de embeddings de texto usando modelos transformer. O modelo all-MiniLM-L6-v2 foi escolhido por oferecer excelente equilíbrio entre qualidade, velocidade e tamanho, com suporte multilíngue (incluindo português).

5.2 Interface e Experiência do Usuário

Streamlit (Framework de UI):

Framework Python para criação rápida de aplicações web interativas. Destacam-se as vantagens:

- `Integração Nativa com Python: Sem necessidade de JavaScript ou frameworks front-end complexos`
- `Desenvolvimento Ágil: Ideal para POC e MVPs, permitindo iteração rápida`
- `Componentes Prontos: Widgets interativos (chat, tabs, forms) sem configuração adicional`
- `Deploy Facilitado: Streamlit Cloud e outras plataformas oferecem deploy com poucos cliques`

Pandas (Exportação de Dados):

Biblioteca de análise de dados utilizada para exportação de métricas e conversas em formato CSV, permitindo análises posteriores em ferramentas como Excel.

5.3 Tracking e Observabilidade

Para garantir transparência, debugging eficiente e otimização de custos, o sistema implementa múltiplas camadas de tracking:

Tokens Tracker:

Componente customizado que contabiliza precisamente tokens consumidos em cada chamada ao LLM, distinguindo entre tokens de input, output, cache creation e cache read. Permite análise detalhada de uso por componente.

Pricing Calculator:

Calcula custos em tempo real baseado nas tabelas de preços da Anthropic, fornecendo estimativas precisas de custo por turno, sessão e componente. Essencial para controle de orçamento e ROI.

Retriever Audits:

Sistema de auditoria que registra cada busca no vector store com metadados completos: query enviada, documentos retornados, scores de similaridade, tempo de execução e fonte dos documentos. Permite debugging do sistema RAG e identificação de melhorias.

LangSmith (Opcional):

Plataforma de observabilidade da LangChain que oferece rastreamento end-to-end de todas as interações, incluindo traces de execução, latências e erros. Útil para análise profunda e debugging em produção.

5.4 Processamento de Documentos

Docling (Conversão de PDFs):

Ferramenta avançada de conversão de PDFs para Markdown, preservando estrutura, formatação e metadados. Essencial para processar whitepapers e manuais em PDF mantendo qualidade.

LangChain Document Loaders:

Conjunto de loaders especializados para diferentes formatos de documentos (Markdown, Python, Solidity). Normaliza documentos de diferentes fontes em estrutura uniforme.

LangChain Text Splitters:

Componentes inteligentes de divisão de texto que respeitam limites semânticos (parágrafos, seções) ao invés de simplesmente dividir por caracteres, melhorando qualidade do retrieval.

6. Próximos Passos e Conclusão

6.1 Roadmap de Desenvolvimento

O projeto encontra-se em versão beta funcional, com três frentes principais de evolução planejadas:

Sistema de Log de Conversas de Usuários para Análise

Objetivo:

Coletar e analisar conversas reais de usuários da versão beta para identificar padrões de uso, dúvidas recorrentes, qualidade das respostas e oportunidades de melhoria.

Status Atual:

O sistema já implementa salvamento automático de conversas em formato JSON enriquecido, incluindo mensagens, tokens, custos, audits de busca e metadados de sessão. Cada conversa recebe um ID único e é armazenada em data/conversations/.

Melhorias Propostas:

- Dashboard de Analytics: Visualização agregada de métricas (perguntas mais comuns, tempo médio de resposta, satisfação estimada, tópicos mais consultados)
- Análise de Padrões: Identificação automática de clusters de perguntas similares e gaps de conhecimento
- Sistema de Feedback: Permitir que usuários avaliem respostas (útil/não útil) para refinamento contínuo
- Métricas de Qualidade: Análise de cobertura (perguntas respondidas vs. não respondidas), latência e acurácia baseada em feedback
- Exportação para BI: Integração com ferramentas de Business Intelligence para análises avançadas

Impacto Esperado:

Ciclo de melhoria contínua baseado em dados reais de uso, permitindo refinamento do system prompt, expansão da base de conhecimento e otimização das ferramentas RAG.

Integração com Aplicativo Regeneration Credit

Objetivo:

Integrar o chatbot diretamente no aplicativo móvel/web do Regeneration Credit, oferecendo assistência contextual aos usuários durante sua jornada.

Implementações Necessárias:

- API REST: Desenvolvimento de endpoints RESTful para comunicação entre app e chatbot (POST /chat, GET /history, etc.)
- Autenticação e Autorização: Sistema de tokens JWT para segurança, garantindo que apenas usuários autenticados acessem o chatbot

- Sincronização de Dados: Acesso a dados do perfil do usuário (nível, atividades, saldo) para personalizar respostas
- Contexto Personalizado: Respostas adaptadas ao estado do usuário (iniciante vs. avançado, tipo de usuário, histórico de atividades)
- Deep Links: Chatbot pode sugerir ações específicas no app com links diretos para telas relevantes

Benefícios:

Redução de fricção no onboarding de novos usuários, suporte 24/7 sem necessidade de equipe humana, aumento de engajamento e retenção através de assistência proativa.

Aprimoramento do Sistema RAG

Objetivo:

Melhorar a performance de recuperação (retrieve) do sistema RAG, garantindo que documentos mais relevantes sejam encontrados, enquanto mantém ou reduz o consumo de tokens (custos).

Estratégias Propostas:

1. Re-ranking de Resultados:

Implementar um segundo estágio de ranking usando modelos cross-encoder após a busca inicial. Busca vetorial retorna top-20, cross-encoder reordena e seleciona top-5 realmente mais relevantes.

2. Filtros Adaptativos Inteligentes:

Sistema que analisa a pergunta e automaticamente aplica filtros de metadados (source_type) para reduzir espaço de busca. Ex: "Como minerar?" → aplica filtro para guias técnicos.

3. Cache de Embeddings:

Perguntas frequentes têm embeddings pré-computados, acelerando busca e reduzindo custos de embedding.

4. Otimização de Chunk Size:

Testar diferentes tamanhos de chunk (500, 1000, 1500) e estratégias de overlap para encontrar configuração ótima entre relevância e consumo de tokens.

5. Compressão de Contexto:

Usar modelos de sumarização para comprimir documentos recuperados antes de enviar ao LLM, mantendo informações essenciais, mas reduzindo tokens.

6. Embeddings Híbridos:

Combinar busca densa (embeddings vetoriais) com busca esparsa (BM25/TF-IDF) para melhor cobertura. Especialmente útil para termos técnicos específicos.

Métricas de Sucesso:

Redução de no consumo médio de tokens por consulta, aumento na precisão de retrieval (medido por testes direcionados).

6.2 Considerações Finais

Status Atual do Projeto:

O Regeneration Credit AI Assistant encontra-se em estágio beta funcional, com todas as funcionalidades core implementadas e testadas:

- Sistema RAG completo com 4 ferramentas especializadas e 120+ documentos indexados
- Interface Streamlit profissional com múltiplas abas de funcionalidade e debug
- Tracking detalhado de tokens, custos e performance de retrieval
- Salvamento automático de conversas com metadados enriquecidos
- Sistema de memória conversacional para diálogos contextualizados

Benefícios Alcançados:

Democratização do Acesso: Qualquer pessoa, independente do nível técnico, pode entender conceitos complexos do projeto através de explicações claras e acessíveis.

Redução de Carga de Suporte: Dúvidas recorrentes são respondidas automaticamente 24/7, liberando equipe para questões mais complexas.

Onboarding Acelerado: Novos usuários conseguem aprender sobre o projeto de forma autoguiada e interativa, reduzindo fricção de entrada.

Base para Evolução: Arquitetura modular e tracking detalhado permitem iterações rápidas baseadas em dados reais de uso.

Potencial de Expansão:

O sistema foi projetado para evoluir além de um chatbot de suporte. Possibilidades futuras incluem:

- Assistente de Onboarding: Guiar novos usuários passo a passo através do processo de cadastro e primeiras ações
- Calculadora Interativa: Simular cenários de tokenomics e distribuição de forma conversacional
- Suporte Multilíngue: Expandir para outros idiomas aproveitando capacidade multilíngue dos modelos
- Agente Proativo: Enviar notificações e dicas personalizadas baseadas no perfil e atividades do usuário
- Integração com Blockchain: Consultar dados on-chain em tempo real para respostas ainda mais precisas e atualizadas

O Regeneration Credit AI Assistant representa um passo importante na democratização do acesso à informação sobre o projeto. A base construída permite evolução contínua e adaptação às necessidades emergentes da comunidade Regeneration Credit.
