Documento Técnico – Arquitetura do Chatbot Inova

1. Visão Geral e Padrão Arquitetural

O sistema é projetado como um Monolito Modular. As responsabilidades são segregadas em módulos bem definidos, facilitando a manutenção e a evolução futura do projeto.

A arquitetura é composta por cinco camadas principais: Frontend, Balanceador de Carga, Backend, Processamento Assíncrono e Persistência de Dados.

2. Componentes da Arquitetura

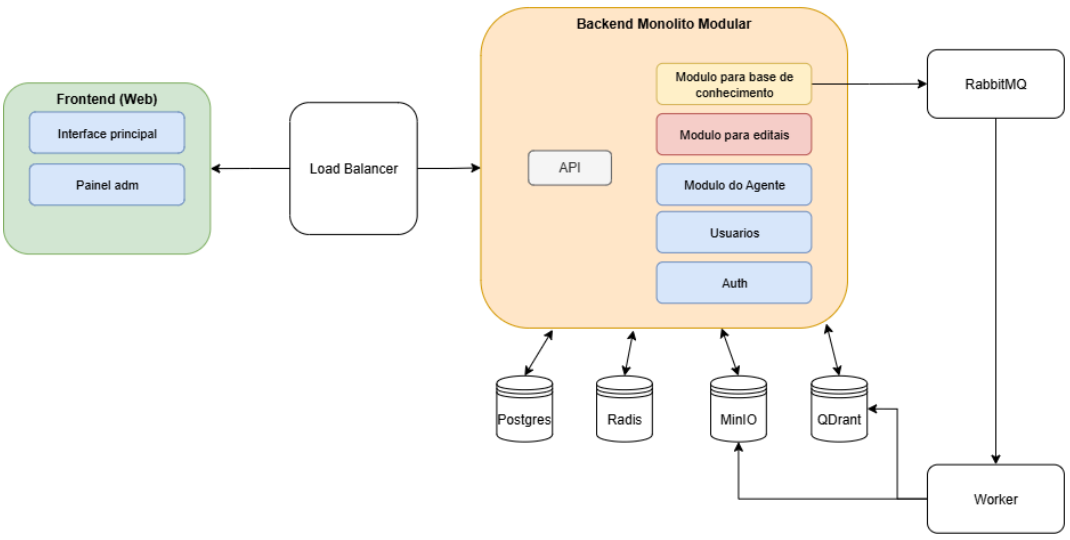
- Frontend (Web): A camada de interação com o usuário, composta por duas aplicações distintas:
    - Chatbot Inova: Interface principal de conversação para pesquisadores e inventores.
    - Painel Administrativo: Interface restrita para a gestão da base de conhecimento, monitoramento e curadoria de conteúdo.
- Balanceador de Carga (Load Balancer): Atua como ponto de entrada único para todo o tráfego, distribuindo as requisições de forma eficiente entre as instâncias do backend. Garante alta disponibilidade e facilita a escalabilidade horizontal da aplicação.
- Backend (Monolito Modular): O núcleo da aplicação, que centraliza a lógica de negócio.
    - API: estrutura e endpoints de comunicação da api.
    - Módulos de Negócio:
        - Auth: Gerencia a autenticação e autorização de usuários.
        - Usuários: Responsável pelo gerenciamento de perfis e dados dos usuários.
        - Módulo de Chat: Orquestra a lógica conversacional. É aqui que reside o sistema multi-agente (Orquestrador, Pré-processamento, etc.) para interpretar, processar e responder às queries dos usuários.
    - Módulos de Tarefas Pesadas:

    - Módulo para base de conhecimento: Lida com o upload de novos documentos.
    - Módulo para editais: Responsável por executar a busca de novos editais em fontes externas.

  Estes módulos iniciam tarefas longas, mas delegam sua execução para a camada de processamento assíncrono.

- Camada de Processamento Assíncrono: Essencial para garantir que a API principal permaneça rápida e responsiva, desacoplando tarefas demoradas.
    - Fila (Queue): Um sistema de enfileiramento que atua como um buffer. Os módulos de crawler e base de conhecimento publicam tarefas nesta fila.
    - Worker: Um processo computacional independente que consome as tarefas da fila. Ele executa a lógica pesada, como a conversão de documentos, geração de embeddings e a inserção nos bancos de dados (MinIO e BD Vetorial).
- Camada de Persistência de Dados: Utiliza uma abordagem de persistência poliglota, selecionando a melhor tecnologia para cada tipo de dado.
    - Postgres: Armazena dados relacionais, como usuários, permissões, histórico de conversas e metadados de editais.
    - Cache: Armazena dados de acesso rápido para reduzir a latência, como sessões de usuário e respostas de perguntas frequentes.
    - Storage: Armazena objetos e arquivos brutos, como os documentos originais da base de conhecimento.
    - QDrant: Armazena os embeddings vetoriais dos documentos, sendo o pilar para a funcionalidade de busca por similaridade semântica.

Arquitetura - Chatbot Inova



3. Sistema Multi-Agente

O núcleo do chatbot é composto por uma arquitetura de agentes especializados, orquestrados para lidar com as diferentes jornadas do usuário de forma eficiente.

- Fluxo de Processamento de Requisições:
    1. Entrada do Usuário: A pergunta do usuário é recebida pela API.
    2. Agente de Pré-processamento: A query é normalizada, limpa e entidades-chave (ex: INPI, edital, patente) são reconhecidas para enriquecer o contexto.
    3. Agente Orquestrador: Interpreta a intenção principal do usuário e delega a tarefa ao agente especialista mais adequado.
    4. Agentes Especialistas: Cada agente consulta suas ferramentas e executa sua lógica para atender a uma jornada funcional específica, como orientação sobre PI, análise de risco, busca por fomento ou redação de patente.
    5. Agente de Pós-processamento: Refina a resposta final para garantir clareza, consistência e segurança, prevenindo a entrega de informações inventadas.
    6. Entrega da Resposta: A resposta final é formatada e enviada para a interface do usuário.

4. Pipelines

- Pipeline de Ingestão e Indexação de Documentos (Base de Conhecimento):
    1. Recepção: O documento é recebido via painel administrativo.
    2. Armazenamento Bruto: O arquivo original é salvo em um object storage (MinIO).
    3. Pré-processamento: O documento é convertido, limpo e dividido em segmentos menores (*chunks*).
    4. Vetorização e Indexação: Embeddings vetoriais são gerados para cada *chunk* e inseridos no banco vetorial (Qdrant) para busca semântica.
- Pipeline de Recuperação de Informação (Retrieval/RAG):

1. Vetorização da Query: A pergunta do usuário é transformada em um embedding vetorial.
2. Busca Híbrida: Uma busca combina similaridade semântica e busca por palavras-chave no Qdrant para encontrar os *chunks* mais relevantes.
3. Contextualização: Os *chunks* recuperados são enviados ao agente especialista como contexto para a geração da resposta final.

5. Stack Tecnológica

- Frontend:
  - Linguagem: JavaScript/TypeScript
  - Framework: React
- Backend:
  - Linguagem: Python
  - Framework API: FastAPI
  - Orquestração de Agentes: LangChain + LangGraph
- Persistência e Cache:
  - Banco Vetorial: Qdrant
  - Banco Relacional: PostgreSQL
  - Cache: Redis
  - Object Storage: MinIO
- Infraestrutura:
  - Conteinerização: Docker
  - Balanceamento de Carga: NGINX
  - Fila de Mensagens: RabbitMQ
  - Hospedagem: pelo próprio INF

6. Observabilidade e Avaliação

Para garantir a confiabilidade e a melhoria contínua do sistema, uma estratégia de observabilidade será implementada.

- Logging Estruturado: Logs detalhados e auditáveis serão gerados para rastrear o fluxo de cada conversa, incluindo a intenção identificada, os agentes acionados, os documentos recuperados e a resposta final. Isso é crucial para a depuração de falhas e a análise de performance.

7. Segurança

A segurança é um pilar fundamental do projeto, dado o manuseio de dados sensíveis e propriedade intelectual.

- Autenticação e Autorização: Implementação de um sistema de autenticação de usuários para proteger o acesso e o histórico de conversas. O painel administrativo terá controles de acesso baseados em perfis.
- Política de Privacidade: Uma política de privacidade clara e acessível será disponibilizada aos usuários.
- Sessões Anônimas: O sistema oferecerá a opção de uso anônimo, sem armazenamento de histórico, para aumentar a confiança do inventor.

8. Escalabilidade

A arquitetura foi projetada para suportar o crescimento do número de usuários e do volume de dados.

- Processamento Assíncrono: Tarefas pesadas, como a indexação de novos documentos, serão executadas em filas de processamento para não impactar a performance do sistema principal.
- Indexação Incremental: A base de conhecimento suportará a adição de novos documentos de forma incremental, sem a necessidade de reindexar todo o conteúdo.