



1º Desafio de Inovação Aberta na CGE

Desafio 2

Como a Controladoria-Geral poderia ter acesso a todo o conhecimento já produzido de forma automatizada e responsivo

www.makes.ai



Proposta de solução para Desafio 2 da CGE/GO

Nossa solução visa permitir que os servidores da Controladoria-Geral acessem, analisem e utilizem rapidamente todo o conhecimento acumulado pela instituição. A solução proposta é um chat com LLM em um sistema robusto que integra tecnologias avançadas de processamento de dados e inteligência artificial para automatizar a coleta, organização, recuperação de informações e processamento de forma simples como conversar com uma IA.

Componentes Técnicos da Solução

Resumo detalhado

O pipeline de desenvolvimento envolve a coleta de dados através de integrações com sistemas internos e migração de documentos legados para um ambiente centralizado com Apache NIFI e Spark. Após a coleta, os dados são armazenados em um data lake com Hive e Hadoop, facilitando o acesso e a recuperação com Elasticsearch. A segurança é mantida por sistemas de controle de acesso com Apache Ranger e Keycloak, garantindo permissões específicas para cada nível de confidencialidade.

O pipeline inclui a criação de metadados para os arquivos a partir do uso de modelos open source de LLM para criação de um Knowledge Graph e a comunicação dessas informações com agentes de conversação usando IA generativa open source, integrando APIs para dados em tempo real e coleta automatizada com Airflow e Kafka. A arquitetura modular permite adaptações a novos tipos de documentos e escalabilidade com MongoDB e Cassandra, além de uma infraestrutura gerenciada com Kubernetes e Docker.

A implementação de Retrieval Augmented Generation (RAG), técnicas avançadas de featurização semântica, e o uso de LLM em um contexto governamental é pioneira.

- **Transformação de Processos:** A solução modifica radicalmente o processo de recuperação de informações, tornando-o mais eficiente e acessível.
- **Impacto Significativo:** A capacidade de acessar rapidamente informações relevantes pode significativamente melhorar a qualidade e a velocidade do trabalho dos servidores, beneficiando a administração pública e a sociedade como um todo.
- **Sustentabilidade e Escalabilidade:** O uso de um sistema que aprende e se adapta continuamente garante a relevância a longo prazo da solução.

A solução para a Controladoria-Geral do Estado de Goiás foi proposta para simplificar e agilizar o fluxo de dados, permitindo que as informações sejam coletadas, processadas e apresentadas de forma eficiente e precisa. Esse pipeline é composto para garantir que cada dado capturado seja tratado com o máximo de segurança e utilidade, facilitando o acesso e a análise para uma tomada de decisão mais informada.

Os processos e dados da Controladoria-Geral do Estado (CGE) bem como os serviços privados em nuvem são essenciais para garantir uma automação eficaz da coleta e tratamento de dados. A estratégia de integração abrange vários aspectos técnicos e inovadores para assegurar uma implementação suave e funcional.



Com um projeto de uma arquitetura modular e escalável, ela pode ser adaptada para outros tipos de processos típicos da atividade de controle interno. A flexibilidade e a capacidade de expansão são considerações centrais, permitindo que a solução evolua conforme as necessidades da CGE e novas tecnologias se tornam disponíveis.

Quanto à escalabilidade, foi projetado especialmente no que diz respeito à capacidade de integrar e processar dados de diversas fontes. Esta flexibilidade é crucial para garantir que a solução possa se adaptar e expandir conforme as necessidades de dados da CGE evoluam ou novas fontes de dados se tornem disponíveis.

Adaptabilidade a Outras Fontes de Dados: A solução pode ser adaptada para incorporar uma ampla variedade de fontes de dados, incluindo, mas não se limitando a, bancos de dados governamentais adicionais, fontes de dados financeiros, redes sociais para análise de sentimentos, e bases de dados de compliance e regulamentação. A arquitetura modular e o uso de tecnologias de integração flexíveis permitem que a solução se ajuste facilmente para incluir essas novas fontes de dados.

Pipeline

A. Coleta de Dados

Análise e Mapeamento: Identificação de fontes de informação internas e externas. Integração com Sistemas Existentes: Utilização do Apache NiFi para conectar com sistemas como o SEI e extrair documentos de forma contínua.

Importação de Dados Legados: Emprego de PySpark, Spark e Apache Spark para migrar dados históricos para um ambiente centralizado.

B. Indexação, Armazenamento e Metadados

Data Lake: Utilização do Apache Hive e Apache Hadoop para armazenar dados processados, garantindo escalabilidade e flexibilidade.

Indexação: Uso do Elasticsearch para acelerar a recuperação de informações através de técnicas avançadas de indexação.

Metadados: Implementação de um modelo de Knowledge Graph para armazenar metadados que facilitam a organização e recuperação de informações.

C. Controle de Acesso e Segurança

Classificação de Documentos: Utilização do Apache Ranger para etiquetar documentos com níveis de confidencialidade.

Gerenciamento de Permissões: Implementação do Keycloak para sistemas de autenticação e autorização.

D. Aprendizado Contínuo e Cache

Machine Learning e LLM: Algoritmos de aprendizado contínuo para aprimorar as buscas, utilizando ferramentas como Pytorch e HuggingFace. O uso de modelos de linguagem de grande escala (LLM) permite uma compreensão profunda do conteúdo e a geração de respostas contextuais precisas.

Cache: Implementação de sistemas de cache para otimizar o desempenho e reduzir custos.

Integração

Integração com o SEI e Outros Sistemas:

APIs e Web Services: Uso de APIs para conectar-se ao SEI e extrair documentos em tempo real. Ferramentas: APIs Resftul e Apache Camel.

Conectores Personalizados: Desenvolvimento de conectores específicos para sistemas legados que não possuem interfaces de integração padrão.

Automatização da Coleta de Dados Ferramentas: Apache Airflow e Apache Kafka;

Agentes de Coleta: Implementação de agentes que monitoram continuamente os sistemas fonte em busca de novos documentos. Ferramentas: Logstash

Agendamento de Processos: Utilização de ferramentas de agendamento para executar tarefas de coleta e processamento em horários pré-definidos. Ferramentas: Apache Airflow e Apache Kafka

Uso de Nuvem Privada: On-premisse e self-hosted

Escalabilidade Vertical e Horizontal: Capacidade de ajustar recursos computacionais conforme a demanda. Ferramentas: Kubernetes e Docker

Integração com Serviços Externos (se aplicável):

APIs de Serviços em Nuvem: Integração com serviços de processamento em nuvem, como reconhecimento de linguagem ou tradução, se necessário.

Compliance e Segurança: Garantia de que todos os dados transitem de forma segura e em conformidade com as legislações vigentes.

Automatização do Tratamento de Dados:

Pipeline de Dados Automatizado: Fluxo contínuo desde a coleta até o armazenamento, sem necessidade de intervenção manual. Ferramentas: Apache Airflow, Apache Kafka, Apache Beam

Monitoramento e Logging: Implementação de sistemas de monitoramento para garantir a integridade e disponibilidade dos dados. Ferramentas: Grafana, Metabase, etc;

Resiliência

Adaptabilidade a Novos Tipos de Documentos e Processos:

Arquitetura Modular: Componentes independentes que podem ser ajustados ou ampliados sem afetar o sistema como um todo. Ferramentas: java, spring boot. Padrões: Microserviços, software

Desenvolvimentos Adicionais Necessários:

Integração com Novos Sistemas: Desenvolvimento de conectores adicionais para integração com outros sistemas internos ou externos. Comunicação entre serviços no agrupamento do serviço;

Escalabilidade

Integração com Outras Fontes de Dados:

Conectores Flexíveis: Desenvolvimento de conectores capazes de se comunicar com diferentes tipos de bancos de dados e sistemas de arquivos. Ferramentas: API Resftul, Pacotes de comunicação com diferentes bancos de dados.

Desenvolvimentos Adicionais Necessários:

Mapeamento de Dados: Criação de mapas de dados para assegurar que as novas fontes sejam corretamente interpretadas e integradas. Ferramentas: Validadores e contratos de entrada e saída explícitos entre cada serviço.

Escalonamento da Infraestrutura: Ajustes na infraestrutura para suportar o aumento no volume de dados, incluindo processamento e armazenamento. Ferramenta: MongoDB, Cassandra auto escalonamento horizontal.

Tecnologias Envolvidas:

Big Data Frameworks: Uso de tecnologias como Hadoop ou Spark para processamento em larga escala.

Serviços de Mensageria: Implementação de filas e tópicos para gerenciar fluxos de dados em tempo real.

Testes

1. Planejamento dos Testes:

Definição do Escopo: Identificação dos componentes e funcionalidades que serão testados.

Metodologia: Escolha das técnicas de teste, como testes automatizados e manuais.

Monitoramento: Sonarqube

2. Testes Unitários:

Cobertura de Código: Garantir que todas as partes críticas do código sejam testadas individualmente.

Ferramentas Utilizadas: Uso de frameworks como JUnit ou PyTest para automatização dos testes.

3. Testes de Integração:

Verificação de Interfaces: Testar a comunicação entre os módulos internos e sistemas externos.

Cenários de Uso: Criação de cenários que refletem o uso real do sistema.

4. Testes Funcionais:

Validação de Requisitos: Assegurar que todas as funcionalidades atendam aos requisitos especificados.

Testes de Interface: Avaliar a usabilidade e a experiência do usuário na interação com o sistema.

5. Testes de Desempenho:

Testes de Carga: Avaliar como o sistema se comporta sob alto volume de usuários ou dados.

Testes de Estresse: Identificar o ponto de falha do sistema sob condições extremas.

6. Testes de Segurança:

Análise de Vulnerabilidades: Uso de ferramentas para identificar possíveis brechas de segurança.

Testes de Acesso: Verificação dos mecanismos de autenticação e autorização.

7. Testes de Aceitação:

Feedback dos Usuários: Envolvimento de usuários finais para validar se a solução atende às necessidades.

Ajustes Finais: Correção de problemas identificados durante os testes de aceitação.

8. Entregáveis ao Longo do CPSI:

Relatórios de Teste: Documentação dos resultados de cada fase de testes.

Manual do Usuário: Guia para auxiliar os usuários na utilização da solução.

Documentação Técnica: Detalhamento da arquitetura, componentes e interfaces do sistema.

9. Monitoramento Pós-Implantação:

Logs e Alertas: Implementação de sistemas de monitoramento para detecção precoce de problemas. Ferramentas: Git Actions, pre-commit

Revisões Periódicas: Sessões de acompanhamento para avaliar o desempenho contínuo da solução.

Tempo de Desenvolvimento

1. Fase de Planejamento e Concepção:

- Duração: 1 mês
- Atividades: Definição de requisitos, planejamento de projeto, e estabelecimento de metas e objetivos com Stakeholders.

2. Fase de Desenvolvimento e Integração:

- Duração: 7 meses
- Atividades: Desenvolvimento de software, criação de conectores de dados, integração com sistemas existentes, e implementação de funcionalidades de processamento e análise de dados.

3. Fase de Testes e Otimização:

- Duração: 3 meses
- Atividades: Testes unitários, testes de integração, testes de desempenho, e otimização de sistemas para garantir que a solução atenda aos requisitos técnicos, de desempenho e ajustes de performance e qualidade.

4. Fase de Implementação e Treinamento:

- Duração: 1 mês
- Atividades: Implementação final da solução no ambiente de produção da CGE, treinamento de usuários, e documentação da solução.

5. Fase de Monitoramento e Ajustes Finais:

- Duração: 2 meses
- Atividades: Monitoramento da solução em operação, coleta de feedback dos usuários, e realização de ajustes finais para otimizar a performance e usabilidade.

Tempo Total Estimado:

- 14 meses

Justificativa para o Cronograma

A complexidade das integrações e a necessidade de desenvolver uma solução altamente adaptável e segura justificam um período de desenvolvimento mais extenso. Cada fase foi cuidadosamente planejada para garantir que todos os aspectos da solução sejam meticulosamente testados e otimizados.

A fase de monitoramento e ajustes finais é crucial para assegurar que a solução funcione eficazmente em um ambiente real e atenda às expectativas dos usuários finais.

A fase de testes é crucial para garantir que aos padrões de qualidade e segurança exigidos, além de confirmar sua funcionalidade e eficácia. A seguir, detalho uma estratégia de testes abrangente, incluindo atividades específicas e os entregáveis associados ao longo do Contrato Público de Solução Inovadora (CPSI).



Estratégia de Monetização

Serviços de Consultoria e Customização: Oferecimento de serviços de consultoria para personalização da solução de acordo com as necessidades específicas da CGE, assim como para integração com outros sistemas e treinamento de usuários.

Suporte e Manutenção: Contrato de suporte e manutenção contínuos para garantir o funcionamento eficiente da solução, com uma equipe dedicada para resolver quaisquer problemas técnicos que possam surgir.



Faturamento

Pagamentos por Serviços Adicionais: Cobrança separada por serviços de consultoria, customização e treinamentos especiais, proporcionando flexibilidade para a CGE escolher apenas os serviços de que necessita.

Composição e Qualificação da Equipe

- Lucas Oliveira Figueiredo: Diretor de tecnologia e desenvolvedor de IA.
- Daniel Ide Futata: Desenvolvedor de IA
- Heitor Marques: Desenvolvedor de IA
- David Occhioni Excel: Desenvolvedor de IA
- Leon Balloni: Desenvolvedor backend, arquiteto de software e tech-lead
- Anna Neves: Engenheira de infra-estrutura de software
- Pedro Faria: Desenvolvedor backend
- Renan Yudi: Desenvolvedor full-stack
- Rafael Mota: Desenvolvedor full-stack
- Gustavo Veiga Zani: Diretor de projetos e gerente de produtos
- Lucas Vaz: Diretor de Inovação
- Henrique Sabino: Designer de UI/UX



Cronograma Detalhado

Fase de Planejamento e Concepção (1 mês): Nesta fase inicial, definiremos detalhadamente os requisitos do sistema, traçaremos o escopo do projeto e selecionaremos as tecnologias mais adequadas para sua construção. Também será elaborado um plano de projeto completo, incluindo cronograma e recursos necessários.

Fase de Desenvolvimento e Integração (7 meses): Nessa etapa, daremos vida ao software, desenvolvendo os módulos e integrando-os aos sistemas já existentes. Serão realizados testes iniciais para garantir a qualidade do código.

Fase de Testes e Otimização (3 meses): Com o software desenvolvido, realizaremos uma série de testes para garantir que ele funcione conforme o esperado. Serão feitos testes unitários, de integração, de sistema e de aceitação do usuário. Além disso, otimizaremos o desempenho do sistema para garantir uma experiência fluida para os usuários.

Fase de Implementação e Treinamento (1 mês): Após os testes, o sistema será colocado em produção. Nessa fase, treinaremos os usuários para que possam utilizar o sistema de forma eficiente e elaboraremos a documentação necessária.

Fase de Monitoramento e Ajustes Finais (2 mês): Após a implementação, iniciaremos o monitoramento contínuo do sistema para identificar e corrigir possíveis problemas. Coletaremos o feedback dos usuários para realizar os ajustes finais e garantir a satisfação dos mesmos.

Fase de Manutenção e Suporte (Contínuo): Essa fase é essencial para garantir a longevidade do sistema. Nela, forneceremos suporte técnico aos usuários, realizaremos atualizações de software e faremos a manutenção preventiva e corretiva do sistema.

Proposta Financeira

Fase de Planejamento e Concepção (1 mês): R\$ 100.000
Fase de Desenvolvimento e Integração (7 meses): R\$ 700.000.
Fase de Testes e Otimização (3 meses): R\$ 300.000.
Fase de Implementação e Treinamento (1 mês): R\$ 100.000.
Fase de Monitoramento e Ajustes Finais (2 mês): R\$ 200.000.

Custo Total: 1.400.000,00

Detalhamento dos Custos

Planejamento e Concepção: Inclui custos de consultoria para definição de requisitos, planejamento estratégico e seleção de tecnologias e salários da equipe. Este valor cobre também as despesas administrativas iniciais e reuniões com stakeholders.

Desenvolvimento e Integração: Abrange salários da equipe de desenvolvimento, custos de licenças de software e ferramentas de desenvolvimento, bem como despesas relacionadas à integração de sistemas e testes iniciais.

Testes e Otimização: Inclui custos com ferramentas de teste, serviços de terceiros para revisões de segurança, salários da equipe e otimização baseada nos resultados dos testes.

Implementação e Treinamento: Engloba despesas com a implementação final no ambiente de produção, salários da equipe, treinamento dos usuários finais, e produção de manuais e material de apoio.

Monitoramento e Ajustes Finais: Cobre o monitoramento da solução, salários da equipe, coleta de feedback dos usuários e ajustes finais baseados neste feedback.

Manutenção e Suporte: Estima os custos anuais com suporte técnico, atualizações regulares de software, e manutenção preventiva e corretiva.