DESAFIO 1 – Como a CGE pode otimizar o monitoramento das compras do estado de Goiás para garantir que os órgãos estejam atendendo aos requisitos estabelecidos para o processo de compras?

Jonathan Santana Silva
(11) 95167-4485
Rua Passo Fundo – CPA1 – Cuiabá – MT

1. Solução Inovadora

Sistema de processamento de dados distribuídos, interpretados, categorizados e analisados utilizando Machine Learning. Com avisos por e-mail e insight inteligentes.

Interface Web: SPA com Vue.js e API Gateway com Java Spring Boot e Postgres ou Oracle.
Motor de processamento e aprendizado: Airflow, Spark, Nessie Catalog, Trino, Kafka, Spark MLlib, Llama e API Python com FastAPI e docker.

2. Proposta

A proposta traz um sistema de processamento de dados distribuído e inovador que combina componentes avançados de big data, aprendizado profundo (deep learning), processamento em larga escala e uma interface simplificada. Esse sistema é estruturado para captar, organizar, interpretar e analisar dados em diferentes formatos e contextos, com uma aplicação prática em análise preditiva e inteligência artificial voltada para decisões informadas.

Arquitetura Distribuída de Alta Escalabilidade:
A integração de Airflow, Spark e Kafka permite uma abordagem distribuída de processamento de dados, onde o sistema pode lidar com grandes volumes de dados em tempo real, desde a coleta até o armazenamento e processamento. O Java Spring Boot serve como uma API Gateway que organiza os dados e interage com o front-end de uma maneira eficiente e rápida.

Esse ecossistema de tecnologias, orientado ao processamento em lote e streaming, proporciona uma alta disponibilidade e escalabilidade horizontal. Dessa forma, o sistema pode crescer conforme as demandas de processamento aumentam, garantindo a manutenção da performance e robustez.

Inteligência Artificial com Deep Learning Integrado:
A utilização de Llama e frameworks de deep learning permite que o sistema não apenas processe dados, mas também aplique aprendizado profundo para interpretações contextuais e classificações automatizadas. Esse diferencial adiciona um aspecto preditivo e interpretativo aos dados, o que eleva a capacidade de gerar insights profundos e personalizados para o usuário.

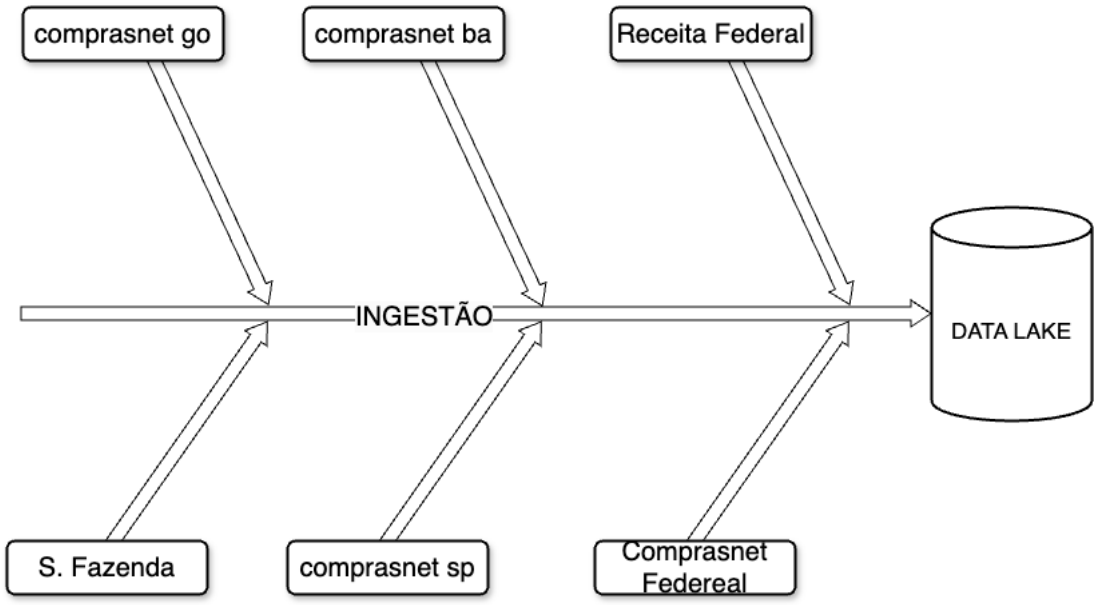
A integração com Spark MLlib e ferramentas como Nessie Catalog e Trino facilita o gerenciamento, a governança e a consulta de dados massivos, proporcionando uma análise ágil e detalhada com potencial de personalização para diferentes cenários e usuários.

Interface Web Moderna e Interativa:
A interface SPA (Single Page Application) em Vue.js garante uma experiência de usuário fluida e responsiva. Esta camada melhora a acessibilidade e a usabilidade do sistema, além de facilitar a visualização de dados complexos de maneira intuitiva e dinâmica, aprimorando a experiência de consulta e exploração de dados.

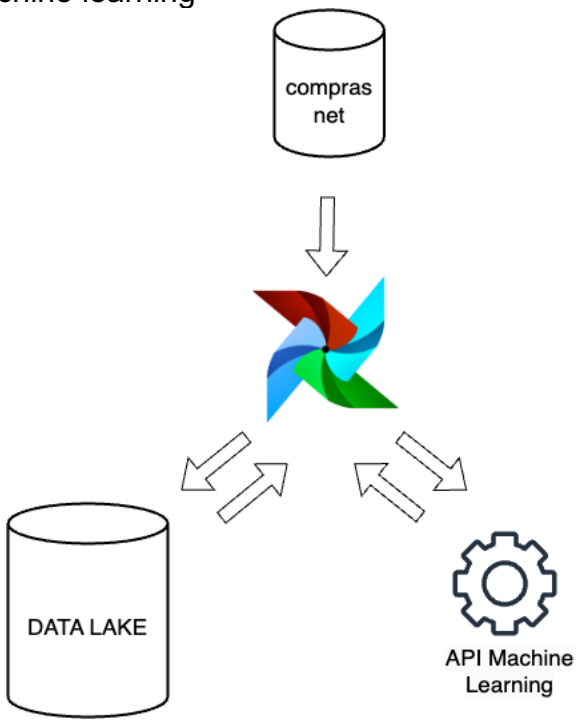
O uso de Nessie como catálogo de dados em conjunto com o Trino oferece uma estrutura de governança robusta, onde cada mudança no conjunto de dados pode ser versionada, auditada e rastreada. Esta estrutura é altamente inovadora, pois une a governança ao processamento em tempo real, otimizando a consistência e a confiança nos dados para análises e relatórios.

3. Solução

Utilizando o Airflow fontes de dados relacionada ao processo licitatório serão observadas, a cada período Jobs vão ser executado para realização da ingestão dos dados no datalake CGE-GO, na sequencia o airflow executará jobs para analise dos dados:



Para cada licitação, os item serão percorridos, e para cada item, será utilizado o serviço da api machine learning



Em um contexto geral a solução conta com a ingestão de dados de fontes distintas para o datalake, criando grande volume da dados com objetivo primário de treinar o modelo de Machine Learning.

4. Processo

Atenção! Essa proposta contempla o desenvolvimento Ágil, utilizamos técnicas do Manifesto Ágil para entregar valor ao cliente de forma rápida.

4.1. Desenvolvimento

a. Planejamento e Design da Arquitetura

Definição de Requisitos: Aprendizado profundo dos requisitos. Envolve a criação de requisitos técnicos e funcionais, como volume de dados esperado e modelos de machine learning.

Design da Arquitetura de Dados: Criação de uma estrutura de dados distribuída e modular. Define-se o pipeline de dados com Airflow, a governança com Nessie Catalog e o processamento com Spark, Trino e Kafka. Além disso, a integração de Llama e Spark MLlib é planejada para atender as necessidades de deep learning.

b. Desenvolvimento dos Componentes

Criação dos Pipelines de Dados no Airflow: Desenvolvimento dos workflows de coleta, processamento e armazenamento de dados. Inclui a criação de DAGs para gerenciar o fluxo de dados entre Spark, Kafka, Trino e Nessie.

Implementação de Modelos de deap learning: Configuração dos modelos em Llama e Spark MLlib, ajustados ao tipo de dados e aos requisitos analíticos do sistema.
Desenvolvimento da Interface e API: Desenvolvimento do front-end SPA em Vue.js
e da API Gateway em FastAPI. A API é projetada para gerenciar requisições de dados e envio de análises para o front-end.

c. Testes e Validação

Testes Unitários e de Integração: Teste de cada componente individualmente (e.g., pipelines de dados, modelos de aprendizado profundo) e integração entre as tecnologias. Isso inclui testar o fluxo de dados completo e verificar a escalabilidade.

4.2. Implementação

Configuração de Ambiente de Produção
Implantação de Contêineres e Orquestração: Uso de contêineres para cada serviço (Airflow, Spark, Kafka, Trino, Nessie, FastAPI) para facilitar a escalabilidade e a gestão de recursos. Uma ferramenta de orquestração, como Kubernetes, pode ser usada para gerenciar a escalabilidade automática.
Configuração de Dados e Segurança: Estabelecimento de mecanismos de segurança (e.g., autenticação, controle de acesso) e configuração do catálogo de dados com o Nessie para gerenciar versões e governança de dados.

Início da Coleta e Processamento de Dados
Coleta de Dados e Ingestão: Airflow ativa o processo de ingestão de dados de fontes externas, seja em lotes ou streaming via Kafka. Nessie e Trino facilitam a catalogação e consulta dos dados no pipeline.
Processamento e Análise com Spark e Modelos de Aprendizado Profundo: Após a ingestão, o Spark aplica transformações e análises aos dados. Os modelos em Llama e Spark MLlib são usados para categorizar e interpretar os dados, entregando insights analíticos.
c. Exposição dos Dados Processados
Entrega ao Front-End: Os resultados das análises são expostos por meio da FastAPI e enviados ao front-end SPA em Vue.js. O front-end exibe visualizações e insights em tempo real, proporcionando uma experiência interativa para o usuário final.

4.3. Monitoramento e Manutenção

a. Monitoramento em Tempo Real
Monitoramento dos Pipelines de Dados com Airflow: O Airflow facilita o acompanhamento dos workflows em tempo real, detectando falhas e gargalos de processamento. Métricas como tempo de execução e falhas são registradas para manter a eficiência do pipeline.
Logs e Alertas: Implementação de logs e alertas automáticos em Kafka e Trino para monitorar a integridade dos dados. Alertas notificam a equipe de suporte sobre qualquer falha nos serviços críticos.

b. Ajustes e Retraining dos Modelos de Deep Learning
Avaliação e Ajuste dos Modelos de Aprendizado: A performance dos modelos de aprendizado profundo é constantemente revisada. Caso necessário, novos dados são utilizados para treinar novamente os modelos em Spark MLlib e Llama, melhorando a precisão e mantendo a relevância dos insights.
Atualizações da Interface e API: Eventuais ajustes na interface ou na API são implementados com base no feedback dos usuários e no desempenho da aplicação, garantindo que o sistema continue intuitivo e funcional.

c. Melhoria Contínua e Escalabilidade

Escalabilidade Vertical e Horizontal: À medida que o volume de dados aumenta, o sistema pode ser escalado horizontalmente (adição de mais contêineres) ou verticalmente (aumento de recursos em máquinas existentes). Essa flexibilidade garante a continuidade e eficiência do sistema.
Documentação e Backup Regular: Toda a estrutura, incluindo pipelines, modelos e arquitetura de dados, é documentada para suporte contínuo e backup periódico, garantindo a segurança e a preservação dos dados e configurações.

5. INTEGRAÇÃO, RESILIÊNCIA e ESCALABILIDADE

Escolhemos utiliza o javaspring boot justamente pela facilidade de implementação com outras aplicações Single Sign-On (SSO). A solução pensada para ficara sob a infraestrutura da CGE “on premise”. Não iremos utilizar serviços privados de terreiros, todos os componentes são open source.

A solução é altamente adaptável para outros processos típicos da atividade de controle interno, com desenvolvimentos adicionais relativamente incrementais. A modularidade dessa arquitetura permite que novos fluxos de dados e regras de análise sejam adicionados sem mudanças estruturais significativas. Abaixo, explico como essa adaptabilidade se aplica a diferentes áreas de controle interno e quais desenvolvimentos adicionais seriam necessários.

A Solução é adaptável a outras fontes de dados, por conta da arquitetura flexível e modular.

6. Time

Por enquanto a equipe é formada por um desenvolvedor e um Engenheiro de Dados:
Jonathan Santana Silva – 29 anos, Paulistano, Bacharel em Sistemas de Informação, AWS Certified Cloud Practitioner, Desenvolvedor de Sistemas com mais de 8 anos de experiencias no desenvolvimento de softwares dos seguimento da indústria, comercio, bancário e publico.

Vitor Henrique Oliveira de Jesus – 29 anos, Cuiabano, Bacharel em Ciências da Computação pela faculdade UFMT, cursando Pós-graduando em Gestão e Ciência de Dados na faculdade UFMT.

7. Cronograma e esforço estimado

1. Estrutura do Cronograma

Sprint	Duração (Semanas)	Fase	Entregáveis	Custo Estimado (R$)
1	2	Preparação, Levantamento de Requisitos e Planejamento	- Backlog inicial - Documento de visão - Protótipo inicial do sistema - Requisitos priorizados - Arquitetura inicial - Configuração do ambiente de desenvolvimento e homologação	90.000
2	2	Levantamento de Requisitos	- Módulo de autenticação implementado	60.000
3	2	Desenvolvimento de Pipelines ETL	- Pipeline 1 (fonte de dados) implementado	80.000
4	2	Desenvolvimento de Pipelines ETL	- Pipeline 2 (fonte de dados) implementado	60.000
5	2	Desenvolvimento de Pipelines ETL	- Pipeline 3 (fonte de dados) implementado	60.000
6	2	Desenvolvimento de Modelos ML	- Modelo de ML - Versão inicial criada	60.000
7	2	Desenvolvimento de Modelos ML	- Ajustes no modelo de ML e documentação	60.000

Sprint	Duração (Semanas)	Fase	Entregáveis	Custo Estimado (R$)
8	2	Implementação do Sistema de Filas	- Sistema de filas configurado e funcional	30.000
9	2	Desenvolvimento de Dashboards	- Dashboard 1 criado e funcional	50.000
10	2	Desenvolvimento de Relatórios	- Relatório 1 automatizado criado	50.000
11	2	Testes de Integração	- Testes de integração concluídos	50.000
12	2	Ajustes Finais e Feedback	- Ajustes baseados em feedback dos testes	50.000
13	2	Treinamento e Entrega	- Treinamento da equipe - Documentação completa	40.000
Total	**6 meses**	-	-	**R$ 990.000**