

Relatório do Projeto Final: LançAI – Agente Autônomo de Automação Contábil-Fiscal

Nome do Grupo: Agentes Autônomos – Projeto final

Integrantes do Grupo:

Anny Beatriz Teixeira

Ramon Ribeiro Rabello

Juliana Rodrigues

Descrição do Tema Escolhido

O projeto **LançAI** consiste no desenvolvimento e implementação de um **Agente Autônomo de Automação Contábil-Fiscal**. O objetivo primário é realizar a geração automática de lançamentos contábeis a partir da extração e do mapeamento inteligente de dados de documentos fiscais eletrônicos (XML de NF-e, CT-e, etc.).

A solução foi desenvolvida com foco no **Setor Industrial Metalúrgico**, um ambiente caracterizado por um alto volume transacional e complexidade tributária. O Agente atua como um motor de regras e inteligência artificial que consome arquivos ZIP contendo XMLs, aplica regras de negócio (CFOP, NCM, etc.) para determinar as contas de Débito e Crédito do Plano de Contas, e utiliza um *Large Language Model* (LLM) do Gemini para realizar a auditoria final, identificar inconsistências (Regra Não Mapeada) e fornecer um resumo gerencial para o analista contábil.

Público Alvo (Quem irá utilizar a solução)

A solução **LançAI** foi concebida para otimizar o fluxo de trabalho de profissionais que lidam diretamente com o ciclo de vida dos documentos fiscais e o fechamento contábil. O público-alvo inclui:

Contadores e Analistas Contábeis: Usuários diretos que se beneficiarão da rotina de lançamentos otimizada, com eliminação da digitação manual e garantia de conformidade.

Analistas Fiscais: A consistência e a rastreabilidade dos dados de origem (XMLs) e o tratamento estruturado das regras fiscais (CFOP, NCM) beneficiam indiretamente a apuração e as obrigações fiscais.

Gerência Financeira e Diretoria: Terão acesso a dados contábeis validados e fechamentos mais ágeis, fundamentais para a tomada de decisões estratégicas e operacionais.

Auditores Internos e Externos: Aumentar a confiabilidade dos registros contábeis automatizados, simplificando significativamente os processos de *due diligence* e auditoria.

Justificativa do Tema Escolhido (Por que isto é importante, o que agrega de valor)

A automação de lançamentos contábeis no setor metalúrgico não é apenas uma melhoria

operacional, mas uma necessidade estratégica:

| Fator Crítico | Agregação de Valor pelo LançAI |
|------------------------------------|--|
| Volume e Complexidade | O Agente processa milhares de XMLs simultaneamente, gerenciando as complexas variações de CFOP (Código Fiscal de Operações e Prestações) e de Plano de Contas , algo inviável e lento no trabalho manual. |
| Redução de Erros e Inconsistências | Elimina erros de digitação e a aplicação subjetiva de regras. A inclusão da auditoria por LLM (Agente Gemini) garante que qualquer lançamento sem regra definida (Regra Não Mapeada) seja imediatamente sinalizado ao contador. |
| Otimização de Tempo e Custo | Libera analistas e contadores de tarefas operacionais, permitindo que se dediquem à análise de valor agregado, planejamento tributário e <i>compliance</i> ativo, crucial em um setor com alta carga tributária como o metalúrgico. |
| Agilidade no Fechamento Contábil | Antecipa o fechamento mensal e anual, fornecendo dados contábeis auditáveis de forma quase imediata, o que impacta diretamente a precisão dos relatórios gerenciais e custos de produção. |
| Conformidade Legal | O módulo de regras assegura que os lançamentos sigam as normas fiscais e contábeis vigentes, minimizando riscos de autuações e penalidades. |

Detalhamento do que foi desenvolvido (Estrutura e Operação da Solução)

O **LançAI** foi arquitetado como um **Sistema de Agente Modular Híbrido** em Python, utilizando o framework Streamlit para a interface e LangChain para a integração com a IA do Gemini. A solução é dividida em três componentes principais:

1. Módulo de Interface (UI/UX) - main.py

O arquivo main.py é responsável pela experiência do usuário e pelo gerenciamento do ciclo de vida do agente.

Gestão de Fluxo de Estado: Utiliza st.session_state e um callback avançado no st.file_uploader para garantir a execução sequencial correta (Upload → Processamento → Resumo Inicial → Interface de Query), evitando *loops* de re-execução.

Modos Híbridos: Suporta dois modos de operação:

LançAI Contábil: Ativado por arquivos ZIP contendo XMLs, gerando lançamentos e a interface do Agente de Query Fiscal.

Data Analysis: Ativado por CSV ou XLSX, permitindo a visualização de dados genéricos e consultas via Agente Gemini.

Identidade Visual: Aplica uma paleta cromática específica (PRIMARY_COLOR = #C05533 - Terracota Metálico) e um sistema de *feedback* visual (Sucesso, Alerta, Erro) para fornecer uma experiência robusta e consistente, alinhada à identidade visual da indústria.

2. Módulo de Dados e Regras - data_handler.py

Este arquivo é o **Motor de Regras** do sistema, manipulando a entrada de dados e aplicando a lógica contábil.

Parsing Robusto: Utiliza `xml.etree.ElementTree` para analisar arquivos XML e extrair metadados críticos como `NFe_Chave`, `Emissor`, `CFOP_Principal` e `Valor_Total`.

Mapeamento de Regras: O coração do módulo é o dicionário de regras (`MAPPING_RULES`) que define o mapeamento fixo de **CFOPs** para as respectivas **Contas de Débito/Crédito**. Qualquer CFOP sem regra definida é marcado com o valor `Regra Não Mapeada`.

Manuseio de Arquivos: Gerencia a descompactação de ZIPs, lidando com diferentes formatos (XML, CSV, XLSX) e implementando a limpeza de arquivos temporários (`TEMP_FOLDER`) com tratamento de exceções para garantir a estabilidade em diferentes ambientes operacionais.

3. Módulo de Inteligência (Agente) - agent_brain.py

O `agent_brain.py` é o **Cérebro do Agente LançAI**, utilizando o poder da IA do Gemini (via LangChain).

Configuração LLM: Empregamos o modelo **gemini-2.5-flash** para equilibrar custo/cota e desempenho, garantindo alta velocidade para o processamento de grandes volumes de dados.

Sistema de Prompt: Utiliza um `SYSTEM_PROMPT_LANCAI` especializado que orienta o Gemini a se comportar como um "Agente de Análise Contábil, especializado em Indústria Metalúrgica".

Funcionalidades Chave:

1. **Auditoria Inicial:** Na primeira execução, o Agente recebe o DataFrame completo em formato Markdown, e sua tarefa prioritária é gerar o `initial_summary`. Esta sumarização **DEVE** incluir a identificação e a contagem de lançamentos com a conta **Regra Não Mapeada**, fornecendo um ponto de partida crítico para a auditoria humana.
2. **Query Detalhada:** O Agente de Query permite aos usuários fazerem perguntas em linguagem natural sobre o DataFrame, transformando consultas complexas em análises rápidas (Ex: "Qual o total de lançamentos de CFOP 5102?").

Detalhamento da Operação da Solução

O usuário final opera a solução seguindo estes passos:

1. Faz o *upload* de um arquivo ZIP (contendo XMLs) ou CSV/XLSX na barra lateral.

2. O `data_handler.py` processa e aplica as regras.
3. O Agente (via `agent_brain.py`) é invocado na primeira re-execução (`st.rerun()`) para auditar os dados e gerar o resumo inicial.
4. O resumo é exibido (incluindo a auditoria de inconsistências), e a interface de Query é renderizada, permitindo a análise detalhada dos lançamentos gerados.
5. O usuário pode exportar os lançamentos prontos para integração no formato CSV.