# Relatório de Análise Gerencial: Agente de IA para Análise Fiscal

**Data:** 29 de outubro de 2025

## 1. Nome do Grupo

* [Nome do Grupo - Preencher]

## 2. Integrantes do Grupo

* [Integrante 1 - Preencher]
* [Integrante 2 - Preencher]
* [Integrante 3 - Preencher]
* [...etc]

## 3. Descrição do Tema Escolhido

O projeto desenvolvido é um **Agente de Análise Gerencial de Notas Fiscais**. Trata-se de uma aplicação web interativa que utiliza um agente de Inteligência Artificial (IA) para analisar dados de notas fiscais.

A solução permite que usuários façam perguntas em linguagem natural (português) sobre seus dados fiscais e recebam respostas diretas, cálculos complexos e visualizações de dados (gráficos) em tempo real. O sistema foi construído utilizando **Streamlit** para a interface do usuário (front-end) e **LangGraph** com o modelo **Google Gemini** para orquestrar a lógica do agente (back-end).

## 4. Público Alvo

O público-alvo desta solução é composto por profissionais que necessitam tomar decisões baseadas em dados financeiros e fiscais, mas que podem não possuir conhecimento técnico avançado em programação ou ciência de dados. Isso inclui:

* **Gestores e Diretores:** Que precisam de *insights* rápidos sobre faturamento, desempenho de produtos ou tendências de clientes.
* **Analistas Financeiros e de Negócios:** Que podem automatizar tarefas de geração de relatórios e análises de rotina.
* **Proprietários de Pequenas e Médias Empresas (PMEs):** Que geralmente não têm um departamento de BI (Business Intelligence) dedicado e precisam de ferramentas acessíveis para entender a saúde financeira do negócio.
* **Contadores:** Que podem usar a ferramenta para realizar auditorias rápidas ou análises para seus clientes.

## 5. Justificativa do Tema Escolhido

Em um mercado competitivo, a capacidade de extrair rapidamente inteligência de dados operacionais é um diferencial estratégico. No entanto, os dados fiscais (como Notas Fiscais Eletrônicas) são notoriamente complexos, volumosos e muitas vezes armazenados em formatos brutos (como arquivos CSV) com colunas inconsistentes.

**A importância deste projeto reside na democratização do acesso à análise de dados.**

O valor agregado para o público-alvo é:

* **Agilidade na Tomada de Decisão:** Em vez de esperar dias por um relatório manual, um gestor pode perguntar ao agente "Qual o faturamento total por mês?" e obter um gráfico em segundos.
* **Redução de Barreiras Técnicas:** O usuário não precisa saber SQL, Python ou Pandas. A interface de chat em linguagem natural elimina a complexidade técnica.
* **Inteligência de Dados Proativa:** O agente pode executar análises complexas que geram *insights* valiosos, como identificar os produtos mais vendidos, os principais clientes ou analisar tendências de gastos.
* **Eficiência Operacional:** Automatiza o processo de carregamento, limpeza e padronização de dados, tarefas que consomem muito tempo quando feitas manualmente.

## 6. Detalhamento do que foi Desenvolvido

A solução é composta por dois módulos principais: a interface do usuário (app.py) e o núcleo do agente (agent.py).

### Funções Principais

O agente possui um conjunto de "ferramentas" (tools) que o capacitam a executar tarefas específicas:

1. **Carregamento e Padronização de Dados (load\_fiscal\_notes):**
   * **Função:** Procura automaticamente por arquivos .csv em diretórios comuns (como nf/, nfe/).
   * **Operação:** Lê múltiplos arquivos, consolida-os em um único *DataFrame* (tabela) e, o mais importante, padroniza nomes de colunas inconsistentes (ex: "Valor da Nota", "vProd", "Total" são unificados para valor\_total). Também converte valores de texto para numéricos e datas.
2. **Análise de Dados com Python (python\_data\_analyst):**
   * **Função:** É o "cientista de dados" do agente. Recebe uma pergunta do usuário (ex: "Liste os 5 produtos mais vendidos") e a transforma em um script Python (usando Pandas e Matplotlib).
   * **Operação:** O LLM (Gemini) gera o código Python dinamicamente. Esse código é executado em *sandbox* sobre os dados já carregados.
   * **Geração de Gráficos:** Se a pergunta solicitar um gráfico (ex: "...gere um gráfico de barras"), o código Python gera a visualização, a salva como plot.png, e a interface a exibe no chat.
3. **Pesquisa Web (web\_search):**
   * **Função:** Permite ao agente buscar informações externas.
   * **Operação:** Utiliza a API Tavily para responder perguntas que não estão nos dados fiscais, como "Qual a cotação do dólar hoje?" ou "Qual a alíquota de ICMS para o produto X?".

### Como a Solução é Operada (Fluxo do Usuário)

1. **Configuração:** O usuário acessa a aplicação web (Streamlit) e insere suas chaves de API (Google e Tavily) na barra lateral.
2. **Carregamento de Dados:** O usuário pede: "Carregue os dados das notas fiscais". O agente executa a ferramenta load\_fiscal\_notes e confirma o carregamento.
3. **Interação (Chat):** O usuário faz perguntas em linguagem natural:
   * "Qual foi o faturamento total?"
   * "Qual o ticket médio por cliente?"
   * "Gere um gráfico de barras do faturamento por mês."
4. **Processamento do Agente (LangGraph):**
   * O agente recebe a pergunta.
   * O LLM (Gemini) decide qual ferramenta usar (ex: python\_data\_analyst).
   * O LLM gera o código Python necessário para responder àquela pergunta específica.
   * A ferramenta executa o código.
5. **Resposta:** O agente retorna a resposta em texto ou, se um gráfico foi solicitado, exibe o texto da análise e a imagem do gráfico (plot.png) diretamente na interface de chat.

## 7. Elementos Adicionais: Diagrama de Fluxo

O diagrama abaixo ilustra a arquitetura e o fluxo de operação da solução.

+------------------+  
| Usuário |  
| (Gestor/Analista)|  
+------------------+  
 |  
 | (Pergunta em Linguagem Natural)  
 V  
+------------------+  
| Interface Web |  
| (Streamlit) |  
| (app.py) |  
+------------------+  
 |  
 V  
+------------------+  
| Agente (LangGraph)|  
| (agent.py) |  
+------------------+  
 |  
 | (Orquestração)  
 V  
+------------------+  
| LLM (Gemini) |  
| (Entendimento e |  
| Geração de Cód.)|  
+------------------+  
 |  
 +----------------------------+-----------------------------+  
 | (Decide a Ferramenta) | |  
 V V V  
+-----------------+ +----------------------+ +----------------------+  
| Tool 1: | | Tool 2: | | Tool 3: |  
| load\_fiscal\_notes| | python\_data\_analyst | | web\_search (Tavily) |  
| (Lê CSVs) | | (Executa Pandas/Plot)| | (Busca Externa) |  
+-----------------+ +----------------------+ +----------------------+  
 | | (plot.png) |  
 | (Dados Carregados) | (Resultado/Gráfico) | (Info Web)  
 | V |  
 +--------------------->+------------------+<--------------------+  
 | Agente (Resposta) |  
 +------------------+  
 |  
 V  
 +------------------+  
 | Interface Web |  
 | (Exibe Resposta) |  
 +------------------+  
 |  
 V  
 +------------------+  
 | Usuário |  
 +------------------+