**Especificação de Software: MVP - Chatbot de Consulta à Lei do Simples Nacional**

Versão do Documento: 1.0

Data: 25 de setembro de 2025

**1. Visão Geral do Projeto**

O projeto consiste no desenvolvimento de um Mínimo Produto Viável (MVP) de um chatbot de IA especializado. O objetivo é criar uma ferramenta de consulta em linguagem natural que permita a profissionais de contabilidade fazer perguntas sobre a Lei Complementar nº 123/2006 (que institui o Simples Nacional) e receber respostas precisas, rápidas e baseadas exclusivamente nos documentos legais fornecidos, com citação direta da fonte.

**2. Objetivos do MVP**

* **Validar a Utilidade:** Testar se um chatbot de consulta legal agiliza o fluxo de trabalho de um escritório de contabilidade parceiro.
* **Testar a Tecnologia:** Avaliar a eficácia da abordagem de RAG (Retrieval-Augmented Generation) usando uma base de conhecimento controlada (PDFs).
* **Coletar Feedback:** Obter feedback qualitativo sobre a precisão, velocidade e usabilidade da ferramenta para guiar as próximas iterações.
* **Prototipagem Rápida:** Utilizar uma stack de low-code/no-code (Lovable + N8N) para acelerar o desenvolvimento e o ciclo de feedback.

**3. Público-Alvo**

* **Usuário Final:** Contadores, analistas fiscais e assistentes contábeis que trabalham diretamente com clientes optantes pelo Simples Nacional.
* **Características do Usuário:** Profissional com profundo conhecimento técnico, que valoriza a precisão acima de tudo, tem pouco tempo e precisa de ferramentas que aumentem sua produtividade.

**4. Requisitos Funcionais (RF)**

| ID | Requisito | Descrição Detalhada | Critérios de Aceitação |
| --- | --- | --- | --- |
| **RF-01** | **Interface de Chat** | O sistema deve prover uma interface de texto simples onde o usuário possa digitar e enviar suas perguntas. | 1. A interface deve apresentar uma caixa de texto para digitação. 2. Deve haver um botão de "Enviar" ou a funcionalidade de envio via tecla "Enter". 3. A conversa (perguntas e respostas) deve ser exibida em formato de chat. |
| **RF-02** | **Processamento da Consulta** | O sistema deve ser capaz de receber a pergunta do usuário em linguagem natural e processá-la para realizar uma busca na base de conhecimento. | 1. O texto inserido pelo usuário é capturado e enviado para o backend (N8N) via API. 2. O backend deve receber a pergunta sem perda de informação. |
| **RF-03** | **Geração de Resposta Baseada em Fonte** | O sistema deve gerar uma resposta baseada **exclusivamente** no conteúdo dos PDFs fornecidos (LC 123 e resumo PGFN). O sistema **não deve** usar conhecimento externo. | 1. A resposta deve ser contextualmente relevante à pergunta. 2. **Toda informação fornecida na resposta deve ser acompanhada por uma citação da fonte no formato ``, onde X é o número da fonte no documento original.** 3. Se a informação não for encontrada nos documentos, o sistema deve responder claramente: "Não encontrei informações sobre este tópico nos documentos fornecidos." 4. A resposta deve ser livre de "alucinações" ou informações não contidas nos documentos. |
| **RF-04** | **Exibição da Resposta** | O sistema deve exibir a resposta gerada de forma clara e legível na interface de chat. | 1. A resposta recebida do backend é exibida na interface de chat. 2. A formatação da resposta (parágrafos, quebras de linha, citações) deve ser preservada. |

**5. Requisitos Não-Funcionais (RNF)**

* **RNF-01 (Performance):** O tempo de resposta, desde o envio da pergunta até a exibição da resposta, não deve exceder 5 segundos em condições normais de rede.
* **RNF-02 (Usabilidade):** A interface deve ser minimalista e intuitiva, não exigindo treinamento prévio para uso.
* **RNF-03 (Segurança):** Toda a comunicação entre o frontend e o backend deve ser criptografada via HTTPS. Nenhum dado de conversa deve ser armazenado permanentemente no frontend.
* **RNF-04 (Confiabilidade):** O sistema deve ter uma disponibilidade de 99% durante o horário comercial (9h-18h, horário de Brasília).

**6. Arquitetura do Sistema**

O sistema seguirá uma arquitetura desacoplada, utilizando as seguintes ferramentas:

* **Frontend:** **Lovable**, responsável pela interface do usuário (UI) e pela captura da entrada do usuário.
* **Backend / Orquestração:** **N8N**, responsável por receber as requisições, processar a lógica de busca e orquestrar a chamada para o modelo de IA.
* **Base de Conhecimento:** Um **Banco de Dados Vetorial** (ex: Pinecone, Supabase pgvector) contendo os embeddings dos documentos PDF.
* **Modelo de IA:** Um serviço de LLM externo (ex: **API da OpenAI** ou **Google AI Platform**).

Fluxo de Dados:

Usuário no Lovable -> API POST para Webhook N8N -> N8N busca no Vetor DB -> N8N monta o prompt -> N8N chama a API do LLM -> N8N formata a resposta -> N8N retorna o JSON para o Lovable -> Lovable exibe a resposta.

**7. Detalhes da Implementação Técnica**

**7.1. Preparação da Base de Conhecimento (Pré-desenvolvimento)**

1. **Processamento dos PDFs:** Utilizar um script (preferencialmente Python) com bibliotecas como PyPDF2, LangChain e OpenAI para:
   * Extrair o texto de ambos os PDFs. **Atenção:** É crucial que o script consiga extrair as referências de fonte (ex: ``) e mantê-las associadas ao texto.
   * Segmentar o texto em "chunks" (pedaços) de tamanho semântico apropriado (ex: 1000 caracteres com 200 de sobreposição).
   * Gerar "embeddings" (vetores numéricos) para cada chunk de texto usando uma API de embedding (ex: text-embedding-ada-002 da OpenAI).
   * Armazenar os chunks de texto, os vetores e os metadados (como o número da fonte) no banco de dados vetorial escolhido.

**7.2. Configuração do Backend (N8N)**

1. **Nó 1: Webhook**
   * Método: POST.
   * Payload esperado do Lovable: { "pergunta": "texto da pergunta do usuário" }.
2. **Nó 2: HTTP Request (para o Vector DB)**
   * Primeiro, este nó deve chamar a API de embedding para vetorizar a pergunta recebida do webhook.
   * Em seguida, deve fazer uma requisição ao banco de dados vetorial, enviando o vetor da pergunta e solicitando os k chunks de texto mais similares (sugestão: k=5).
3. **Nó 3: Set / Function (Montagem do Prompt)**
   * Este nó irá construir o prompt final para o LLM. É um passo crítico.
   * **Template do Prompt:**
   * Você é um assistente especialista em legislação do Simples Nacional, baseado na Lei Complementar 123/2006. Sua tarefa é responder à pergunta do usuário de forma precisa, objetiva e baseada \*\*exclusivamente\*\* no contexto de trechos da lei fornecidos abaixo.
   * \*\*Regras Estritas:\*\*
   * 1. Use APENAS a informação presente no "Contexto Fornecido". Não use nenhum conhecimento prévio.
   * 2. Sua resposta deve ser direta e concisa.
   * 3. Ao usar uma informação do contexto, você DEVE citar a fonte imediatamente após a frase ou trecho, usando o formato.
   * 4. Se a resposta não puder ser encontrada no contexto, responda exatamente com a frase: "Não encontrei informações sobre este tópico nos documentos fornecidos."
   * \*\*Contexto Fornecido:\*\*
   * {{ $json.resultado\_do\_no\_2 }}
   * \*\*Pergunta do Usuário:\*\*
   * {{ $json.pergunta\_do\_webhook }}
   * \*\*Resposta:\*\*
4. **Nó 4: Chamada ao LLM (OpenAI / Google AI)**
   * Conectar à API do provedor de IA escolhido.
   * Modelo: gpt-4-turbo ou gemini-1.5-pro (modelos com boa capacidade de seguir instruções).
   * Parâmetros: temperature = 0.2 (para respostas mais factuais e menos criativas).
   * Input: O prompt gerado no Nó 3.
5. **Nó 5: Resposta do Webhook**
   * O próprio Nó 1 (Webhook) será usado para enviar a resposta de volta.
   * Corpo da Resposta: { "resposta": "{{ $node['Nó\_LLM'].json.choices[0].message.content }}" }. (A sintaxe pode variar).

**7.3. Configuração do Frontend (Lovable)**

* Configurar a chamada de API para a URL do Webhook do N8N, passando o JSON esperado.
* Mapear o campo resposta do JSON de retorno para ser exibido na área de chat.

**8. Critérios de Aceitação do MVP**

O MVP será considerado pronto para testes quando os seguintes critérios forem atendidos:

* [ ] O usuário consegue enviar uma pergunta pela interface do Lovable.
* [ ] O fluxo no N8N é executado com sucesso após o envio da pergunta.
* [ ] Uma pergunta sobre o Art. 17 da LC 123 retorna uma resposta contendo informações daquele artigo.
* [ ] A resposta contém as citações `` corretamente.
* [ ] Uma pergunta sobre um tópico não relacionado à lei (ex: "Qual a capital da França?") retorna a mensagem padrão de "informação não encontrada".
* [ ] O tempo de resposta médio está abaixo de 5 segundos.