Jr AI Engineer

Na Charla, um dos desafios constantes é a automação de processos que envolvem documentos não estruturados. Frequentemente, recebemos documentos em PDF e precisamos extrair informações importantes para alimentar nossos fluxos agênticos.

Sua **missão** é nos ajudar a dar o primeiro passo na automação desse processo.

O Desafio

Você deve criar um script em Python que utiliza a biblioteca **PydanticAI** para criar um agente de IA que extraia informações específicas de uma Nota Fiscal de Serviço (PDF) que forneceremos. O objetivo é ler o conteúdo do PDF, usar um LLM para "entender" o texto e retornar os dados de forma **estruturada** e **validada**.

Recursos Fornecidos

1. Um arquivo PDF de exemplo:
2. Este documento com as instruções.
3. Chave de acesso ao VertexAI (serviço de IA da Google Cloud Platform).

A Tarefa: Passo a Passo

1. Configuração do ambiente:
    a. Crie um ambiente virtual em Python;
    b. Instale as bibliotecas necessárias. No mínimo, você necessitará de:
        i. pydantic-ai
2. Leitura do PDF:
    a. Há duas abordagens possíveis:
        i. Leitura do conteúdo do PDF com bibliotecas como pypdf ou pdfplumber;
        ii. *Parseamento* direto do PDF pelo LLM. Dica carinhosa do AI Lead: considere **fortemente** utilizar o seguinte recurso https://ai.pydantic.dev/input/#document-input, especificamente, gosto bastante da abordagem sugerida na seção que fala do uso de **BinaryContent**.

        Evidentemente, a escolha é sua. Os dois caminhos possíveis. Um é bem mais fácil do que o outro. Choose wisely 🧙
3. Modelagem dos Dados com Pydantic:
    a. Dica carinhosa 2: Pydantic e PydanticAI são bibliotecas diferentes.
    b. Defina um modelo Pydantic que represente a estrutura da informação que você deseja extrair da nota fiscal. Queremos os seguintes campos:
        i. Descrição do Serviço
        ii. Valor do Serviço
        iii. Número da Nota
        iv. Data de emissão:
        v. Valor Total
        vi. CNPJ do Prestador
    c. Sim, sabemos que Pydantic é chatinho, então segue um exemplo de código pra te ajudar nessa parte:

```python
from pydantic import BaseModel, Field

class ExtracaoOutput(BaseModel):
        descricao: str = Field(description="Descrição detalhada do serviço prestado.")
        valor: float = Field(description="Valor do item de serviço.")
        numero_nota: str = Field(description="O número da nota fiscal")
        # Insira o restante dos campos necessários.
```

4. Extração com pydanticAI:
    a. Você recebeu credenciais do VertexAI. Essas credenciais são acessíveis através de uma service account. Para configurar o provedor, veja essa seção da documentação: https://ai.pydantic.dev/models/google/#service-account
    b. Instancie o LLM 'gemini-2.5-flash'.
    c. Crie um agente de IA que execute a extração. https://ai.pydantic.dev/agents/
5. Output:
    a. Ao final, seu script deverá imprimir o objeto ExtracaoOutput preenchido em formato JSON no console (ou Jupyter Notebook, você que manda).

O que vamos avaliar?

- Qualidade do Código: organização, clareza, uso de boas-práticas e comentários quando necessário.
- Modelagem Pydantic: a estrutura do seu modelo de dados está correta e bem definida?
- Funcionalidade: O script roda sem erros e a extração dos dados é precisa?
- Documentação: um README.md simples explicando como configurar e rodar seu projeto. SIMPLES, não precisa escrever uma bíblia 😀

Como Entregar

1. Crie um repositório público no GitHub;
2. Suba todo o seu código, um requirements.txt com as dependências e o README. Envie o link do repositório para bert@charla.chat até 15/10/2025.

Uma nota final para você

Estamos mais interessados em ver como você pensa e aborda o problema do que em uma solução 100% perfeita (aliás, será que isso existe?). Não hesite em documentar no README as dificuldades que encontrou e as decisões que tomou.

Boa sorte!