# Documentação de Arquitetura e Escopo

**Projeto:** Sistema Modular de Análise de Dados da Câmara dos Deputados **Fonte de Dados:** API de Dados Abertos da Câmara dos Deputados **Linguagem:** Python 3.10+ **Nível de Complexidade:** Intermédio (6 meses de experiência)

## 1. Visão Geral

O projeto consiste no desenvolvimento de um ecossistema de scripts e aplicações em Python desenhado para consumir, processar e visualizar dados públicos do poder legislativo brasileiro. Para garantir a manutenibilidade e escalabilidade, o sistema será dividido em cinco módulos independentes.

## 2. Configuração do Ambiente

Sendo este um projeto pessoal/freelance, a gestão de dependências e do ambiente virtual será feita utilizando **Poetry** (recomendado para gestão moderna de pacotes) ou **Conda** (recomendado caso o foco se vire fortemente para Data Science pesada).

**Comandos de Inicialização (Exemplo com Poetry):**

```
poetry init
poetry add requests pandas networkx matplotlib python-telegram-bot streamlit plot
```

## 3. Estrutura de Diretórios

A arquitetura do projeto seguirá o padrão de monorepo com pacotes separados por domínio de negócio:

```
projeto_camara_api/
├── pyproject.toml          # Configurações do Poetry (ou environment.yml para Cc
├── README.md               # Visão geral do repositório
└── modules/                # Módulos independentes
    ├── tracker_gastos/
    ├── network_analyst/
    ├── legis_notifier/
    ├── parlamentar_dashboard/
    └── tema_miner/
```

## 4. Especificação dos Módulos

### 4.1. Módulo 1: `tracker_gastos`

- **Objetivo:** Automação, extração e consolidação de dados financeiros (CEAP).

- **Bibliotecas Core:** `requests`, `pandas`.
- **Fluxo de Trabalho:**
  1. Consumir o endpoint `/deputados/{id}/despesas`.
  2. Implementar lógica de paginação (loop `while`) para extrair todos os registos.
  3. Limpar dados e tratar valores nulos utilizando o Pandas.
  4. Exportar relatórios agregados (ex: somatório mensal por categoria) para `.csv` ou `.parquet`.

#### 4.2. Módulo 2: `network_analyst`

- **Objetivo:** Mapeamento de redes de influência e ligações políticas.
- **Bibliotecas Core:** `networkx`, `matplotlib`.
- **Fluxo de Trabalho:**
  1. Consumir os endpoints `/frentes` e `/deputados`.
  2. Processar dicionários aninhados para cruzar deputados com as respetivas frentes parlamentares.
  3. Gerar uma matriz de adjacência (Grafos).
  4. Visualizar as "pontes" (nós de ligação) entre diferentes grupos de interesse.

#### 4.3. Módulo 3: `legis_notifier`

- **Objetivo:** Automação de alertas e integração com serviços de mensagens.
- **Bibliotecas Core:** `python-telegram-bot` (ou `requests` para webhooks).
- **Fluxo de Trabalho:**
  1. Monitorizar o endpoint `/proposicoes` focado em temas ou palavras-chave específicas.
  2. Implementar persistência local (ex: num ficheiro `last_id.json`) para registar a última proposição lida e evitar notificações duplicadas.
  3. Disparar uma notificação via Telegram ou X (Twitter) sempre que uma nova atualização for detetada.

#### 4.4. Módulo 4: `parlamentar_dashboard`

- **Objetivo:** Visualização interativa de dados e Front-end.
- **Bibliotecas Core:** `streamlit`, `plotly`.
- **Fluxo de Trabalho:**
  1. Criar uma interface web de utilizador para seleção de filtros (ex: Estado/UF, Partido).
  2. Apresentar rankings interativos de presenças e votações.
  3. Implementar sistema de cache (`@st.cache_data`) para minimizar as requisições repetitivas à API e melhorar o tempo de resposta da aplicação.

### 4.5. Módulo 5: `tema_miner`

- **Objetivo:** Processamento de Linguagem Natural (NLP) aplicado a ementas legislativas.
- **Bibliotecas Core:** `wordcloud`, `nltk` (ou `spacy`), `re`.
- **Fluxo de Trabalho:**
  1. Extrair os textos das ementas dos Projetos de Lei num dado período.
  2. Aplicar expressões regulares ( `re` ) para limpeza do texto.
  3. Remover *stopwords* (palavras de ligação sem valor semântico isolado).
  4. Gerar uma *WordCloud* ou análise de frequência de termos para identificar as pautas mais debatidas.

*Documento gerado para orientação de desenvolvimento iterativo.*