TCC - Cidadão.AI

→ Evolução do Transparência BR - Analytics.
→ API do portal da transparência para coleta de dados.
→ Projeto anterior já tinha ML.

Cidadão.AI vai ser um sistema com chatbot com IA (PLN - processamento de linguagem natural para responder perguntas ao cidadão sobre informações a qual ela tem acesso.

- Sistema com agente de IA. (múltiplos agentes)
- Interface user-friendly.
- Engenharia de Requisitos (continuar a partir da base do analytics.)
- Arquitetura do sistema com múltiplos agentes (modelos do huggingface!

Iniciar a documentação base. criar repo e iniciar desenvolvimento da idéia na IA Claudecode.

Notas do TCC.

- Implementação inicial V. 1.0.
  - Definição da arquitetura do sistema
  - API de dados.
  - Modelos de IA para agentes.
  - Github & Hugging face p/ deploy.
  - Revisão da Bibliografia.
  - Tem de estar de acordo com as normas ABNT e padrão dos documentos do IF.
  - Contato c/ Aracle p/ updates.

- Próximos passos:
  - Revisão da Bibliografia.
  - Estudar conceitos matemáticos dos algoritmos: Entropia de Shannon.

- Entropia de Shannon:

- Teoria da informação

- Medida de incerteza no dataset

quanto maior a entropia mais imprevisível a informação.

- Conceito matemático:

É a soma de todos os eventos possíveis multiplicada pelo $\log_2$ da probabilidades dos eventos, calculando matematicamente a incerteza.

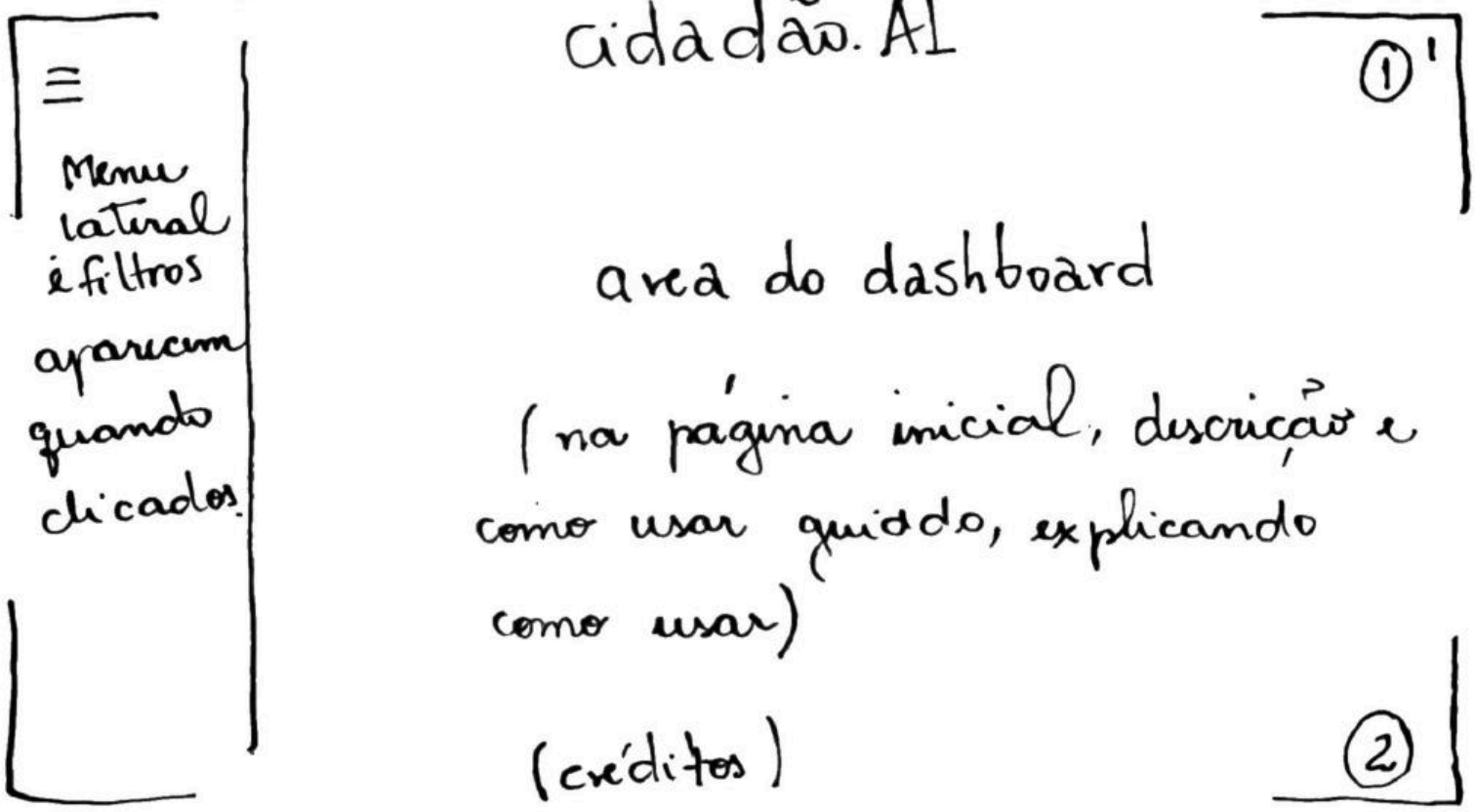
Mock-Up - TCC - Cidadão.IA

① Landing Page

① Aqui ficarão os botões de idioma / tema claro ou escuro.

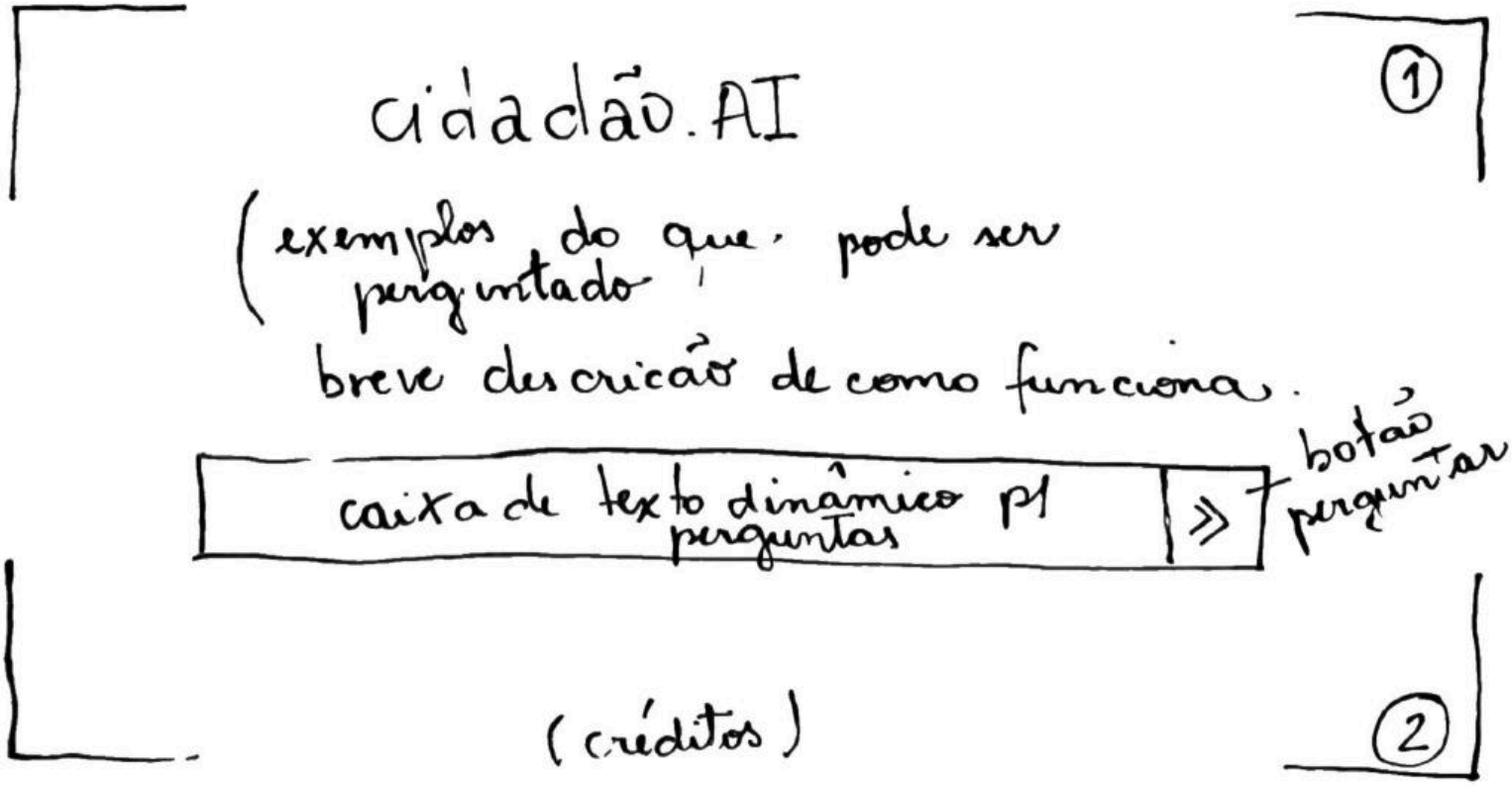
② Aqui ficarão os créditos de desenvolvimento (nome do autor, IFSul de Minas, etc). acesso ao repositório. link p/ a documentação técnica • detalhes das APIS.

(2) Página consulta avançada:



OBS.: Botões 1 & 2 idênticos ao Landing-Page.

(3) Pergunte ao Modelo:



TCC - Pesquisa do dia

Teoria dos jogos para Coordenação multi agente.

- O que é a teoria dos jogos?

Estuda como agentes racionais tomam decisões interdependentes. Cada agente age considerando o que os outros farão. Cada agente tem objetivos, percepções e capacidades distintas, mas deve coordenar ou competir com os demais.

Seja $T = \{A, E, C, D, \Omega\}$

$$T = A \cdot E \cdot C \cdot D \rightarrow \Omega$$

ou seja: o sistema T recebe um agente especializado $a_i \in A$, uma entidade governamental $e_j \in E$, um contrato público $c_k \in K$ e um documento $d_l \in D$ e mapeia esse quadruplo para uma possível anomalia. $\omega \in \Omega$.

jogo Cooperativo: $\Gamma = (N, v)$ onde:

- N é o conjunto de agentes
- $v(S)$ = função característica que define valor coligação

Teorema 1: Estabilidade de Coalizões:

Para uma coalizão $S \subseteq N$ ser estavel no sistema, deve satisfazer:

$$\sum_{i \in S} \phi_i(v) \geq v(S) \quad e \quad \sum \phi_i(v) = v(N)$$

onde ϕ i' o valor de Shapley[1] do agente i, garantindo eficiência e estabilidade.

- Shapley, LS. .= (1953). - Wooldridge, M. 2009

- Lopes, R. A. C (2008)

1. Valor de Shapley, em teoria dos jogos mede a contribuição média de cada participante, ou (variável) para um resultado considerando todas as combinações possiveis.

Pesquisa do Dia

Modelo Probabilísticos de Anomalias:

- definição: $\alpha = D \rightarrow [0,1]$, uma função de pontuação de anomalia onde:

$$\alpha(d) = 1 - \max_{i=1}^{K} P(d \in C_i | \theta_i)$$

onde C_i, são classes normais de documentos e θ_i, são os parâmetros do modelo para cada classe.

Esse modelo é ensemble de algoritmos especializados; em vez de "confiar" em um único detector, ele utiliza vários e combina sua saída para obter uma pontuação final robusta.

Filosoficamente, rs; esse modelo atua como uma "assembléia" estatística. Cada algoritmo emite um parecer sobre o documento - e o sistema os escuta antes de declarar: "há algo suspeito aqui". A força do método está na

<u>diversidade</u>

dos modelos e da robustez da votação.

Análise de complexidade:

- Complexidade Temporal

$O_{(n.m.K)}$ onde $n = |D|$, $m = |A|$, $K =$ complexidade média dos algoritmos

- Complexidade Espacial

$O_{(n.m)}$ para armazenar pontuações intermediárias

- Complexidade de Comunicação

$O_{(m.\log(n))}$ para sincronização entre agentes.

Referências:

Barros, R.C. & Basgalupp, M.P. (2019)
Souza, J.T. & de Mello, R.F. (2012)
Chandola, V., Banerjee, A, Kumar, V (2009)

TCC- Vapinik - Chervonenenkis

A dimensão de Vapinik- Chervonenkis (VC-dimensão) é uma medida de uma capacidade de um modelo (ou uma classe de funções) de classificar corretamente diferentes arranjos de dados.

Mede a capacidade de generalizacão: quanto mais complexa a classe de hipóteses (maior a VC-dim), maior o risco de overfitting. Baixa demais o modelo não aprende.

É usada para 1. (estimar generalização) 2. (comparar modelos. 3. Controlar a complexidade.

Autores fundaram a base do aprendizado estatístico, depois trouxeram o conceito de "máquina de vetores de suporte". A SVM, por sinal, busca um limite ótimo de generalização, (ideia de margem) que se relaciona a VC-dimensão

"A VC dimensão é o número de dimensões que a ignorancia pode vestir antes de ser descoberta Modelo pode saber de tudo, mas se ele consegue fingir que sabe em todos os arranjos possíveis, ele <u>não aprendeu</u>; ele <u>decorou</u>!

TCC: Teoria do aprendizado Estatístico:

- Bound PAC para detecção de anomalias

Com probabilidade pelo menos $1-\delta$, o erro de generalização do ensemble é limitado por:

$$R(h) \leq \hat{R}(h) + \sqrt{\frac{d \log(2m-d) + \log(4/\delta)}{2m}}$$

onde $R(h)$ é o erro verdadeiro, $\hat{R}(h)$ é o erro empírico, d é a dimensão VC, e m o tamanho da amostra.

A prova segue da teoria de Vapnik-Chervonenkis aplicada ao caso específico de detecção de anomalias em dados governamentais. Considerando a natureza estruturada dos documentos públicos, a dimensão VC pode ser limitada superiormente por $\log(|V|)$ onde V é o vocabulário especializado.

Izbicki, Rafael (2024)
Vapnik, Vladmir (1998)

TCC.

Teorema de Convergência para Sistema Multi-Agente

Garante que, sobre certas condições de suavidade (condição de Lipschitz[1]) nas funções de comunicação entre agentes), o sistema inteiro converge para um estado de equilíbrio ε-aproximado; isto é, se estabiliza próximo a uma solução ótima ao longo do tempo.

Equação fundamental:

$$\lim_{t \to \infty} \| s(t) - s^* \| \leq \varepsilon$$

indica que a distância entre o estado do sistema no tempo t $(s(t))$ e o estado ótimo s^*, torna-se arbitrariamente para t suficientemente grande. Isso significa que o sistema é estável e convergente, mesmo os agentes trabalhando de forma descentralizada. A convergência coletiva assegura que o comportamento coletivo não derive para o caos do tempo - comum em arquiteturas multi-agentes sem controle teórico.

Izbicki, Rafael (2024)

TCC · Cidadão.AI

Na área que meu projeto está sendo desenvolvido (Transparência pública) já temos alguns projetos similares e/ou com propósitos compatíveis.

São eles:

· EuroAI: Machine learning for European Public Procurement Analysis (Müller, Dubois, Rossi, 2023)

· TransparencIA: Deep learning para auditoria governamental Brasileira (Silva, Santos, Oliveira)

· Cooperative multi-agent systems for distributed government data analysis. (Kim, Zhang, Anderson '23

· Opengov: A platform for automated government transparency (Chen, Rodriguez, Johnson, 2022)

· Esemble methods for financial fraud detection in Public Sector (Brown, Williams, Taylor, 2023)

Vou pegar o que cada um tem de melhor! Vou fazer análise de todos usando IA.

Cidadão.AI

Lang Chain:

Lang Chain é um framework open-source para a construção de aplicações com LLMs (Large Language Models) que combinam linguagem natural com lógica de programação. Ele permite que os modelos interajam com fontes externas de dados, memória, API, ferramentas, banco de dados - tornando os agentes mais "inteligentes" e úteis em CONTEXTOS REAIS!

Componentes - Chave:

1. Prompt Templates: moldam as perguntas que serão enviadas ao modelo. Ex.: f strings inteligentes com variáveis dinâmicas.

2. LLMs: A interface com modelos como GPT, Mistral Claude, Groq. Pode ser local ou API externa.

3. Chains: Encadeiam múltiplas etapas (prompts + LLMs + lógica) em um só fluxo Ex.: consulta → busca → resumo

4. Agents: Entidades que tomam decisões sobre quais ferramentas usar. São o "cérebro pensante" da pipeline

5.// Tools: Funções que o agente pode usar, como buscar na internet, calcular, consultar um banco de dados, etc.

6.// Memory: Armazena conversa passadas contextos ou estados, essencial para diálogos contínuos

7.// Retrievers + Vector Stores: Usados em RAG[1] (retrieval-argumented generation. Indexa documentos e busca vetorialmente com embeddings[2].

---

1. RAG: (Retrieval-augumented generation) Técnica que combina recuperação de informações com geração de texto Em vez de confiar só na "memória" do modelo, ele busca dados relevantes em uma base vetorial (FAISS ou Chroma) e usa essas evidências para gerar respostas.

2. Embeddings: Representações numéricas vetoriais de palavras. onde a semântica é preservada. Permitem medir similaridade entre textos para buscas inteligentes e classificação de contexto.

Lewis, Patrick et al: (v. 33, p 9459 - 9474), 2020
Mikolov, Tomas Efficient Estimation of Word representation. 2013
Souza, Nogueira, Lotufo: BERTimbau (2020 p. 403 -> 418)

TCC. F1-Score - Cidadão.AI

F1-Score é a métrica que combina precisão e recall em um único valor harmônico, refletindo o equilíbrio entre exatidão e completude de um classificador.

Calculo:

- Precisão (P): fração de previsões positivas corretas

$$P = \frac{TP}{TP + FP}$$

- Recall (R): fração de casos positivos corretamente identificados:

$$R = \frac{TP}{TP + FN}$$

- F1-Score: média harmônica entre P & R.

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}$$

TP: true positives (positivos corretamente previstos)
FP: false positives (falsos alarmes)
FN: false negatives (casos perdidos)

Powers, David M.W: Evaluation: From precision, recall and F-Measures. Technical Report SIE-07-001.

TCC. Cidadão.AI

FAISS - Facebook AI Similarity Search

Faiss é uma biblioteca de código aberto, criada pelo Facebook, especializada em busca eficiente por similaridade entre vetores em espaço de alta dimensão. Ou seja, é uma ferramenta para encontrar os vetores mais parecidos com um dado vetor de consulta, mesmo quando lidamos com milhões (ou bilhões) de vetores. É usado principalmente em sistemas de recomendação, RAG (retrieval-augumented generation), pesquisa semantica com embeddings, e deduplicação de dados vetoriais. Quando você transforma texto, imagem ou áudio em vetores, o FAISS entra em cena para indexar e consultar esses vetores de forma rápida e escalável.

Conceitos-chave:

1. Índice vetorial: estrutura interna que FAISS cria para acelerar a busca.
2. Approximate Nearest Neighbors (ANN): FAISS sacrifica um pouco de exatidão para ganhar muita

velocidade: útil quando a precisão absoluta não é necessária.

3/. Suporte a GPU: FAISS roda tanto em CPU quanto em GPU, sendo extremamente rápido com CUDA. (GPU: grafic processing unit) (CUDA: Compute Unified Device Architecture).

É um motor de busca semantico, vetorizado feito para rodar rápido.

Estruturas Matemáticas:

1/. Index flat (brute-force)

- Busca exata, complexidade $O(n \cdot d)$

$$\|q - x_i\|^2 = \|q\|^2 + \|x_i\|^2 - 2q \cdot x_i$$

2/. Index IVF (Inverted file index)

- Usa K-means para particionar X em c clusters com centróide μ_j.
- No momento da busca, só examina clusters mais próximos de q.
- Reduz o tempo de $O(n)$ para $O(n/c)$

3/1 · PQ : Product Quantization :

· Divide o vetor $x \in \mathbb{R}$ em m subvetores.

· Cada subvetor é quantizado usando K-means

· A distância aproximada é computada a partir de uma tabela pré-calculada de distâncias

FAISS aplica:

$$\min_{x_i \in X} \|q - x_i\|^2 \approx \min_{\text{cod.}} \hat{d}(q, x_i)$$

onde $\hat{d}$ é uma distância aproximada baseada em quantização, como no PQ.

garante:

· mais precisão ⇔ mais custo computacional

· exatidão vs velocidade controlados via parâmetros: nº de clusters, nº de probes, tipo índice

Johnson, Douze, Jégou. Billion-scale similarity search with GPUs – 2021.

Jégou, Douze, Schmid. Product quantization for nearest neighbor search. p. 117-128 – 2011.

Neves. Aprendizado de máquina com python – 2020 guia prático.

JWT Auth: TCC - Cidadão.AI

JWT: Jason Web Token, é um token codificado em base 64[1], dividido em três partes):

1ii Header: contêm o algoritmo de assinatura e o tipo de token (JWT)

2ii Playload: contém as informações (claims) do usuário, como id, email, exp (expiração)

3ii Signature: é a parte criptografada que mantém e garante a autenticidade do token. (normalmente com HS256 ou RS256[2].

Como funciona a autenticação JWT?

1ii Login: usuário envia informações.

2ii Geração do Token: o servidor verifica as credenciais e retorna um JWT assinado.

3ii Armazenamento: o cliente guarda esse token em um local seguro (cookie, localStorage, sessionStorage)

4ii Requisições futuras: em cada requisição, o cliente envia o token no cabeçalho

5ii Validação: o servidor decodifica o token, verifica a assinatura e, se válido, autoriza a ação.

- Vantagens:
  - Stateless: não armazena sessão no servidor.
  - Escalável: ideal para sistemas distribuídos e microserviço
  - Portável: pode ser usado entre domínios e com múltiplas plataformas (Web, mobile, etc.)

FERREIRA, João. JWT: Autenticação moderna em APIs REST. 2021

notas de rodapé

1: Base64: codificação binária → texto legível. usada para transportar dados como strings seguras.

2: HS256: algoritmo de assinatura HMAC (Hash-based message authentication code). Rápido e Simétrico (mesma chave para assinar e verificar).

RS256: algoritmo de assinatura com RSA + SHA-256. Assimétrico: usa chave privada para assinar e chave pública para verificar.

KAUFMAN, Charlie. Network security: Private Communication in a Public person, 2010.

Cidadão.AI - Chroma DB

Chroma DB é um banco de dados vetorial especializado no armazenamento, indexação e busca de embeddings, usado principalmente com LLMs e sistemas RAG.

- armazenamento de vetores (embeddings)
- busca por similaridade semântica
- distâncias: Cosseno, Euclidiana, Manhattan
- integração com LangChain, Hugging face ...
- suporte a filtragem por metadados.
- indexação local e persistente
- atualização e exclusão dinâmica de documentos
- suporte a coleção nomeadas.

Bases Matemáticas:

- Vetores em $\mathbb{R}^N$:
  $\vec{v} = [v_1, v_2, \dots v_n]$
- Distância Euclidiana:
  $d(\vec{u}, \vec{v}) = \sqrt{\sum_{i=1}^{n} (u_i - v_i)^2}$
- Similaridade cosseno.
  $\cos(\theta) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \cdot \|\vec{v}\|}$
- Busca KNN
  retorna K-vetores próximos

→ Boyd, Stephen. AI-Powered Search. O'Reilly, 2023
→ Oliveira, Leandro. Sistemas de Recuperação de infos com IA, 2022

TCC : 8 Agentes de IA Especializados

1. Master Agent : orquestração com auto-reflexão

- Função: coordenador central do sistema multi-agente
- Capacidades:
  - plan_investigation: Planeja estratégias de investigação.
  - coordinate_agents: coordena outros agentes?
  - monitor_progress: monitora processo das investigações
  - reflect-on-results: auto reflexão sobre qualidade.
  - generate_explanations: gera explicações detalhadas
  - adapt_strategies: adapta estratégias baseado em resultados?
- Características Técnicas?
  - sistema de reflexão com threshold de qualidade (0.8
  - loops de reflexões maximos: 3 iterações
  - registro de agentes subordinados? (register_agent)
  - planeja investigações compleras com multiplos passos?
  - score de confiança calculado automaticamente

2. Investigator Agent: especialista em detectar irregularidades em dados públicos

- Capacidades

  - detect_anomalies: detecção de multiplos tipos de anomalias.

- Algoritmos especializados:

  - Price anomaly: estastística (>2,5 desvio padrão)
  - Vendor concentration: >70% p/ único fornecedor.
  - Temporal patterns: análise de freqüência.
  - Duplicate contracts: similaridade >85% entre contratos
  - Payment Patterns: discrepância entre valores.

3. Analyst Agent

- Função: Análise profunda de padrões financeiros e correlações.

- Capacidades

  - analyse_patterns: Análise de padrões complexos.

- Tipos de análise:

  - Spending Trends: tendências temporais de gastos
  - organizational patterns: comparação entre órgãos
  - Vendor behavior: comportamento de fornecedores.
  - seasonal patterns: sazonalidade.
  - Value distribution: distribuição por faixas de valor.

- correlation analysis: correlações estatísticas.
- efficiency metrics: métricas de eficiência organizacional.

Análises Estatísticas
- correlacão mínima: 0,3
- janela de tendência: 6 meses
- Z-score para outliers
- Regressão linear p/ tendências

Faixas de Valor
- micro: R$ 0 - 8 K (dispensas)
- small: R$ 8 - 176K (convites)
- medium: R$ 176K - 1,5M (tomada de preço)
- large: > R$ 1,5M (concorrências

4. Report Agent:
   - Função: gera relatórios profissionais em vários formatos
   - Capacidade:
     - generate_report: geração de relatórios
   - Tipos de Relatório:
     - Investigation Report: relatório investigação completo.
     - Analysis Report: análise de padrões.
     - Combined Report: Investigação + análise consolidada
     - Executive Summary: resumo executivo.
     - Anomaly Summary: foco em anomalias.
     - Trend Analysis: análise de tendências.

Formatos de Relatórios	Adaptação p/ audiência
- Markdown - HTML - JSON	- Linguagem adaptada automaticamente - Technical, Executive, Public.

5. Memory Agent:

- Função: sistema de memória inteligente multi-dimensional.
- Capacidades:
  - store_episodic: mem. de investigações específicas?
  - retrieve_episodic: recuperação de contexto histórico
  - store_semantic: conhecimento geral sobre padrões
  - store_conversation: contexto conversacional
  - get_relevant_contex: busca semântica por relevância
- Tipos de Memória:
  - Episódica: investigações específicas (1.000 máx. 30 dias)
  - Semântica: conhecimento sobre padrões (60 dias)
  - Conversacional: contexto de dialogo (50 turnos, 24h)
- Tecnologias:
  - Redis - Chroma DB - Auto-decay?

6. Semantic Router:

- Função: roteador que dirige consultas para agentes apropriados.
- Capacidades:
  - route_query: roteamento principal
  - detect_intent: detecção de intenção
  - analyse_query_type: análise de tipo/complexidade
  - suggest_agents: sugestão de agentes alternativos

7. Observer Agent

8. Validator Agent.

Nota: outros agentes podem ser pensados e inseridos no sistema, como já está sendo planejado a implementação da Transformada de Fourier para análises temporais mais complexas.

P.S.: Quanto mais pesquiso, mais vejo pontos de melhorias, com técnicas e algoritmos mais avançados!

Agentes 7 e 8 ainda não implementados
Considerar:
BERTimbau pra PLN em PT-BR!

Plano de Ação - Cidadão.AI.

Atual: github + hugging face

- github → repositório completo
  - documentação com github pages
  - hugging face p/ deploy
- hugging face → Modelos de ML
  - transformers
  - front com gradio

© que vai mudar?

1) repositório atual: renomear p/ cidadão-ai-backend onde terá só o backend, agentes e algoritmos de consulta.

2) Novo repositório: cidadão-ai-frontend.
vai ter somente o frontend, e a documentação técnica e institucional do projeto. "frontend será em next.js"

Como vai ficar:

1. github (backend) → hugging face (backend).
2. github (frontend) → vercal (front)
   ↓
   conectado ao hugging face.