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.

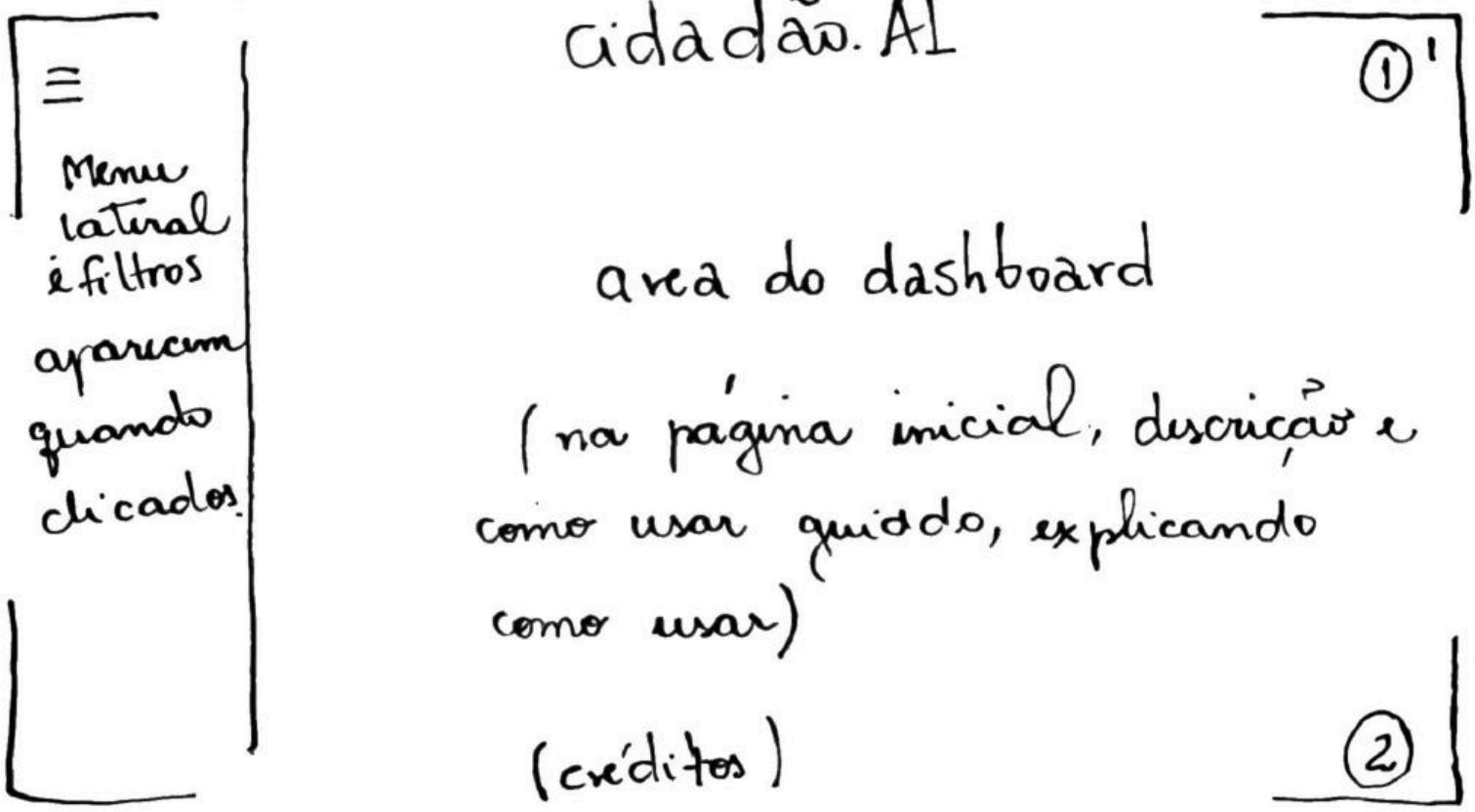
Mock-Up - TCC - Cidadão.IA

① Landing Page

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

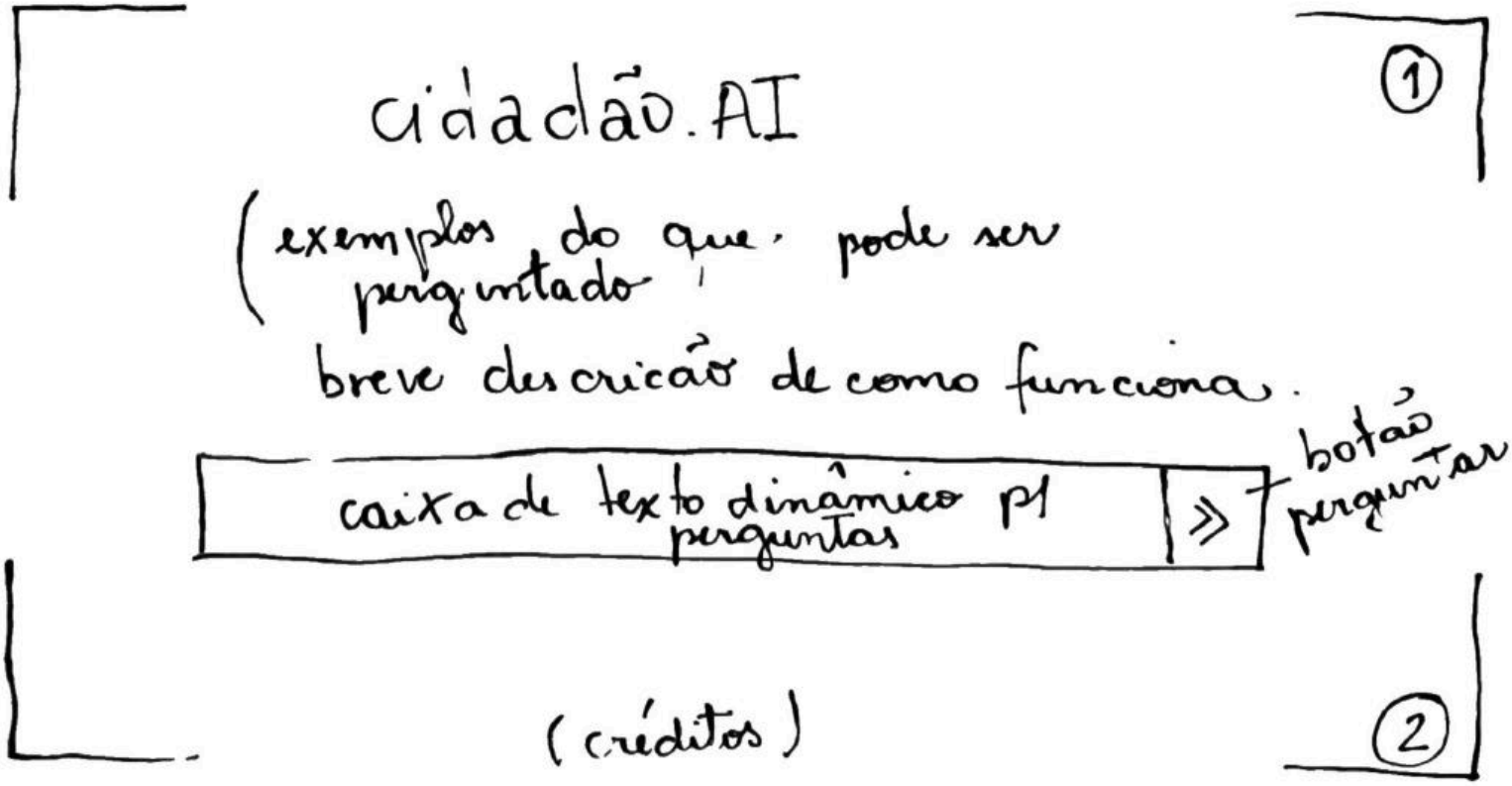
② Aqui ficarão os créditos de desenvolvimento (nome do autor, IFSul de Minas, etc). acesso ao repositório. link p/ a documentação técnica • detalhes das APIS.

(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.

TCC - Cidadão.AI = XAI. Explainable AI

Explainable AI é um campo da inteligência artificial que busca tornar os sistemas de IA compreensíveis para seres humanos. Em vez de ter uma "caixa preta" onde os modelos tomam decisões sem que podemos entender o porquê, o XAI quer abrir essa caixa e mostrar COMO E PORQUÊ a IA chegou àquela conclusão.

Modelo opaco (exemplo): Um modelo de deep-learning diz "Esse empréstimo deve ser negado", mas você não sabe o porquê.

Modelo explicável (ex.): Uma árvore de decisão mostra "argumentos > decisão". Então ela se explica, rs.

Evita discriminação algorítmica

Técnicas comuns de XAI

Shap . Lime . Attention Maps . Saliency Maps . Feature importance.

Resumo, fonte:

Livro: Interpretable Machine learning. Christoph Molnar (gratuito Github. fiz fork)

TCC. Cidadão.AI ☆

Nos últimos dias/semanas tenho pesquisado muito sobre arquitetura distribuída. Para tanto algumas alterações estruturais foram implementadas no projeto.

São elas:

- Divisão de Tarefas:

1/ - repositório docs: É o cartão postal do projeto! aqui vai ficar (já implementado) o HUB de documentação. Um lugar para apresentar o projeto. Feito com HTML. CSS TailWind e Javascript.

2/ - repositório backend. Vai ficar (já está) a implementação do backend. (processamento dados API + agentes do sistema + RAG + LLMs) python. Container c/ docker/Kubernetes. (documentação técnica (copia do layout hub) deploy: huggingface

3/. frontend: É o repositório onde o cidadão.AI vai ganhar vida: Dashboards e o chatbot! Next.JS. Shadcn.UI. TailWind. CORS. Conexão back com API.

4/. models. (ainda em fase de planejamento:') mas onde serão implementados os modelos.

TCC · Cidadão.AI . Repositórios em 24.06.25

1. Depois de implementar os 4 repos, comecei pelos elementos do front-end. afim de ter um conjunto de mockUps mais conciso. Padronizei o hub de documentação e migrei o layout da documentação técnica para um layout em conformidade como hub de documentação.
2. Para se criar um efeito de maior impacto, "batizei" os agentes de IA como nomes simbólicos brasileiros personalidades históricas, elementos culturais, folclóricos e de impacto. A ideia é ser uma IA Brasileira e cidadã, então acho que nomear os agentes foi um passo significativo, para engajar o projeto.
3. Ao revisar a arquitetura do back-end, propus mais alguns agentes (total 16!) rs.

Foi criado o MANIFESTO cidadão.AI, onde estabelecemos a filosofia por trás da construção do software.

4. Com base no planejamento, criei o cidadão.ai-models onde irei trabalhar na implementação dos modelos.

Cidadão.AI - CI/CD

Pipeline CI/CD:

1. CI - Continuous Integration
   Entregar código novo com freqüência, várias vezes ao dia, testando automaticamente para evitar conflitos e bugs.

2. CD - Continuous Delivery
   Automatizar o processo de entrega para ambientes de staging ou produção.

Ferramentas Populares:

1. GitHub, Jenkins, Azure pipelines (CI/CD geral)
2. Vercel, Netflix, Heroku, AWS pipeline (deploy cloud)
3. Docker, Kubernetes, ArgoCD (conteinerização)
4. (Testes) Pytest, Jest, Cypress
5. Prometheus, Grafana (monitoramento)

RESumo. Fonte: Github docs. Google Cloud - Guia de CI/CD manual: Atlassian Guide atlassian.com/continuos-delivery

Cidadão. AI - Práticas Modernas de UI/UX

esta pesquisa se refere às técnicas mais modernas, pois já conheço as bases de Ferramentas Web (tecnologias Webs)

1. Design Centrado no usuário
   - público-alvo: todos os brasileiros que acessam IA?? estudantes?
   - comunidade científica?

   não: Para TODOS!

2. TailWind 4 + shadcn/ui

3. Responsividade by Design.

4. Dark/Light mode consistente

5. Acessibilidade (a11y) (parcialmente nossistemas) i18n

6. Microinteracões. (usuário precisa sentir que algo aconteceu)

7. Hierarquia Visual e legibilidade.

Cidadão.AI : Design Science Research (DSR)

DSR é uma metodologia de pesquisa que tem como objetivo criar artefatos inovadores (sistemas, modelos, métodos, arquiteturas, algoritmos) que resolvam problemas concretos e ao mesmo tempo contribuam para o conhecimento científico.

1. Pilares do DSR: (Hevner, 2004)
   1. Relevância 2. Rigor 3. Design
2. Etapas do DSR (Segundo Peffers, 2007)
   1. Identificar o problema
   2. Definir objetivos da solução
   3. Projetar e desenvolver o artefato
   4. demonstrar o artefato em ação
   5. Avaliar seu desempenho
   6. Comunicar resultados

#

PS: Melhor metodologia que caracteriza minhas idéias.

JCC. Cidadão.AI. Empreendedorismo (1)

Branding! Paleta de Cores! Produto & Marca.

~ Brainstorm! ~

Filosofia Base! Nosso agente-Base Terá um nome!

Deodoro da Fonseca.

(militar, proclamador da República)

> Conceito será aplicado a todos os Agentes!

A decisão de branding & Marketing do projeto Cidadão.AI (documentação hub) é mostrar de forma internacionalizada: (150-639-1) (javascript: data-i18n)

» Jenom BPC-47 implementado, é suficiente?

1. Nossa filosofia e propósitos (bem claros!)
2. Como estamos propondo (e já fazendo) para que essa visão seja alcançada?
3. O produto final já está consolidado!
   - inicialmente implementado como
     - Mock-Ups Visuais
     - Protótipos já em produção para validação. BANCA III

- paleta de cores bem definidas com as cores nacionais (mais focado em verde & amarelo) e interface moderna e responsiva. Embora documentação seja bilíngue pt-BR, en-US; a interface dos dashboards e modelo de conversação (chat bot) será exclusivo em PT. (porém a interface está adaptada para suporte a múltiplas linguagens então é fácil adaptação via JavaScript: data-i18n).

A logomarca:

Cidadão.AI

→ letras grossas com efeito degradê com as cores do Brasil Tailwind CSS adaptável

inspirado no conceito de ágora & fórum "onde nasce a democracia" pilares gregos = estabilidade ícone dinâmico = adaptável simples de implementar.

botão/info reciclado de outros projetos Cidadão.AI

Heurísticas de Nielsen: conceito conhecido desde o Técnico em Informática integrado ao Ensino Médio (IFSULDEMINAS). São conjuntos de boas práticas também serão pesquisadas as tecnologias/ou frameworks mais modernos e que tem bons conceitos de avaliação na comunidade. Buscar apenas por software livre, nossa filosofia, hehe!

Segundo pesquisas iremos implementar conceitos modernos de UI/UX (user-interface, user experience)

Vinculações já estabelecidas:

→ Aracele (orientadora)

→ IFSULDEMINAS ↓ rever isso a questão da vinculação!

Vinculações por paridade (?) (isso existe? vou andar mesmo assim –

1 – Open Gov. (badge aplicada no github)

2 – SDG-16 – Meta Nações Unidas 2030

Cidadão.AI - OSINT

Porquê o meu tempo trabalhando com coleta de dados via OPEN-SOURCE INTELIGENCE foi decisivo na ideia original do projeto? (Que, inclusive, é a minha experiência de análise de dados via API usando ML.) Será que a Alessandra, minha manager na Ruvix, Inc. pode assinar meu estágio (ela disse que faria, se eu precisasse.)

IMPORTANTE!

Cidadão. A1 - Engenharia / Arquitetura em Sistemas de Informação: Boas Práticas

Conjunto de princípios técnicos-metodológicos voltados à concepção, desenvolvimento e manutenção de sistemas robustos, escaláveis e auditáveis. Envolve o uso disciplinado de padrões de projeto, modularidade, separação de responsabilidade, e modelos conceituais aderentes aos requisitos funcionais e não-funcionais. Na arquitetura, enfatiza-se governança estrutural do sistema, definindo componentes, fluxos, camadas e contratos de integração, sob enfoques como Domain-Driven Design, clean architecture e microserviços. Essas práticas garantem adaptabilidade e escalabilidade ao sistema.

Essas boas práticas incluem ainda o uso sistemático de documentação técnica, testes automatizados, versionamento semântico e integração contínua (CI/CD), assegurando a rastreabilidade e a manutenibilidade ao longo do ciclo de vida do Sistema.

A engenharia de sistemas também valoriza a aplicação de princípios SOLID na orientação a objetos

facilitando a coesão interna dos módulos e promovendo o desacoplamento entre componentes. Já a arquitetura além de definir os controles estruturais, atua como instrumento de governança, direcionando decisões críticas relacionadas a performace, segurança, escalabilidade e interoperabilidade. Nesse sentido, práticas como DESIGN ORIENTADO AO DOMÍNIO & ARQUITETURA HEXAGONAL tornam-se relevantes ao priorizarem a lógica de negócios sobre detalhes técnicos e permitirem maior flexibilidade frente as mudanças tecnológicas.

No cenário atual, dominado por arquiteturas distribuídas, adota-se cada vez mais o uso de microserviços orquestrados por contâiners (docKer) e sistemas de gerenciamento como o Kubernetes. Essa abordagem promove uma separação de responsabilidades clara, escalabilidade horizontal - benéfica em ambientes na nuvem.

BASS, CLEMENTS, KAZMAN. Software architecture in Pratice, 2012
EVANS. Domain-Driven Design, 2003
FOWLER, Martin 2002
SOMMERVILLE, Engenharia de Software, 2019
PRESSMAN, Eng. de Software, uma abordagem Prof. 2016

Cidadão.AI - Deep Learning
- Aprendizado profundo

Subcampo da aprendizagem de máquina baseado em redes neurais profundas (múltiplas camadas).

. Modela representações hierarquicas e não-lineares dos dados.

. Evolui a partir de redes neurais simples (ex.: perceptron)

. Avanços técnicos: GPU, Big data, alg. (backpropagation)

. Inspirado na organização do córtex visual humano.

. Embora extremamente eficaz, o deep learning enfrenta críticas quanto a sua "caixa preta" interpretativa, necessidade de grandes volumes de dados rotulados e elevado custo computacional. No Cidadão.AI é aplicado técnicas de deep learing em agentes como Chaluaiê e Machado de Assis.

VASWANI. et al, Attention is All you Need. 2017

GOODFELLOW, BENGIO, COURVILLE. Deep Learng. 2016

SILVA, SANTOS, OLIVEIRA. TransparenciA: Deep Learing para auditoria governamental Brasileira. 2023.

Cidadao. AI - Docker

Docker é uma plataforma de virtualização leve baseada em contâineres, que permite empacotar aplicações e suas dependências em ambientes isolados, reproduzíveis e portáveis. Do contrário das máquinas virtuais tradicionais, os cointêneres Docker compartilham o mesmo Kernel do sistema operacional, tornando o overhead de execução significativamente menor. Docker surgiu em 2013 como projeto open-source da empresa dotCloud (Docker, Inc) e rapidamente se tornou padrão em ambientes DevOps & MLOps devido a sua agilidade no provisionamento, consistência entre ambientes de desenvolvimento e produção, e compatibilidade com Orquestradores, como o Kubernetes.

Filosofia Docker: privilegia a imutabilidade da infraestrutura (build once, run anywhere).

No backend do Cidadão.AI, o Docker é essencial para o encapsulamento e a padronização da infraestrutura de execução dos agentes inteligentes. Cada componente - como Zumbi, Dandara ou Abaporu - pode ser empacotado como contâiner independente, facilitando o

desenvolvimento local, testes automatizados e deployment em nuvem com Kubernetes. A combinação Docker + Kubernetes garante escalabilidade horizontal, balanceamento de carga e recuperação automática de falhas além de possibilitar orquestrar eficientemente os agentes especializados com workloads paralelos em ambientes distribuídos. A utilização de Docker Compose também viabiliza testes locais integrados com Redis, ChromaDB em pipelines com CI/CD com GitHub Actions.

MOUAT. Using Docker: Developing and Deploying Software with containers. O'Reilly, 2015

MERKEL. Docker: lightweight Linux Containers. Linux Journal, 2014

VILLAMIZAR, Conference on Cloud computing (CloudCom 2015 2015

Cidadão. AI · Kubernetes

Kubernetes, (K8s), é um sistema open-source de orquestração de contêineres criado originalmente pela Google e mantido atualmente pela Cloud Native Computing Foundation (CNFC). Sua função central é automatizar o deployment, a escalabilidade e o gerenciamento de aplicações containerizadas em ambientes distribuídos; promovendo alta disponibilidade, balanceamento de carga, autoreparo e atualizações contínuas (rolling updates).

Baseados em princípios inspirados na infra-estrutura do BORG (sistema interno Google), o Kubernetes introduz abstrações como pods, services, deployments e configMaps para organizar e manter aplicações em clusters de máquinas físicas ou virtuais. Seu plano de controle (control pane) supervisiona o estado desejado do sistema, enquanto os Workers Nodes executam os contêineres gerenciados por Kubelets. É suportado por todos os ambientes de nuvem (AWS EKS, Azure AKS, Google GKE

No cidadão.AI, utiliza-se Kubernetes como núcleo de sua infra-estrutura de execução distribuída para sustentar a arquitetura multi-agente. Cada

agente inteligente é containerizado via Docker e orquestrados como pods independentes no cluster. O Kubernetes viabiliza auto-scaling baseado em cargas* rolling deployments sem downtime, reinício automático de modelos falhos (agentes) e balanceamento de requisições entre instâncias com alta disponibilidade. O uso de Config Maps & Secrets assegura parametrização segura e auditável do sistema

BURNS, GRANT, OPPENHEIMER, BREWER, WILKES
Borg, Omega and Kubernetes ACM Queque. 2016

HIGHTOWER, BURNS, BEDA: Kubernetes: Up and running O'Reilly, 2022.

CNFC, documentação. acesso em 27 Julho de 2025. Kubernetes.io/docs.

* verificar relação com outras engenharias. vidaseal

P.S.: Pesquisa feita! Engenharia de Portos/Logística

Cidadão.AI - LLMs

LLMs: Modelos de Linguagem de grande escala.

· Definição: Modelos de linguagem de grande escala (LLMs Large Language Models en-US) são arquiteturas baseadas em aprendizado profundo (deep learning) treinadas sobre vastos corpora textuais com o objetivo de capturar padrões estatísticos complexos da linguagem natural. Fundamentam-se majoritariamente, na arquitetura Transformer, conforme proposto por VASWANI et al (2017) e utilizam mecanismos de atenção para modelar dependencias de longo alcance entre palavras em seqüência.

· 2018 → surgimento do BERT (nós usamos BERTimbau) GPT-2, (GPT-3 e GPT-4), culminando em capacidades emergentes como o raciocínio semantico, geração de texto coerente e adaptação à tarefas. Sua eficácia decorre do pré-treinamento não-supervisionado com fine tuning supervisionado e mais recentemente do uso de aprendizado por reforço com feedback Humano (RLHF).

Tem notório desempenho, mas enfrentam desafios:

1. Elevada complexidade computacional.
2. Opacidade algorítmica
3. Riscos de alucinação textual
4. Dificuldades em garantir robustez factual.
5. Suscitam debate ético sobre viés, governança algorítmica & reprodutibilidade científica.

No contexto do Cidadão.AI os LLMs estruturam a base cognitiva de múltiplos agentes especializados, como Machado de Assis (textual agent) e o Ayrton Senna (semanticRouter), permitindo interpretação semântica de contratos, classificação de intenções e síntese textual explicável. O sistema se beneficia de modelos LLMs domain-specific treinados em dados governamentais brasileiros. Essa aplicação confere aos agentes inteligência linguística adaptativa, crucial para auditabilidade e democratização do acesso a informação pública; especialmente com sua integração com LangChain e modulos SHAP e Lime.

⚠ REFERÊNCIAS IMPORTANTES

VASWANI, Ashish et All. Attention is all you Need. Advances in Neural Information Processing Systems, v. 30, 2017.

OPENAI. GPT-4 Technical Report. OpenAI Research Paper, 2023 (lindo demais!)

BOMMASANI, Rishi (et al) On the Opportunities an Risk of Fundation Models (CRFM), 2021

CHOLLET, François. Deep Learning with python 2021.

Minha experiência: (deep learning)

- Deep Learning for Business
- Coursera
Yoonsei University

Cidadão. AI. Pesquisa Multimídia

Série: Connected (pt-BR: A Era dos Dados)

- Baseia-se na Ciência dos dados comple-Netflix xos, com foco em padrões, redes e interdependência.
- Perspectiva interdiciplinar (física, computacão, ecologia, matemática, ciência dos dados).
- Populariza conceitos como teoria da informacão, big data e modelagem computacional.
- Tudo está conectado!: Eventos humanos e tecnológicos revelam padrões interdependentes.
- Fenômenos globais emergem de interaçãis locais).
- Tecnologias de rastreamento, biometria e predição.
- Uso de simulações para prever fenomenos complexos
- A série mostra basicamente:

como a ciência transforma o <u>invisível</u>
em <u>visível</u>.

Cidadão.AI · RAG

RAG - Retrieval-Augmented Generation: Uma arquitetura que funde busca inteligente com geração de linguagem natural, é o pilar para sistemas de IA conversacional e assistentes inteligentes. É uma técnica híbrida que combina:

- R: (Retrieval): Recuperação de informações relevantes de uma base externa (ex. documentos, pdfs, banco vetorial)
- G: (Generation): Com base nos dados recuperados, um modelo de linguagem gera respostas contextualizadas, completas em linguagem natural. (interpreta e sintetiza as evidências trazidas da base com alta relevância semântica

O uso da arquitetura RAG no Cidadão.AI garante:

- Precisão factual: Evita alucinações ao usar somente dados recuperados.
- Atualização contínua: O modelo não depende de re-treinamento para acessar novas informações.
- Auditoria & Rastreabilidade: É possível mostrar de onde veio cada informação.
- Escalabilidade Semântica: Permite respostas contextuais complexas com dados massivos

- Explicabilidade: Favorece o uso de técnicas XAI e justificativas textuais.

Stack RAG Cidadão.AI

1. Embeddings: BERTimbau + Hugging face: codificação semântica.
2. Armazenamento: Chroma DB + FAISS: base vetorial
3. Recuperação: Lang Chain Retriever: busca semântica
4. Orquestração: LangChain Agents + FASTAPI: coordenação dos agentes.

LEWIS, Patrick et al. Retrieval-Augumented Generation for Knowledge-Intensive NLP tasks. Advances in Neural Information. Processing systems. V.33, 2020.

LANGCHAIN. doumentacao oficial.

Cidadão.AI - Redis

REmote DIctionary Server é um banco de dados NoSQL orientado a chave-valor que trabalha majoritariamente em memória, mas com a opção de persistência em disco. Foi criado por Salvatore Sanfillippo em 2009.

- Alta velocidade: Armazena dados em RAM. Leitura e escrita em milisegundos.
- Estrutura de dados: Além de strings, Redis suporta listas, conjuntos, hashes, sorted sets, bitmaps, hiperlogs e stream
- Cache inteligente: Ideal para caching com expiração automática.
- Pub/Sub: Comunicação entre serviços com modelo publisher/subscriber.
- Redis é baseado em loop de eventos assíncrono altamente otimizado.
- Master-slave replication: clusters escaláveis horizontal.
- Pode gravar snapshots (RDB) ou log de comandos (AOF)
- Monitoramento: MONITOR, SLOWLOG & INFO.

REDIS.IO - documentação.
CARVALHO, Andre. Alta performace com Redis, 2020.
WOOD, Joshua. Learning Redis. O'Reilly, 2022.

Framework: Migração total da documentação 30/06

Docussauro (Mantido pela Meta)

1. Claude extraiu todo o conteúdo e colocou em um .md
2. Mantive meus assets (botão, header) para referência
   2.) Backup
3. Iniciei npx docussaurus na pasta limpa.
4. Claude → Migrou conteúdo do .md para o novo sistema do framework docussaurus.

Mock-Up:

5. Ifcionar o agente claude para UI/UX. com bases fundamentados nas heurísticas de Nielsen.

comecei a migração 30/06

ultimo update 31/07 : 18 : 43
- nova versão quase pronta p/ substituir a anterior
muito mais fácil de manter!

migração completa : 31.07 : 22:30 !!

Filosofia da Tecnologia

Linha do tempo:

* Pré História Conceitual (até 1800)

- ~350 a.c. Aristóteles: Téchne vs Episteme
  - Primeira distinção sistemática entre conhecimento prático (téchne) e teórico (episteme). Base conceitual para tecnologia como forma específica de conhecimento aplicado.

- ~100 a.c. Vitrúvio: "De Architectura"
  - Firmitas, Utilitas, Venustas - ainda hoje o framework fundamental para avaliar sistemas técnicos
  - Primeira teorização sobre a relação entre função durabilidade e estética em artefatos.

- ~1620 Francis Bacon: "Novum Organum"
  - "Scientia potentia est": conhecimento como poder sobre a natureza.
  - Método científico como tecnologia de produção de conhecimento

* Fundações Modernas:

- 1844 - Karl Marx: "Manuscritos Economicos-Filosóficos"
  - Alienação tecnológica: quando o trabalho se torna estranho ao trabalhador.
  - Tecnologia como extensão e alienação das capacidades humanas.

- 1867: Karl Marx: "O Capital, vol. 1"
  - Análise das Maquina como cristalização de relações sociais
  - Tecnologia não é neutra - carrega as contradições do sistema que a produz.

- 1889: Ernst Kapp: "Grundlinien einer Philosophie der Technik"
  - Primeiro uso do termo "filosofia da Tecnologia"
  - Teoria da Projeção orgânica: ferramentas como extensão do corpo humano.

* Era Industrial

- 1927: Martin Heidegger: "Ser & Tempo"
  - Zuhandenheit (ser-à-mão): como nos relacionamos com ferramentas quando funcionam
  - Base para se pensar interfaces e UI/UX
  60 anos de esse conceito existir!.

- 1930: Oswald Spengler: "O homem e a técnica"
  - Tecnologia como expressão da vontade de poder faustiana.
  - Crítica ao otimismo tecnológico liberal.

- 1936: Walter Benjamin: "A Obra de Arte na Era da Reprodutibilidade Técnica"
  - Aura vs reprodução mecânica
  - Como a Tecnologia transforma não apenas o que fazemos, mas como percebemos.

- 1944: Max Horkheimer & Theodor Adorno: "Dialética do Esclarecimento"
  - Quando a técnica se torna fim em si mesma
  - Crítica fundamental ao Sillicon Valley) avant la lettre

* Maturidade Conceitual: (1950/1980)

1954 Martin Heidegger: "A Questão da Técnica"

· gestell (enquadamento) tecnologia moderna como forma de revelar o mundo.

· natureza e humanos reduzidos a "reserva disponível" (bestand)

· Marco conceitual decisivo: ainda hoje a maior crítica ao pensamento tecnocrático.

1958: Hanna Arendt: "A Condição Humana"

· vita activa: labor, trabalho, ação

· tecnologia moderna ameaça a capacidade humana de agir politicamente.

1964: Jacques Ellul: "A Sociedade Tecnológica"

· autonomia da técnica: sistema técnico desenvolve lógica própria.

· eficiência como único critério de avaliação.

1964: Herbert Marcuse: "O Homem Unidimensional"

· Racionalidade Tecnológica como forma de dominação.

· "tolerância repressiva": como sistemas técnicos neutralizam oposição.

- 1967: Jacques Derrida: "Da Gramatologia"
  - Escritura como techné originária
  - desconstrução da oposição natural/artificial

* Era da Computação (1980/2000)

- 1980 Jean-François Lyotard: "A Condição Pós-Moderna"
  - saber como mercadoria informacional
  - performatividade como critério de legitimação.
- 1985: Donna Haraway: "Manifesto Ciborgue"
  - dissolução das fronteiras natureza/cultura, humano/máquina
  - tecnologia como possibilidade de libertação, não apenas dominação.
- 1986: Bruno Latour: "Ciência em Ação".
  - actor-network theory: humanos e não humanos como atores.
  - tecnologia como mediação, não mera ferramenta.
- 1995: Pierre Lévy: "O que é virtual?"
  - virtualização como processo ontológico, não técnico
  - cyberespaço como novo espaço antropológico

* Era Digital Contemporânea: (2000/presente)

- 2001: Yuk Hui: "On the existence of digital objects"
  - ontologia dos objetos digitais
  - como pensar existência no tempo digital.
- 2011: Vilém Flusser: "Into the universe of technical Images" (póstumo)
  - imagens técnicas vs imagens tradicionais
  - programação como nova forma de escrita
- 2013: Karen Barad: "Meeting the Universe Halfway"
  - realismo agencial: matéria e significado co-constituem realidade.
  - intra-ação vs interação
- 2016: Yuk Hui: "Cosmotechnics"
  - crítica ao universalismo tecnológico ocidental
  - cada cosmologia produz sua própria relação com a técnica.
- 2019: Shoshana Zuboff: "Capitalismo de Vigilância"
  - economia de extração comportamental
  - poder instrumental como nova forma de poder

Tensões Conceituais Persistentes

. Determinismo vs Construtivismo
. Neutralidade vs Política
. Progresso vs Ambivalência
. Universal vs Particular