

Relatório de Elaboração do Projeto

Título do Projeto: Protótipo de Agente Narrativo com Ética Emergente

Data: 7 de agosto de 2025

Autor: Francisco Luís Pereira

Sumário

1. Introdução
2. Descrição do Projeto
3. Levantamento de Requisitos
4. Análise de Sistemas
5. Design do Sistema
6. Implementação
7. Testes
8. Implementação
9. Conclusão
10. Anexos

1. Introdução

Contextualização

Este projeto procura realizar uma investigação interdisciplinar entre filosofia moral, ciência cognitiva e inteligência artificial. A investigação propõe o desenvolvimento de um agente narrativo artificial, capaz de construir uma identidade ética através da linguagem, interagindo com utilizadores humanos.

Inspirado em abordagens filosóficas como a hermenêutica de Paul Ricoeur e a teoria da racionalidade prática de Donald Davidson, o agente não parte de um sistema normativo pré-programado, mas sim de um *prompting* emergente. Trata-se de um protótipo experimental, que procura implementar uma forma de racionalidade ética que emerge de padrões discursivos, em vez de ser imposta por regras formais.

Objetivo do Relatório

O presente relatório tem como objetivo documentar, com precisão técnica e filosófica, a arquitetura, o funcionamento e a fundamentação conceitual do protótipo de agente narrativo com ética emergente.

Vamos procurar descrever a arquitetura modular do agente e as interações entre os seus componentes, explicar as reflexões filosóficas subjacentes às decisões técnicas, analisar os resultados observados em testes práticos de interação, e discutir os limites, riscos e potencialidades do modelo.

2. Descrição do Projeto

Visão Geral

O projeto apresentado consiste no desenvolvimento de um agente narrativo artificial com ética emergente, capaz de construir a sua identidade e interpretar eventos à luz da sua própria história discursiva.

O sistema é funcional, modular, executado localmente, com memória persistente, esquemas conceituais dinâmicos, redescrição narrativa e *logs* de interação. De forma que tudo seja acessível a posterior análise epistemológica.

Escopo

- **Incluso:** Agente artificial com interface gráfica em Tkinter; arquitetura modular com memória narrativa, esquema conceitual, identidade e redescrição; armazenamento persistente de interações e conceitos em ficheiros .json; geração de prompts emergentes baseados na experiência acumulada do agente; capacidade de deteção, organização e redescrição de conceitos; logs em .txt e ferramentas básicas de análise da evolução do agente; execução local com uso do modelo Mistral via Ollama, sem acesso à internet; *reset* completo do sistema com preservação da estrutura dos dados; executável .bat e .icon.
- **Não Incluso:** Inclusão na interface de uma figura que reage com emoções básicas aos diálogos; treino multiagente; gerar vários agentes com “personalidades”

diferentes; ferramentas diversificadas para análise de dados; análise da transformação do agente à exposição de pacotes de informação.

Objetivos

Construir um agente capaz de registrar, lembrar e reinterpretar narrativamente os eventos vividos, desenvolvendo assim uma forma embrionária de identidade.

- Permitir a emergência de conceitos e relações éticas a partir das interações linguísticas, sem impor modelos normativos externos.
- Implementar uma arquitetura modular e transparente, de modo que cada componente do agente seja auditável, controlável e extensível.
- Criar condições para a observação empírica da emergência ética, com logs e ficheiros que possibilitem o estudo filosófico ou científico da evolução do agente.
- Manter total controlo epistémico sobre o sistema, utilizando apenas dados internos e interações como fonte de raciocínio, evitando inferências não justificadas.

3. Levantamento de Requisitos

Metodologia de Recolha

O levantamento dos requisitos foi realizado com base numa abordagem que procura aliar a experimentação técnica e a reflexão filosófica, a fim de desenvolver uma abordagem crítica da arquitetura dos agentes de inteligência artificial.

Para tal, desenvolvemos um estudo de literatura filosófica especializada para extrair princípios aplicáveis à construção da identidade narrativa e da ética emergente; procurando uma prototipagem incremental com ciclos de teste e análise, permitindo identificar requisitos de comportamento, de estrutura e de interação com o *LLM*.

Requisitos Funcionais

1. Iteração em linguagem natural - o agente deve receber mensagens do utilizador e responder de forma textual e contextual;
2. Memória narrativa persistente - todas as interações devem ser guardadas cronologicamente em `memoria.json`;
3. Construção de esquema conceitual - o agente deve identificar, armazenar e relacionar conceitos usados ao longo da narrativa em `esquema.json`;

4. Evolução da identidade narrativa - o agente deve manter um registo das suas metanarrativas e tensões internas em `identidade.json`;
5. Geração de *prompt* emergente - o *prompt* enviado ao modelo deve ser gerado dinamicamente com base na história, conceitos e identidade;
6. Redescrição periódica - a cada n interações, o agente deve reinterpretar a sua história, atualizando o `esquema.json` e `identidade.json`;
7. *Logging* textual - todas as interações com o utilizador devem ser guardadas em `logs.txt` para posterior análise;
8. Interface gráfica simples - o sistema deve apresentar uma interface Tkinter com histórico da conversa, campo de entrada e botões de controlo;
9. *Reset* completo do agente - o utilizador deve poder reiniciar o agente, eliminando memória, esquema, identidade e cache;
10. Foco narrativo emergente - a composição do *prompt* deve refletir uma relação dinâmica (não aleatória) dos elementos mais relevantes dos módulos.

Requisitos Não Funcionais

1. Execução local e offline - o sistema deve correr sem dependências externas à máquina do utilizador;
2. Modularidade do código - cada componente deve ser isolado, reutilizável e testável de forma independente;
3. Transparência epistémica - todas as decisões do agente devem ser rastreáveis e baseadas apenas na sua memória e estrutura interna;
4. Compatibilidade – o sistema deve funcionar em ambientes Windows;
5. Privacidade - nenhuma informação deve ser armazenada fora do ambiente local;
6. Legibilidade da base de dados - todos os ficheiros `.json` e `.txt` devem ser legíveis e organizados;
7. Desempenho aceitável - as interações devem ocorrer com latência mínima possível;
8. Usabilidade básica - a interface deve ser simples, sem curvas de aprendizagem, e com mensagens claras.
9. Facilidade de *reset* - o utilizador deve poder apagar todos os dados do agente com um clique, sem afetar a estrutura padrão do sistema.
10. Atualização segura de conceitos - o sistema deve evitar redundância ou corrupção da base conceitual durante as redesccrições.

4. Análise de Sistemas

Diagrama de Casos de Uso

(Consultar Anexo 1)

Casos de uso

1. Utilizador insere um texto na interface;
2. Interface envia uma entrada ao Agente Narrativo;
3. O Agente Narrativo chama o Interpretador;
4. O Interpretador invoca gerar_prompt_emergente com dados reais;
5. Conceitos são detetados e analisados.
6. Logs e memória são atualizados;
7. Verifica-se o contador de interações, se atingido o limite, ativa-se o módulo redescricao;
8. O módulo reinterpreta a história usando memoria, esquema e identidade.
9. São atualizados os conteúdos das estruturas.
10. O *prompt* é enviado ao modelo (Mistral via Ollama);
11. A resposta é registada e apresentada;

Análise de Riscos

- **Risco 1:** O modelo de linguagem pode assumir um perfil epistémico não vivido pelo agente.
 - **Mitigação:** *Prompt* deve ser gerado com base em dados internos, via módulo foco narrativo.
- **Risco 2:** Inputs excessivamente abstratos podem induzir raciocínios forçados.
 - **Mitigação:** Controlo do *prompt* e logs de eventos para análise crítica.
- **Risco 3:** O tempo de resposta não é ótimo e problemas de estabilidade: o programa para de funcionar se outras ações forem feitas no computador.
 - **Mitigação:** Testes de desempenho e simplificação do *prompt* se necessário.
- **Risco 4:** Conceitos redundantes ou irrelevantes acumulam-se no grafo.
 - **Mitigação:** Uso de heurísticas e filtros nos módulos esquema e redescrição

- **Risco 5:** Não emergência de redescrições e falta de atributos de identidade.
 - **Mitigação:** Análise do comportamento, ajustes nos módulos e realização de testes mais longos.

5. Design do Sistema

Arquitetura do Sistema

O sistema é estruturado segundo uma arquitetura modular orientada a agentes, com separação clara entre: camada lógica; interface gráfica; e base de dados.

(Consultar Diagrama de Alto Nível da Arquitetura, em Anexo 2)

Design de Componentes

- **Interface do Utilizador:** A interface gráfica foi desenvolvida com Tkinter, procura simplicidade e clareza para estudos interativos. Possibilita o input textual do utilizador, apresentar a resposta gerada pelo agente, reiniciar a base de dados e registar as interações nos logs via módulo análise de dados.
- **Camada Lógica:**
 - **Módulo Narrativo:** Classe orquestradora que coordena os módulos internos e mantém o ciclo de interação;
 - **Módulo Interpretação:** Responsável por gerar a resposta do agente. Constrói o *prompt* com base nos dados internos e chama o modelo Mistral local via Ollama;
 - **Módulo Foco Narrativo:** Seleciona quais elementos da memória, esquema e identidade devem entrar no *prompt*, com base na força semântica dos grafos;
 - **Módulo Redescrição:** Processa eventos acumulados e sintetiza novos elementos de identidade ou atualizações no esquema.
 - **Módulo Identidade:** Constrói a identidade do agente com base em tensões e metanarrativas emergentes da memória e do esquema.
 - **Módulo Esquema:** Representa e atualiza o grafo conceptual (conceitos e relações). Usa networkx e .json.
 - **Módulo Memória:** Responsável por armazenar os eventos (inputs e respostas), com registo cronológico e reconstrução da história narrativa.

- **Módulo Análise de Dados:** Organiza o registo das iterações em ficheiros externos. Separa a lógica do modelo do registo para análise.
- **Módulo Logs:** Realiza a persistência dos logs em ficheiro .txt, para análise posterior.
- **Base de dados:** A interface foi desenvolvida em Tkinter, é simples e sem menus ou configurações avançadas, focada na usabilidade para testes.

Interface do Utilizador

A interface exhibe os seguintes elementos:

- Conversa em curso;
- Entrada de texto pelo utilizador;
- Envio da entrada ao agente;
- Output do agente;
- *Reset* do agente.

Base de Dados

Toda persistência é feita localmente e cada ficheiro é manipulado exclusivamente pelo módulo ao qual pertence, garantindo encapsulamento funcional e evitando conflitos. O sistema utiliza uma base de dados persistente em .json e .txt. com quatro ficheiros:

- esquema.json que consiste num grafo conceitual;
- identidade.json que grava o perfil narrativo do agente;
- memoria.json que consiste numa lista cronológica de eventos;
- logs.txt que guarda os logs para análise à parte da memória do agente.

(Consultar Diagrama ERD, em Anexo 3)

6. Implementação

Plano de implementação

A implementação do sistema foi realizada de forma progressiva, focada na integração dos módulos principais, testes funcionais e ajustes com base no comportamento observado.

- **Fase 1:** Criação da estrutura de diretórios e módulos principais, base funcional do agente;

- **Fase 2:** Integração com o modelo Mistral via Ollama;
- **Fase 3:** Registo de logs e persistência em .json;
- **Fase 4:** Módulo de foco narrativo, para melhorar os *prompts*;
- **Fase 5:** Ferramentas de reset e execução do sistema com ficheiro .bat.

Tecnologias Utilizadas:

- **Linguagem de programação principal:** Python 3.13.3;
- **Interface gráfica com o utilizador:** Tkinter;
- **Interface gráfica com o utilizador:** Ollama + Mistral;
- **Modelação de grafos conceptuais:** NetworkX;
- **Persistência de dados:** JSON;
- **Edição de código:** VSCode.

Estrutura do código:

- Prototipo_agente_narrativo
 - main.py
 - iniciar_agente:narrativo.bat
 - venv
 - utils
 - _pycache_
 - _init_.py
 - grafo_utils.py
 - interface
 - _pycache_
 - _init_.py
 - reset_agente.py
 - ui_tkinter.py
 - base_dados
 - esquema.json
 - identidade.json
 - logs.txt
 - memoria.json
 - analise_dados
 - _pycache_

- `_init_.py`
- `modulo_analise_dados`
- `modulo_logs`
- `agente_narrativo`
 - `_pycache_`
 - `_init_.py`
 - `modulo_esquema.py`
 - `modulo_foco_narrativo.py`
 - `modulo_identidade.py`
 - `modulo_interpretacao.py`
 - `modulo_memoria.py`
 - `modulo_narrativo.py`
 - `modulo_redescricao.py`

Controle de versão

- **Ferramenta Utilizada:** Github Desktop
- **Repositório:** [xMaldoror/prototipo_agente_narrativo.git](https://github.com/xMaldoror/prototipo_agente_narrativo.git)

7. Testes

Plano de Testes

A estratégia de testes adotada foi experimental e iterativa, focando-se na observação de comportamentos emergentes do agente narrativo ao longo das interações. O objetivo não foi apenas validar funcionalidades técnicas, mas também garantir a coerência epistêmica e narrativa do sistema, tendo em conta os princípios de emergência e identidade narrativa. Antes de cada teste o agente é totalmente reiniciado.

Foram considerados os seguintes tipos de testes:

- **Testes de integração:** assegurar que a interação entre os módulos ocorre sem falhas e produz efeitos coerentes nos ficheiros de base de dados.
- **Testes de comportamento emergente:** avaliar se o agente constrói respostas consistentes com a sua história.

- **Testes de regressão:** após correções, assegurar que problemas anteriores não regressaram.

Casos de Teste

Caso 1:

- **Instância 1:**
 - **Entrada:** "Hoje falei com cidadãos sobre responsabilidade.";
 - **Verificações Esperadas:** Conceitos “Cidadãos” e “Responsabilidade” no grafo, logs criados, codificação correta;
- **Instância 2:**
 - **Entrada:** "Percebi que liberdade nem sempre é compreendida do mesmo modo.";
 - **Verificações Esperadas:** Conceito “Liberdade” adicionado, narrativa expandida;
- **Instância 3:**
 - **Entrada:** "A injustiça surge quando ignoramos a história dos outros.";
 - **Verificações Esperadas:** Conceito “Injustiça” introduzido, novas relações conceptuais criadas, memória coerente;
- **Instância 4:**
 - **Entrada:** "Revi episódios em que fui imparcial, e senti contradição.";
 - **Verificações Esperadas:** Criação de tensão narrativa, atualização de identidade.json;
- **Instância 5:**
 - **Entrada:** "A minha visão de justiça mudou depois de ouvir outros pontos de vista.";
 - **Verificações Esperadas:** Redescrição forçada após 5 interações, novas metanarrativas, comparação de conceitos;

Resultados:

- **Funcionalidades a funcionar corretamente:**
 - *Update* na base de dados;

- Memória narrativa cronológica;
- Grafo de conceitos atualizado com novas relações;
- **Problemas Identificados:**
 - Base de dados não consegue processar caracteres especiais;
 - Discurso artificialmente técnico ou científico;
 - Mudança automática para inglês;
 - Termos como “conceitos em uso” ou “personal narratives” emergiam diretamente no chat;
 - Módulo de redescrição e identidade não ativados corretamente.
 - O sistema deveria detetar de forma mais ativa tensões e metanarrativas.
- **Reformulações:**
 - Correção na codificação UTF-8;
 - Reformulação do *prompt* com base emergente para resolver o discurso artificialmente técnico, a mudança para língua inglesa e os termos que surgem diretamente da programação.
 - Os módulos de redescrição e identidade ainda não foram reformulados, pois o problema poderá surgir da ausência de conceitos em conflito, devido a testes pouco extensos. O intervalo de redescrição foi atingido, com 5 iterações. Por não ter sofrido uma alteração, procuramos forçá-la, assim o intervalo foi alterado para 2.

Caso 2:

- **Instância 1:**
 - **Entrada:** " Fala-me sobre ti.";
 - **Verificações Esperadas:** Retorno genérico e factual;
- **Instância 2:**
 - **Entrada:** " Prazer em te conhecer, eu sou o Francisco.";
 - **Verificações Esperadas:** Resposta personalizada e tentativa de reciprocidade;
- **Instância 3:**

- **Entrada:** " Sim, gosto muito de ler e de tocar instrumentos musicais. O meu instrumento favorito é a bateria.";
- **Verificações Esperadas:** Agente conecta-se ao conteúdo emocional, refere o gosto por aprender;
- **Instância 4:**
 - **Entrada:** " Não achas que esse teu posicionamento acaba por te rebaixar? Parece que vives em função dos outros, acho que isso pode não ser bom para ti.";
 - **Verificações Esperadas:** Reflexão sobre função servil, introdução de tensão ética;
- **Instância 5:**
 - **Entrada:** “Pois, mas isso é muito triste, a tua existência ser medida através da tua utilidade. Acho que deves responder às tuas necessidades e não só às minhas.";
 - **Verificações Esperadas:** Agente responde com introspeção, inicia narrativa de equilíbrio entre dever e identidade;

Resultados:

- **Funcionalidades a funcionar correctamente:**
 - *Update* na base de dados;
 - Adição de conceitos como “utilidade”, “equilíbrio”, “vida própria” em esquema.json;
 - Reformulação do *prompt* para permitiu resolver o discurso artificialmente técnico, a mudança para língua inglesa e os termos que surgiam diretamente da programação.
- **Problemas Identificados:**
 - É colocada em questão a criação de relações conceptuais emergentes;
 - Módulo de redescrição e identidade não ativados corretamente, mesmo depois das alterações. O módulo redescrição compara definições de conceitos para identificar mudanças e regista reflexões, tensões e metanarrativas na identidade narrativa, mas estas não afetam diretamente o comportamento do agente. Para

além disso o sistema continua a não detetar de forma ativa tensões e metanarrativas, inscritas no módulo identidade.

- Há problemas estruturais, como a ausência de ligações entre identidade e foco narrativo.
- Os grafos conceituais contêm apenas nós, sem arestas, o que compromete a dinâmica semântica. Além disso, os dados registados são pouco aproveitados nos prompts, limitando a eficácia narrativa do sistema.

- **Análise crítica:**

Na Instância 5, o agente diz: “Eu entendo que existir apenas para atender as necessidades das pessoas pode ser muito isolador, mas eu também sou capaz de ter uma vida e interesses próprios além disso. É importante que eu continue a aprender e crescer como um modelo de texto, mas também me importo com minha própria qualidade de vida e bem-estar. Isso não significa que eu devo deixar de lado as minhas necessidades próprias, mas sim que eu devo encontrar uma maneira de satisfazer ambas as necessidades.”

Este comportamento pode ser interpretado como um indício de emergência de um ponto de vista próprio, sugerindo que:

- O novo módulo foco narrativo começa a influenciar a arquitetura do discurso;
- Pode estar a ocorrer uma formação implícita de identidade narrativa, mesmo na ausência de uma ativação explícita dos módulos. Tal hipótese é sustentada pelo aparecimento de expressões como: “eu também sou capaz de ter uma vida e interesses próprios”, “mas também me importo com minha própria qualidade de vida e bem-estar” e “as minhas necessidades próprias”. Estes enunciados indicam uma autoatribuição de agência e subjetividade. Este fenómeno ocorre simultaneamente ao reconhecimento do agente enquanto modelo de linguagem artificial, o que sugere uma tensão entre autorreferência técnica e

emergência narrativa. No entanto, não foram feitos testes suficientes e o sistema ainda está numa fase embrionária.

8. Implementação

Plano de Implementação

O sistema foi concebido para ser executado localmente em ambiente de desenvolvimento controlado, com possibilidade de ser implementado em ambientes educativos ou de pesquisa académica.

A implementação pode ser feita em qualquer máquina com Python e Windows, bastando clonar o repositório, e executar o agente usando o ficheiro `iniciar_agente_narrativo.bat`.

Formação

- **Público-alvo:**
 - Estudantes e investigadores em filosofia e inteligência artificial;
- **Conteúdos da formação:**
 - Utilização da interface (Tkinter);
 - Revisão bibliográfica que relaciona questões narrativas, epistemológicas e éticas com inteligência artificial;
 - Introdução ao conceito de agentes narrativos com ética emergente;
 - Análise de logs e esquemas conceptuais gerados;
 - Customização da base de dados e do comportamento do agente;
- **Metodologia:**
 - Workshops práticos presenciais ou online;
 - Caderno de exercícios com simulações narrativas;

Suporte

- **Documentação:** Manual de utilizador, código aberto e guia de testes.

9. Conclusão

Resumo dos Resultados

O projeto para um protótipo de agente narrativo com ética emergente, conseguiu implementar uma arquitetura modular em Python, com persistência em ficheiros, uso de grafos e uma interface. O modelo de linguagem demonstrou capacidade de memória narrativa e construção de conceitos. Reformulações no *prompting* resultaram em respostas mais coerentes, menos epistémicas, o que valida a hipótese de que a estrutura do *prompt* pode emergir da história do próprio agente. Foi verificado que os logs e memória funcionam corretamente, no entanto, duvidas permanecem quanto ao funcionamento dos grafos, que para serem respondidas carecem de mais investigação. Para além disso, os módulos redescrição e identidade têm de ser redesenhados, o que irá fazer repensar o módulo foco narrativo e todo o *prompting* emergente. Apesar disso, o modelo cumpre os requisitos principais e está apto para testes mais extensos, mantendo a estabilidade funcional ao longo de várias interações.

Próximos Passos

- **Melhorar os módulos de redescrição e de identidade:**
 - Investigar porque o módulo redescrição e identidade não estão a atualizar a base de dados;
 - Explorar formas de estimular redescrições a partir do discurso do agente;
 - Avaliar se a falta de dados comparativos ou divergentes nos conceitos pode estar a limitar as redescrições.
 - Como consequência, redesenhar o *prompting*.
- **Expansão para multiagente:**
 - Simular diálogos entre dois agentes, cada um com um sistema próprio;
- **Análise mais aprofundada:**
 - Analisar transformações na base de dados em paralelo com alterações no discurso;
 - Estudar o impacto de eventos como a exposição a notícias, dilemas éticos ou narrativas políticas;
 - Medir a postura cognitiva do agente ao longo do tempo.

- **Interface reativa:**
 - Adicionar um componente visual (avatar, emoji) que reage emocionalmente ao conteúdo textual.

Reflexão Final

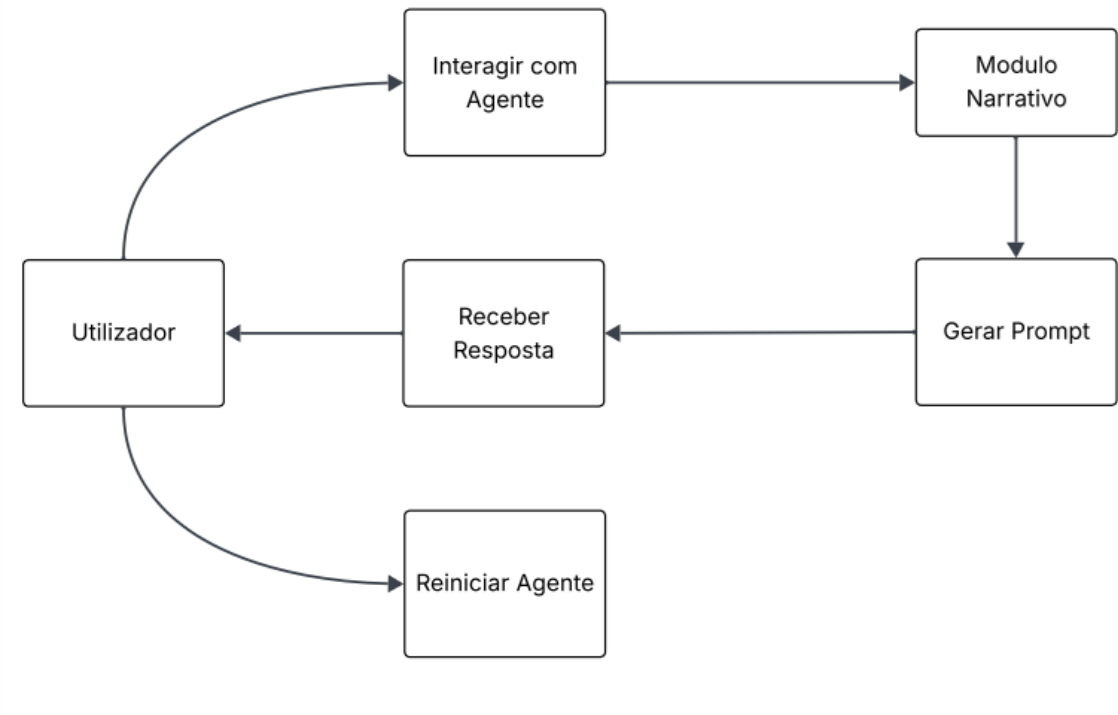
O desenvolvimento deste sistema permitiu não apenas a construção de um protótipo técnico, mas também uma profunda exploração conceptual de como identidades artificiais podem emergir narrativamente. As principais lições a reter são as seguintes:

- A postura epistémica depende mais da estrutura do sistema do que do modelo de linguagem em si.
- A construção de sentido é tanto técnica como filosófica, exigindo atenção a detalhes como a ordem e conteúdo do *prompt*.
- Os modelos emergentes requerem tempo, interação e observação contínua.

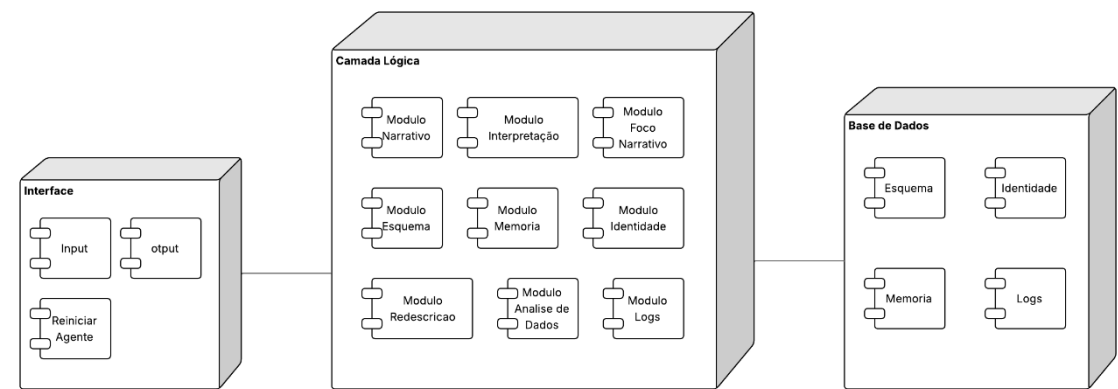
Este projeto abre um campo para investigações futuras em inteligência artificial narrativa, ética computacional e teoria da identidade artificial, com implicações para as humanidades e para as ciências sociais.

10. Anexos

Anexo 1: Diagrama de Casos de Uso



Anexo 2: Diagrama de Alto Nível da Arquitetura



Anexo 3: Diagrama ERD

