UNIVERSIDADE
LUSÓFONA

Ferramenta de Anotação

Trabalho Final de Curso

Relatório Intercalar 1º Semestre

Fábio Lopes - a22103261
Orientador:
Bruno Saraiva
Co-orientador:
Zuil Filho

Trabalho Final de Curso | LEI | 2024/25

www.ulusofona.pt

Direitos de cópia

Este documento foi gerado com o processador (pdf/Xe/Lua)LaTeX e o modelo ULThesis (v1.0.0) Matos-Carvalho 2024.

Resumo

Este trabalho apresenta o desenvolvimento de uma ferramenta de anotação que implementa um Módulo de Chat Disentanglement especializado, concebido para apoiar a criação de datasets anotados para investigação em conversation disentanglement. O módulo disponibiliza uma interface de utilizador onde os anotadores podem identificar e marcar diferentes threads de conversa que ocorrem simultaneamente em dados de chatrooms. O Chat Disentanglement, conforme descrito por Elsner e Charniak (2010), é a tarefa de separar múltiplas conversas concorrentes num único canal de comunicação. A nossa ferramenta de anotação foca-se em fornecer uma experiência de utilizador intuitiva e fácil para minimizar erros de anotação, e será utilizada pelo AISIC Lab (Artificial Intelligence and Social Interaction and Complexity) para criar datasets anotados de conversas de chatrooms, que podem depois ser utilizados por investigadores e anotadores para estudar e desenvolver soluções automatizadas de disentanglement.

Abstract

This work presents the development of an annotation tool that implements a specialized Chat Disentanglement Module, designed to support the creation of annotated datasets for conversation disentanglement research. The module provides a user interface where annotators can identify and mark different conversation threads occurring simultaneously in chatroom data. Chat disentanglement, as described by Elsner e Charniak (2010), is the task of separating multiple concurrent conversations in a single communication channel. Our annotation tool focuses on providing an intuitive and easy user experience to minimize annotation errors, and will be used by the AISIC Lab (Artificial Intelligence and Social Interaction and Complexity) to create annotated datasets of chatroom conversations, which can then be utilized by researchers and annotators to study and develop automated disentanglement solutions.

Conteúdo

Resumo **2**

Abstract **3**

Conteúdo **4**

Lista de Figuras **6**

Lista de Tabelas **7**

1 Introdução **8**
1.1 Enquadramento 8
1.2 Motivação e Identificação do Problema 8
1.3 Objetivos 9
1.4 Estrutura do Documento 9

2 Pertinência e Viabilidade **11**
2.1 Pertinência 11
2.2 Viabilidade 11
2.3 Análise Comparativa com Soluções Existentes 11
2.3.1 Soluções Existentes 11
2.3.2 Análise de Benchmarking 12
2.4 Proposta de Inovação e Mais-Valias 12
2.5 Identificação de Oportunidade de Negócio 12

3 Especificação e Modelação **14**
3.1 Análise de Requisitos 14
3.2 Modelação da Solução 14
3.2.1 Arquitetura da Solução 14
3.2.2 Modelo de Dados 14
3.2.3 API RESTful 15
3.3 Protótipos de Interface 15
3.3.1 Mapa de Navegação 15
3.3.2 Principais Interfaces 15

4 Solução Proposta **20**
4.1 Apresentação 20
4.2 Arquitectura 20
4.3 Tecnologias e Ferramentas Utilizadas 21
4.4 Ambientes de Teste e de Produção 22
4.5 Abrangência 23
4.6 Componentes 23
4.6.1 Frontend 23
4.6.2 Backend 24
4.6.3 Fluxo de Dados 25
4.7 Interfaces 25

5 Testes e Validação **27**
5.1 Introdução . . . 27
5.2 Plano de Testes . . . 27
5.2.1 Participantes e Perfis de Teste . . . 27
5.2.2 Cenários e Tarefas Principais . . . 27
5.2.3 Ambiente de Teste . . . 28
5.3 Recolha e Análise de Dados . . . 28

6 Método e Planeamento **30**
6.1 Metodologia de Desenvolvimento . . . 30
6.1.1 Princípios Metodológicos . . . 30
6.1.2 Organização do Trabalho . . . 30
6.2 Planeamento e Cronograma . . . 30
6.2.1 Fases do Projeto . . . 30
6.3 Análise Crítica ao Planeamento . . . 31
6.3.1 Progresso Realizado . . . 32
6.3.2 Desafios Encontrados e Adaptações ao Plano . . . 32
6.3.3 Implicações no Planeamento Futuro . . . 33

Referências Bibliográficas **35**

Lista de Figuras

3.1 Modelo de Entidade-Relação final do Sistema. 17
3.2 Mapa de Navegação do Sistema. 19

4.1 Diagrama de Arquitectura do Sistema. 26

6.1 Cronograma detalhado do projeto (Gantt Chart) - Estado Atualizado 34

Lista de Tabelas

2.1 Comparação técnica detalhada das soluções 13

3.1 Tabela de Requisitos e Estado de Implementação. 18

1 - Introdução

No contexto atual do AISIC LAB - Artificial Intelligence, Social Interaction and Complexity, pertencente ao CICANT da Universidade Lusófona, surge a necessidade de desenvolver uma infraestrutura modular para anotação de dados, com particular incidência em aplicações de Processamento de Linguagem Natural (NLP). Este projeto visa responder aos desafios específicos encontrados no laboratório, nomeadamente na análise e processamento de interações em ambientes de chat.

1.1 Enquadramento

A anotação de dados é um processo fundamental no treino de modelos de Machine Learning, particularmente importante em aplicações de NLP. Este processo envolve a atribuição sistemática de etiquetas, categorias ou metadados a conjuntos de dados não processados, permitindo que os algoritmos aprendam padrões e características específicas. Os dados anotados funcionam como referência fundamental, possibilitando o treino de modelos para reconhecer e classificar informações de forma precisa.

1.2 Motivação e Identificação do Problema

A motivação principal deste projeto emerge das necessidades específicas identificadas no AISIC LAB, onde a análise de mensagens em grupos de chat, requer ferramentas especializadas de anotação. Através de análises preliminares e feedback dos investigadores, identificámos três desafios fundamentais:

1. **Complexidade das Tarefas de Anotação**:
    - Necessidade de suporte a diferentes tipos de anotação
    - Gestão de múltiplos anotadores e controle de qualidade
    - Requisitos específicos para diferentes domínios de aplicação
2. **Limitações das Ferramentas Existentes**:
    - Ferramentas estabelecidas como o BRAT[1] apresentam limitações tecnológicas e falta de evolução
    - Soluções atuais como o Doccano[2], que utiliza Django como framework backend, impõem uma estrutura mais rígida, o que limita a capacidade de realizar adaptações específicas às necessidades do projeto.
    - Necessidade de maior flexibilidade na modelação de dados e lógica aplicacional
    - Dificuldade de integração com fluxos de trabalho existentes e específicos
3. **Necessidade de Integração Completa do Workflow**:
    - Gestão integrada das fases pré-anotação, anotação e pós-anotação
    - Necessidade de implementação de métricas e análises específicas

[1] *BRAT Rapid Annotation Tool* 2024; *About BRAT* 2024
[2] *Doccano* 2024; *Doccano Developer Guide* 2024

- Suporte a fluxos customizados de processamento de dados
- Integração com pipelines de NLP e análise de dados existentes

A decisão de desenvolver uma solução dedicada, em vez de adaptar ferramentas existentes, fundamenta-se na necessidade de maior controlo sobre todo o processo de anotação. Esta abordagem permite-nos focar no desenvolvimento de funcionalidades específicas para o nosso caso de uso principal - a tarefa de disentanglement de chat - garantindo a flexibilidade necessária para implementar workflows customizados de pré-processamento, anotação e análise posterior dos dados.

1.3 Objetivos

O projeto tem como objetivos principais:

1. **Desenvolvimento de Infraestrutura Modular**:
    - Criar uma plataforma flexível e extensível para anotação de dados
    - Implementar sistema de plugins para diferentes tipos de tarefas
    - Desenvolver interfaces programáticas (APIs) para integração com outros sistemas
2. **Automação e Controlo de Qualidade**:
    - Implementar distribuição automática de tarefas
    - Desenvolver sistema robusto de métricas e validações
    - Criar mecanismos de controle de qualidade e consistência
3. **Suporte a Múltiplos Domínios**:
    - Permitir configuração de diferentes esquemas de anotação
    - Implementar suporte para diversos tipos de dados
    - Facilitar a extensão para novos casos de uso
4. **Usabilidade e Produtividade**:
    - Desenvolver interfaces intuitivas para anotadores
    - Implementar ferramentas de gestão e monitoramento
    - Criar documentação abrangente e guias de utilização

1.4 Estrutura do Documento

Este documento está organizado da seguinte forma:

- **Capítulo 2 - Pertinência e Viabilidade**: Apresenta análise detalhada do contexto atual, comparação com soluções existentes e identificação de oportunidades de inovação.
- **Capítulo 3 - Especificação e Modelação**: Detalha requisitos funcionais e não-funcionais, casos de uso e a modelação da solução proposta.
- **Capítulo 4 - Solução Proposta**: Descreve a arquitetura, tecnologias utilizadas e os componentes principais da solução em desenvolvimento.

- **Capítulo 5 - Testes e Validação**: Apresenta o plano delineado para testar e validar a ferramenta, incluindo a metodologia, cenários e métricas de avaliação.
- **Capítulo 6 - Método e Planeamento**: Detalha a abordagem metodológica seguida e analisa o progresso do desenvolvimento face ao planeamento.
- **Capítulo 7 - Resultados**: (A desenvolver) Discutirá os resultados obtidos nos testes e avaliará o cumprimento dos objetivos.
- **Capítulo 8 - Conclusão**: (A desenvolver) Sintetizará as contribuições do trabalho e apresentará perspetivas futuras.

Os anexos poderão incluir documentação técnica adicional, guiões de teste ou outros materiais de suporte relevantes.

2 - Pertinência e Viabilidade

2.1 Pertinência

O desenvolvimento de uma ferramenta modular para anotação de dados surge como resposta a uma necessidade crítica no campo do processamento de linguagem natural (NLP). A pertinência desta solução é evidenciada por múltiplos fatores.

O AISIC LAB, validou a necessidade desta ferramenta através de feedback direto dos anotadores sobre as limitações das ferramentas atuais, bem como através da avaliação dos investigadores sobre o impacto na qualidade dos dados e análise das necessidades específicas de projetos em andamento.

O desenvolvimento desta ferramenta promete uma redução significativa no tempo de anotação, além de proporcionar uma melhoria substancial na qualidade e consistência dos dados anotados. A solução também facilita a colaboração entre anotadores e oferece suporte flexível a múltiplos tipos de tarefas de anotação, adaptando-se às necessidades específicas de cada projeto.

2.2 Viabilidade

A implementação da solução demonstra-se viável em múltiplas dimensões. Do ponto de vista técnico, a experiência prévia com um protótipo funcional em React, aliada à atual implementação utilizando FastAPI e SQLite, combinada com a disponibilidade de frameworks modernos para desenvolvimento, fornece uma base sólida para o projeto. A arquitetura modular planeada permite um desenvolvimento incremental e sustentável, aproveitando a infraestrutura existente para deployment.

Quanto à viabilidade económica, o projeto beneficia de um baixo custo de desenvolvimento inicial, principalmente devido à utilização de tecnologias open-source. Além disso, apresenta potencial para comercialização futura e promete uma redução significativa nos custos operacionais relacionados à anotação de dados.

A aceitação social do projeto é confirmada pelo feedback positivo dos utilizadores finais e pelo forte alinhamento com as necessidades do departamento. O potencial de aplicação em outros contextos académicos e a contribuição significativa para a pesquisa em NLP reforçam sua relevância no ambiente académico e científico.

2.3 Análise Comparativa com Soluções Existentes

2.3.1 Soluções Existentes

Doccano

O Doccano é uma plataforma open-source para anotação de dados em NLP. A ferramenta oferece suporte a múltiplos tipos de anotação e possui uma arquitetura modular que permite extensões dentro da sua estrutura.

BRAT

O BRAT é uma ferramenta estabelecida para anotação de texto, com foco em simplicidade e interface intuitiva. Apesar de sua maturidade, apresenta limitações em termos de extensibilidade e modernização.

2.3.2 Análise de Benchmarking

A Tabela 2.1 apresenta uma comparação detalhada das características técnicas de cada solução. Esta análise permitiu identificar pontos fortes e limitações de cada ferramenta, orientando o desenvolvimento da solução proposta.

2.4 Proposta de Inovação e Mais-Valias

A solução desenvolvida foca-se especificamente nas necessidades de anotação para a tarefa de chat disentanglement nesta fase inicial, permitindo uma implementação direta e eficiente deste tipo de tarefa. A abordagem modular facilita a adaptação para diferentes necessidades de anotação, com o objetivo futuro de suportar múltiplos tipos de tarefas de anotação, mantendo a simplicidade de uso.

2.5 Identificação de Oportunidade de Negócio

O projeto apresenta potencial para aplicação em contextos acadêmicos e de pesquisa, especialmente em grupos que trabalham com processamento de linguagem natural. A capacidade de adaptação para diferentes tipos de anotação permite atender necessidades específicas de diversos projetos de pesquisa.

Solução	BRAT[1]	Doccano[2]	Solução Proposta
Frontend	jQuery 1.7.1 + jQuery UI, SVG para visualização, JavaScript vanilla, XHTML templates	Nuxt.js framework, Vue.js (implícito via Nuxt), Modern JavaScript/TypeScript	React 18, TypeScript, Componentes funcionais, Hooks customizados
Backend	Python 2.5+, CGI/FastCGI, Bibliotecas JSON	Python 3.8+, Django 4.0+, REST API architecture	Python 3.x, FastAPI, Arquitectura REST API
Formato de Anotação	Arquivos .ann para anotações, Arquivos .txt para texto, JSON para comunicação	REST API endpoints, JSON para comunicação	REST API para comunicação; anotações via base de dados; suporte atual a importação CSV (planeado para mais formatos)
Dados	Sistema de arquivos, Estrutura em diretórios, Sem banco de dados	Database (Django ORM), Suporte a PostgreSQL	Base de dados relacional SQLite gerida via ORM; armazenamento estruturado de dados e anotações.
Deployment	Apache/Lighttpd com CGI, Standalone Python server	Docker containers, Docker Compose support, Production/Development configs	Docker containers, Setup simplificado
Extensibilidade	Sistema via arquivos .conf, Arquitetura modular, Plugins jQuery	Modular Django architecture, REST API extensibility, Frontend component system	Componentes React modulares, API REST extensível (FastAPI)
Estado de Manutenção	Última atualização 2012, Tecnologias desatualizadas, Projeto inativo	Projeto ativo, Tecnologias modernas, Atualizações regulares	Em desenvolvimento ativo, Stack moderna, Iterações frequentes
Documentação	README detalhado, Exemplos incluídos, Tutoriais no código	Documentação estruturada, Guias de desenvolvimento, Diagramas de arquitetura	Documentação focada, Exemplos práticos, Guias de desenvolvimento
Requisitos Sistema	Python 2.5+, Servidor Web com CGI, Browser com SVG	Python 3.8+, Docker (recomendado), Node.js para desenvolvimento	Node.js 18+, Python 3.x, Navegador moderno, Docker (opcional)
Segurança	Autenticação básica, Controle via arquivo	Django authentication, Modern security practices	Autenticação baseada em roles (administrador/anotador)

Tabela 2.1: Comparação técnica detalhada das soluções

[a] *BRAT Rapid Annotation Tool* 2024; *About BRAT* 2024

[b] *Doccano* 2024; *Doccano Developer Guide* 2024

3 - Especificação e Modelação

Neste capítulo, são detalhadas as especificações técnicas da solução desenvolvida. A análise de requisitos é apresentada em primeiro lugar, seguida pela modelação da arquitetura, da base de dados e da API, que em conjunto definem a estrutura e o comportamento do sistema.

3.1 Análise de Requisitos

A tabela seguinte resume os requisitos funcionais (RF) e não-funcionais (RNF) identificados para o projeto, juntamente com o estado final da sua implementação.

3.2 Modelação da Solução

A modelação da solução foi um processo iterativo que resultou numa arquitetura de três camadas, um modelo de dados relacional robusto e uma API RESTful bem definida.

3.2.1 Arquitetura da Solução

A plataforma segue uma arquitetura cliente-servidor moderna e desacoplada, composta por dois componentes principais:

- **Frontend (Cliente):** Uma Single-Page Application (SPA) desenvolvida com a biblioteca **React**. É responsável exclusivamente pela apresentação da interface do utilizador e pela gestão do estado local da UI. Toda a lógica de negócio e manipulação de dados é delegada ao backend através de chamadas à API.
- **Backend (Servidor):** Uma API RESTful desenvolvida com a framework **FastAPI** (Python). É responsável por toda a lógica de negócio, incluindo a autenticação de utilizadores, controlo de permissões, operações na base de dados (CRUD) e cálculos de métricas como o IAA.

Esta separação estrita de responsabilidades (‘separation of concerns‘) garante uma maior manutenibilidade, escalabilidade e a possibilidade de desenvolver e testar os dois componentes de forma independente.

3.2.2 Modelo de Dados

O coração do backend é o seu modelo de dados relacional, implementado com **SQLAlchemy** como ORM (Object-Relational Mapper), que mapeia as classes Python para tabelas numa base de dados SQL. A Figura 3.1 apresenta o Diagrama de Entidade-Relação (ERD) final do sistema.

As entidades centrais do modelo são:

- **User, Project, e ProjectAssignment:** que em conjunto gerem os utilizadores e as suas permissões de acesso aos diferentes projetos.
- **ChatRoom e ChatMessage:** que estruturam os dados a serem anotados.
- **Annotation:** a tabela principal que armazena a ligação entre uma mensagem, um anotador e um *thread*, sendo a base para toda a análise posterior.

A utilização do SQLAlchemy e do Alembic para migrações de base de dados permitiu uma evolução estruturada do esquema ao longo do desenvolvimento do projeto.

3.2.3 API RESTful

A comunicação entre o frontend e o backend é realizada através de uma API RESTful. A API expõe um conjunto de endpoints para todas as operações necessárias, desde a autenticação até ao cálculo de métricas. A API está documentada seguindo o standard OpenAPI, o que facilita a sua exploração e integração.

Os endpoints estão logicamente agrupados por responsabilidade:

- **Auth:** Gestão de autenticação e tokens.
- **Projects:** Operações de utilizador normal sobre projetos e salas de chat.
- **Annotations:** Criação e gestão de anotações.
- **Admin:** Operações de administração, incluindo importação/exportação e cálculo de IAA.

3.3 Protótipos de Interface

3.3.1 Mapa de Navegação

A estrutura de navegação da plataforma está idealizada para adaptar-se aos diferentes perfis de utilizador. O sistema prevê um portal de login onde após autenticação, os utilizadores terão acesso a um dashboard principal, que funcionará como ponto central de acesso às diversas funcionalidades da plataforma. A estrutura completa do mapa de navegação pode ser visualizada na Figura 3.2, que foi modificada para ocupar uma página inteira em landscape.

A ferramenta de anotação será estruturada de forma modular, onde o dashboard principal permitirá acesso aos vários módulos disponíveis. Numa primeira fase, o módulo de disentanglement será o unico modulo desta arquitetura modular, no entanto o objetico da ferramenta de anotação será de acomodar mais módulos no futuro.

3.3.2 Principais Interfaces

Portal de Entrada

O acesso à plataforma é controlado através de um portal de autenticação minimalista. Esta decisão de design visa garantir a integridade dos dados e a atribuição correta das tarefas de anotação, sendo o login obrigatório para qualquer interação com os dados do sistema.

Dashboard Principal

Após autenticação, o utilizador acede a um dashboard que apresenta uma visão geral da plataforma. Este componente central adapta-se dinamicamente ao perfil do utilizador (administrador ou anotador), apresentando as funcionalidades relevantes e o estado atual das tarefas atribuídas.

Módulo de Disentanglement

O primeiro módulo implementado na plataforma foca-se na tarefa de disentanglement de chat, apresentando duas visões distintas:

Visão do Anotador Para os anotadores, a interface apresenta:

- Lista das chatrooms atribuídas automaticamente pelo sistema
- Interface de anotação com visualização sequencial das mensagens
- Sistema intuitivo de tagging para classificação de threads
- Indicadores de progresso da tarefa
- Mecanismos de validação em tempo real

Visão do Administrador Os administradores têm acesso a funcionalidades adicionais:

- Gestão completa dos datasets
- Monitorização do progresso dos anotadores
- Visualização de métricas e estatísticas
- Configuração da distribuição automática de tarefas

Interface de Anotação

O componente central do módulo de disentanglement é a interface de anotação, que foi projetada para maximizar a eficiência do processo de anotação. Cada chatroom é apresentada como uma sequência temporal de mensagens, onde o anotador pode facilmente:

- Visualizar o contexto completo da conversa
- Criar e atribuir tags de thread às mensagens
- Acompanhar o progresso da anotação em tempo real
- Navegar eficientemente entre diferentes chatrooms

O sistema mantém um salvamento automático do progresso, permitindo que os anotadores retomem seu trabalho de forma seamless em qualquer momento.

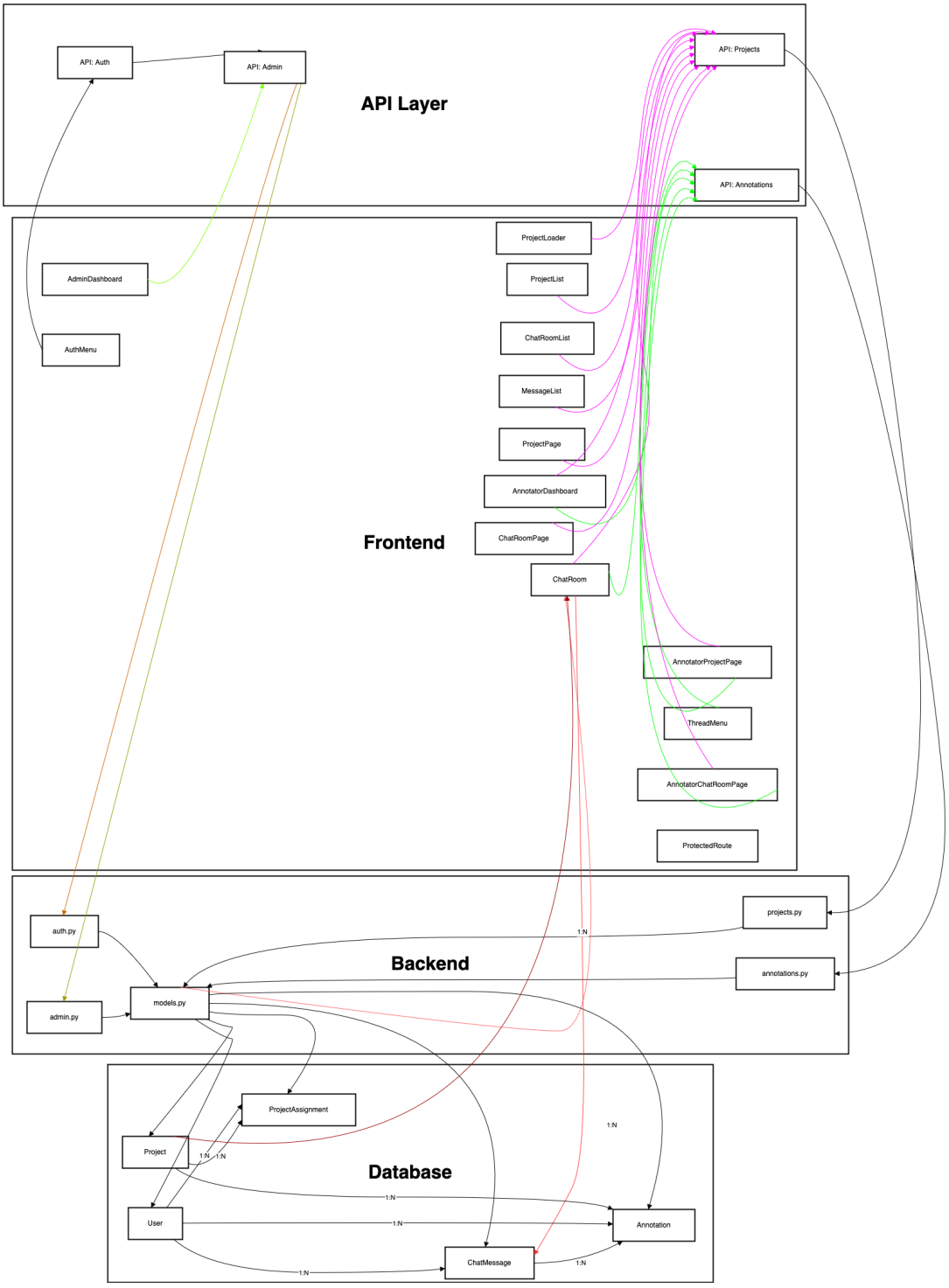

Figura 3.1: Modelo de Entidade-Relação final do Sistema.

ID	Descrição	Estado	Notas
	Requisitos Funcionais		
RF1	Suporte à importação de dados (CSV, JSON)	Cumprido	Implementada importação de mensagens via CSV e anotações via JSON.
RF2	Organização de dados por projetos	Cumprido	Estrutura central da aplicação.
RF3	Exportação dos dados anotados (JSON)	Cumprido	Funcionalidade de exportação por sala de chat implementada.
RF4	Interface de utilizador clara e funcional	Cumprido	Interface React desenvolvida com foco na usabilidade para anotação.
RF5	Suporte a múltiplos anotadores por projeto	Cumprido	Sistema de atribuição de utilizadores a projetos.
RF6	Gestão de atribuição de tarefas	Cumprido	Administradores podem atribuir/remover utilizadores de projetos.
RF7	Interface especializada para chat disentanglement	Cumprido	Ecrã de anotação dedicado com gestão de threads.
RF8	Sistema de tagging para classificação em threads	Cumprido	Funcionalidade nuclear da anotação.
RF9	Visualização sequencial das mensagens	Cumprido	A sala de chat apresenta as mensagens por ordem.
RF10	Armazenamento para cálculo de métricas	Cumprido	O modelo de dados permite o cálculo de IAA.
RF11	Autenticação de utilizadores	Cumprido	Implementado com JWT (access e refresh tokens).
RF12	Definição de roles (admin/anotador)	Cumprido	O modelo

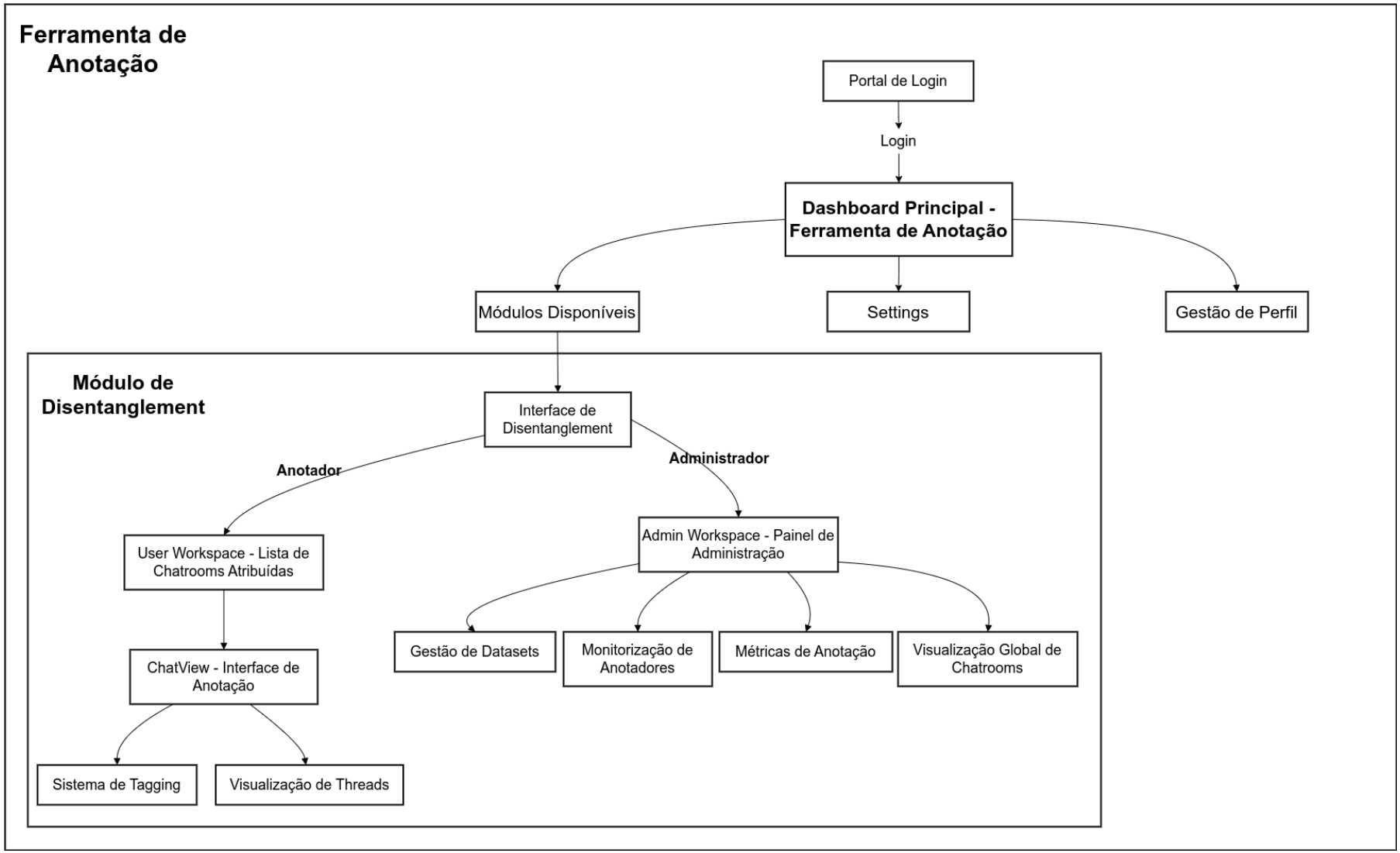

Figura 3.2: Mapa de Navegação do Sistema.

4 - Solução Proposta

4.1 Apresentação

A solução materializa-se numa plataforma web para anotação de dados de chat, com foco inicial no processo de disentanglement. O desenvolvimento baseou-se num protótipo funcional Lopes 2024 que permitiu validar conceitos base e estabelecer fundações para evolução futura.

Protótipo Inicial O protótipo, desenvolvido entre outubro e novembro de 2024, implementou funcionalidades essenciais:

- Interface base de visualização de chat
- Sistema de gestão de tags para anotação
- Processamento de ficheiros CSV com formato específico
- Gestão básica de workspace para ficheiros

Requisitos de Dados O sistema processa ficheiros CSV com estrutura específica, incluindo:

- user_id: identificador do utilizador
- turn_id: identificador único da mensagem
- turn_text: conteúdo da mensagem
- reply_to_turn: referência explícita a outra mensagem

Insights do Protótipo O desenvolvimento exploratório inicial proporcionou descobertas importantes:

- Validação da interface de anotação para disentanglement
- Confirmação da viabilidade do processamento de ficheiros CSV
- Identificação de pontos de optimização no fluxo de trabalho
- Feedback directo dos utilizadores sobre funcionalidades essenciais

4.2 Arquitectura

A arquitectura da solução, representada na Figura 4.1, incorpora elementos de diferentes abordagens analisadas no benchmarking. Esta decisão reflecte-se em dois princípios base:

Operacionalização O primeiro foca-se na operacionalização através de:

- Deployment via containers para simplicidade operacional
- Configuração reduzida para facilitar manutenção
- Monitorização integrada de componentes
- Backup e recuperação de dados simplificados

Modularidade O segundo centra-se na modularidade através de:

- Interfaces bem definidas entre componentes
- Separação clara de responsabilidades
- Sistema de plugins para extensões
- Gestão independente de dados por módulo

Evolução Planeada A arquitectura suporta evolução em várias dimensões:

- Adição de novos módulos de anotação
- Integração com ferramentas de análise
- Expansão das capacidades de processamento
- Adaptação a diferentes tipos de dados

4.3 Tecnologias e Ferramentas Utilizadas

A implementação assenta em tecnologias web estabelecidas, escolhidas conforme estado da arte e conforme os requisitos do projeto. O frontend utiliza React para a interface de anotação, enquanto o backend em Python, desenvolvido com FastAPI, gere o acesso aos dados e a lógica aplicacional.

Frontend A escolha do React como framework principal mantém-se desde o protótipo inicial, fundamentada por:

- Gestão eficiente de estado através de Hooks e Context API.
- Componentes reutilizáveis para consistência da interface.
- Rendering optimizado para operações interativas na anotação.
- Extensa documentação e comunidade ativa, facilitando o desenvolvimento.

Backend O backend da aplicação foi desenvolvido em Python (3.11) utilizando a framework **FastAPI**. Esta escolha foi fundamentada pela performance, facilidade de criação de APIs REST e suporte assíncrono, permitindo uma comunicação eficiente com o frontend React.

Arquitectura do Backend O backend segue uma arquitectura em camadas padrão:

- **API Layer**: Implementação dos endpoints REST utilizando FastAPI, responsáveis por receber pedidos do frontend e devolver respostas.
- **Service Layer** (ou Lógica de Negócio): Onde reside a lógica principal da aplicação, orquestrando operações como gestão de projetos, utilizadores e o processo de anotação.
- **Data Layer**: Camada de acesso a dados, utilizando **SQLAlchemy** para interagir com a base de dados **SQLite**, abstraindo as operações de persistência.

Processamento de Dados No contexto atual, focado no *chat disentanglement*, o backend implementa o seguinte processamento:

- Parsing e validação de ficheiros CSV na importação.
- Estruturação e armazenamento das conversas e mensagens na base de dados num formato adequado para a anotação.

Sistema de Persistência A persistência de dados é garantida pela base de dados **SQLite**, interagindo através do ORM **SQLAlchemy**. Esta abordagem permite:

- Gestão eficiente dos dados relacionais (utilizadores, projetos, mensagens, anotações).
- Definição clara do esquema da base de dados através de modelos Python.
- Abstração das queries SQL, facilitando o desenvolvimento e manutenção.

4.4 Ambientes de Teste e de Produção

O desenvolvimento segue uma abordagem iterativa com dois ambientes distintos:

Ambiente de Desenvolvimento Suporta o desenvolvimento activo e testes:

- Configuração simplificada para desenvolvimento local
- Dados de teste para validação de funcionalidades
- Ferramentas de debugging e monitorização
- Automatização de testes unitários e de integração

Ambiente de Produção Garante estabilidade e performance:

- Configuração optimizada para performance
- Backup automático de dados
- Monitorização de métricas operacionais
- Gestão de logs e diagnóstico

4.5 Abrangência

A solução abrange inicialmente o processo de disentanglement de conversas em chatrooms, estabelecendo bases para expansão futura. A arquitectura modular permite adicionar novos tipos de anotação sem alterações estruturais significativas.

Módulo de Disentanglement O primeiro módulo implementado foca-se em:

- Interface especializada para visualização de chatrooms
- Ferramentas para identificação e separação de conversas
- Sistema de validação de anotações
- Métricas de progresso e qualidade

Extensibilidade A arquitectura suporta expansão através de:

- Interfaces bem definidas para novos módulos
- Gestão independente de dados por módulo
- Documentação para desenvolvimento de extensões

4.6 Componentes

4.6.1 Frontend

O frontend da aplicação assenta em React 18, escolha fundamentada pela experiência bem-sucedida do protótipo inicial e pelas capacidades da framework para desenvolvimento de interfaces complexas. A implementação aproveita características modernas do React, como Hooks e componentes funcionais, permitindo uma gestão de estado eficiente e código mais manutenível.

A arquitectura do frontend organiza-se em componentes modulares, cada um com responsabilidades bem definidas:

Sistema de Componentes O desenvolvimento segue uma abordagem baseada em componentes reutilizáveis, organizados em três níveis:

- **Componentes Base**: Elementos UI fundamentais como botões, inputs e cards, implementados com styled-components para consistência visual
- **Componentes Compostos**: Agregações de componentes base que implementam funcionalidades específicas, como o visualizador de mensagens ou o sistema de tagging
- **Páginas**: Composições completas que integram múltiplos componentes para criar interfaces funcionais

Gestão de Estado A gestão de estado utiliza uma combinação de React Hooks nativos e contextos, evitando a complexidade adicional de soluções como Redux. Esta decisão baseou-se na experiência do protótipo, onde se verificou que:

- O estado local com useState é suficiente para a maioria dos componentes
- useContext permite compartilhar estado eficientemente entre componentes relacionados
- useReducer oferece gestão de estado mais complexa quando necessário

Interface de Anotação O componente central da aplicação - a interface de anotação - foi redesenhado com base no feedback do protótipo. Implementa:

- Visualização em split-view das mensagens e threads identificadas
- Sistema de drag-and-drop para classificação de mensagens
- Preview em tempo real das alterações
- Atalhos de teclado para operações frequentes

4.6.2 Backend

O backend da aplicação será desenvolvido em Python, escolha fundamentada pela necessidade de integração com ferramentas de processamento de dados e análise. A implementação utilizará uma web framework Python moderna (Flask ou FastAPI), permitindo o desenvolvimento de uma API REST robusta e eficiente.

Arquitectura do Backend O backend segue uma arquitectura em camadas:

- **API Layer**: Implementação de endpoints REST utilizando uma web framework Python (Flask ou FastAPI)
- **Service Layer**: Lógica de negócio e processamento de dados
- **Data Layer**: Gestão de persistência e acesso a dados

Processamento de Dados O sistema implementa um pipeline de processamento específico para chatrooms:

- Parsing e validação de ficheiros CSV
- Estruturação de conversas em formato adequado para anotação
- Cálculo de métricas e estatísticas
- Cache de resultados frequentes

Sistema de Persistência A persistência de dados será implementada utilizando uma base de dados SQL leve como SQLite, através de um ORM para Python. Esta abordagem permite:

- Gestão eficiente de dados estruturados
- Queries optimizadas para diferentes volumes de dados
- Backup e versionamento de dados

4.6.3 Fluxo de Dados

O fluxo de dados na plataforma foi desenhado para reflectir as necessidades práticas do processo de anotação:

Importação Os administradores podem carregar ficheiros CSV contendo conversas de chatrooms. O processo é deliberadamente simplificado, permitindo:

- Upload directo de ficheiros
- Validação automática do formato
- Feedback imediato sobre a qualidade dos dados

Processamento O backend processa os ficheiros, preparando-os para anotação:

- Validação estrutural dos dados
- Organização das mensagens em sequência temporal
- Preparação para disentanglement

Distribuição As conversas são disponibilizadas através da interface web:

- Atribuição automática aos anotadores
- Tracking de progresso
- Sistema de validação em tempo real

4.7 Interfaces

A comunicação entre componentes realiza-se através de uma API REST, permitindo independência entre frontend e backend. Esta abordagem facilita:

- Desenvolvimento paralelo de componentes
- Manutenção e evolução independente
- Integração com ferramentas externas
- Testes isolados de funcionalidades

O sistema de dados organiza-se por módulos, onde cada tipo de anotação mantém seu próprio modelo de dados. Para o disentanglement, isto inclui:

- Estrutura de dados optimizada para chatrooms
- Sistema de versionamento de anotações
- Métricas específicas para avaliação
- Export em formatos standard

O diagrama apresentado na Figura 4.1 detalha os principais componentes da solução, incluindo o frontend em React, backend em Python, sistema de persistência de dados e as interfaces de comunicação entre os diferentes módulos. A arquitectura modular permite a extensão futura com novos tipos de anotação além do módulo inicial de disentanglement.

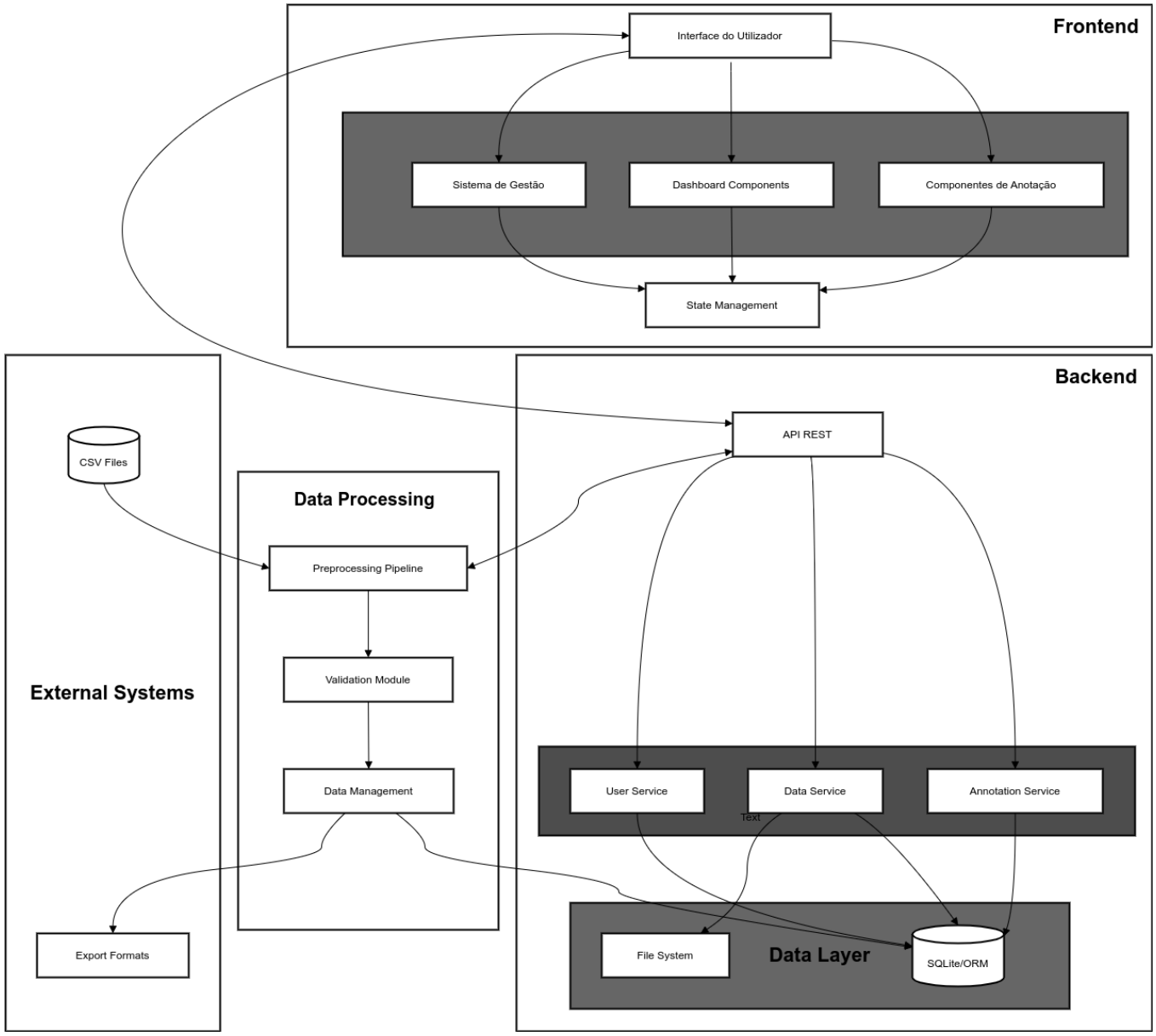

Figura 4.1: Diagrama de Arquitectura do Sistema.

5 - Testes e Validação

5.1 Introdução

Este capítulo descreve como a ferramenta de anotação será testada e validada. O objetivo é verificar se a ferramenta funciona corretamente, se é fácil de usar para a tarefa de *chat disentanglement*, e se representa uma melhoria face a métodos de anotação mais manuais e sem suporte para a tarefa. A validação pretende confirmar a aplicabilidade prática da solução.

5.2 Plano de Testes

O plano de testes foca-se na experiência prática de utilizadores idealmente com diferentes perfis, avaliando tanto a funcionalidade como a usabilidade da ferramenta.

5.2.1 Participantes e Perfis de Teste

Os testes planeados envolverão dois grupos principais de utilizadores:

- **Administradores:** Um pequeno grupo de utilizadores familiarizados com a gestão de sistemas ou projetos, que testará as funcionalidades administrativas da ferramenta.
- **Anotadores:** Um grupo mais alargado de participantes (e.g., estudantes, investigadores) com interesse ou necessidade na análise de chats, que se focará na tarefa central de anotação para *chat disentanglement*.

Será obtido consentimento informado dos participantes e os dados recolhidos serão tratados de forma anónima, garantindo a confidencialidade.

5.2.2 Cenários e Tarefas Principais

As tarefas de teste procurarão simular o ciclo de vida da utilização da ferramenta:

- **Tarefas de Administração:** Executadas pelo grupo de administradores, incluirão:
  - Criação e gestão de projetos de anotação.
  - Gestão de utilizadores (criação, atribuição de roles e permissões).
  - Carregamento de dados (ficheiros de *chat*) para os projetos.
  - Verificação da correta importação e organização dos dados.
- **Tarefas de Anotação:** Executadas pelo grupo de anotadores, incluirão:
  - Navegação e visualização das conversas carregadas.
  - Utilização da interface para identificar e marcar diferentes *threads*.
  - Atribuição de mensagens às *threads* correspondentes.
  - Edição e correção de anotações realizadas.
  - Exportação das anotações finalizadas.

Um dos focos da avaliação será perceber se estas tarefas, especialmente as de anotação, são percebidas como mais eficientes ou intuitivas quando realizadas com a ferramenta, comparativamente a abordagens manuais.

5.2.3 Ambiente de Teste

Prevê-se que os testes decorram num ambiente controlado, como um laboratório, onde os participantes utilizarão a ferramenta em computadores disponibilizados. Esta abordagem simplifica a observação e o suporte durante os testes, focando na interação direta com a aplicação.

5.3 Recolha e Análise de Dados

A avaliação da ferramenta basear-se-á na recolha de dados sobre a execução das tarefas e na perceção dos utilizadores:

- **Eficiência e Dificuldades:** Serão registados os tempos aproximados para completar tarefas chave e, através de observação e questionamento direto, identificar-se-ão os principais obstáculos ou pontos de fricção na utilização da ferramenta.
- **Feedback de Usabilidade Detalhado:** Para além da observação, a opinião dos utilizadores será recolhida através de questionários pós-teste e, potencialmente, breves entrevistas. Estes instrumentos incluirão:
  - **Escalas de Avaliação:** Perguntas utilizando escalas tipo Likert (e.g., de 1 a 5) para quantificar a perceção sobre aspetos específicos, como: ”Quão fácil foi identificar as diferentes threads?”, ”Quão clara considerou a interface de anotação?”, ”Quão satisfeito ficou com a ferramenta para esta tarefa?”.
  - **Perguntas Abertas:** Questões para recolher feedback qualitativo e sugestões, tais como: ”Qual foi a maior dificuldade que sentiu ao usar a ferramenta?”, ”Que aspetos da ferramenta mais o(a) ajudaram na tarefa de anotação?”, ”Tem sugestões para melhorar a ferramenta?”, ”Considera que esta ferramenta torna o processo de disentanglement mais fácil ou rápido do que métodos manuais que conheça?”.

  Estes métodos têm como objetico capturar a satisfação geral, identificar pontos fortes e fracos específicos da interface e do fluxo de trabalho, e perceber se a ferramenta cumpre a promessa de facilitar a tarefa de anotação. Os questionários exatos serão definidos posteriormente, mas focar-se-ão nestes eixos de avaliação.
- **Avaliação da Consistência da Anotação:** Um dos objetivos da ferramenta é promover a consistência entre múltiplos anotadores. Planeia-se incluir na versão final da ferramenta mecanismos para calcular métricas de concordância inter-anotador (IAA - *Inter-Annotator Agreement*). Caso esta funcionalidade esteja operacional durante a fase de testes, será utilizada para avaliar quantitativamente a consistência das anotações produzidas por diferentes utilizadores nos mesmos dados. Se a implementação destas métricas automáticas não estiver concluída a tempo, a avaliação focará numa análise qualitativa, observando se a interface e o fluxo de trabalho da ferramenta parecem facilitar uma maior consistência em comparação com abordagens manuais.

A análise dos dados recolhidos procurará identificar padrões de eficiência, principais temas no feedback de usabilidade (pontos fortes e áreas a melhorar) e indicações sobre o potencial da ferramenta para melhorar a qualidade e consistência do processo de anotação. Os resultados destinam-se a validar a abordagem da solução e a informar o desenvolvimento futuro.

Embora reconhecendo potenciais limitações relacionadas com o número de participantes ou a artificialidade do ambiente de teste, espera-se que esta validação forneça indicações claras sobre a pertinência e eficácia da ferramenta desenvolvida.

6 - Método e Planeamento

6.1 Metodologia de Desenvolvimento

O desenvolvimento deste projeto segue uma abordagem iterativa e incremental, alinhada com metodologias adaptadas ao contexto académico e às necessidades específicas do AISIC LAB. Esta escolha fundamenta-se na necessidade de validação contínua com stakeholders e na natureza evolutiva dos requisitos de uma plataforma modular de anotação.

6.1.1 Princípios Metodológicos

A metodologia adotada assenta em três princípios fundamentais:

- **Iterações Curtas**: Ciclos de desenvolvimento de duas semanas, permitindo feedback regular e ajustes frequentes
- **Validação Contínua**: Envolvimento regular dos stakeholders do AISIC LAB para validação de funcionalidades
- **Desenvolvimento Incremental**: Construção progressiva da plataforma, começando pelo módulo de disentanglement

6.1.2 Organização do Trabalho

O desenvolvimento está estruturado em sprints quinzenais, com os seguintes elementos:

- **Planeamento**: Definição de objetivos e tarefas no início de cada sprint
- **Desenvolvimento**: Implementação das funcionalidades priorizadas
- **Revisão**: Avaliação do progresso e demonstração aos stakeholders
- **Retrospetiva**: Análise do processo e identificação de melhorias

6.2 Planeamento e Cronograma

O planeamento do projeto, representado na Figura 6.1, está organizado em fases distintas que refletem a evolução da plataforma desde o protótipo inicial até à solução final.

6.2.1 Fases do Projeto

Fase Inicial (Outubro - Dezembro 2024)

Esta fase focou-se na validação de conceitos e estabelecimento de fundações:

- Desenvolvimento do protótipo
- Validação da interface de anotação
- Levantamento tecnológico
- Documentação inicial

MVP - Módulo Disentanglement (Dezembro 2024 - Janeiro 2025)

Consolidação do protótipo existente:

- Refinamento da interface frontend
- Testes de usabilidade
- Implementação de feedback inicial
- Validação com utilizadores piloto

Infraestrutura Base (Janeiro - Março 2025)

Estabelecimento da arquitetura modular:

- Setup do ambiente de desenvolvimento
- Migração do backend para Python
- Implementação da arquitetura modular
- Sistema de autenticação

Plataforma Core (Março - Maio 2025)

Desenvolvimento das funcionalidades principais:

- Framework para múltiplos módulos
- Sistema de gestão de datasets
- API base para integração
- Interface de administração

Finalização (Maio - Junho 2025)

Preparação para disponibilização:

- Testes extensivos
- Validação com utilizadores
- Documentação técnica
- Deployment em produção

6.3 Análise Crítica ao Planeamento

A presente secção visa analisar o progresso do projeto face ao planeamento inicial, identificando os principais marcos alcançados, os desafios encontrados e as adaptações realizadas durante o desenvolvimento subsequente à primeira entrega.

6.3.1 Progresso Realizado

Desde a validação inicial do conceito através do protótipo, o desenvolvimento focou-se na construção da infraestrutura base e do módulo principal de *chat disentanglement*. Os principais avanços incluem:

- Implementação do *backend* utilizando a framework FastAPI em Python, estabelecendo uma API REST para comunicação com o *frontend*.
- Configuração da persistência de dados com uma base de dados SQLite, gerida através do ORM SQLAlchemy.
- Desenvolvimento e integração do sistema de autenticação e gestão de utilizadores com papéis (administrador/anotador).
- Refinamento e desenvolvimento contínuo do *frontend* em React, implementando as interfaces necessárias para a gestão de projetos e a anotação de *disentanglement*.
- Implementação da funcionalidade de importação de dados a partir de ficheiros CSV.

Este progresso permitiu obter uma versão funcional da ferramenta centrada na sua tarefa principal, conforme pode ser visualizado no cronograma atualizado (Figura 6.1).

6.3.2 Desafios Encontrados e Adaptações ao Plano

Durante o desenvolvimento, alguns desafios e constrangimentos levaram a ajustes no plano e na abordagem inicial:

- **Priorização vs. Generalidade:** A visão inicial de uma plataforma altamente modular e genérica, capaz de suportar diversos tipos de anotação de forma extensível, revelou-se demasiado ambiciosa para o âmbito temporal e os recursos disponíveis no TFC. Para garantir a entrega de uma solução funcional e útil para o caso de uso principal do AISIC Lab, foi tomada a decisão consciente de **priorizar o desenvolvimento do módulo de *chat disentanglement***. Isto implicou que a implementação do *backend* e do modelo de dados fosse mais específica para esta tarefa, adiando a introdução das abstrações necessárias para uma generalização mais ampla a outros tipos de anotação.
- **Foco no Formato CSV:** Pelas mesmas razões de priorização e alinhamento com as necessidades imediatas do caso de uso, o suporte à importação de dados foi, nesta fase, limitado ao formato CSV. O suporte a outros formatos (JSON, TXT, etc.) permanece como um objetivo para evolução futura.
- **Complexidade Técnica:** A integração entre o frontend e o backend, a gestão do estado da aplicação e a implementação correta das interações de anotação apresentaram desafios técnicos que exigiram tempo e esforço de desenvolvimento significativo.
- **Balanceamento de Funcionalidades:** Foi necessário balancear o desenvolvimento de novas funcionalidades com a necessidade de refinar e estabilizar as funcionalidades existentes, respondendo também a feedback e requisitos que emergiram durante o processo.

Estas adaptações, embora desviando parcialmente da visão mais abrangente inicial, foram consideradas necessárias para garantir a viabilidade da entrega de um produto funcional e relevante dentro do contexto do TFC. A modularidade e a extensibilidade continuam a ser princípios orientadores para o futuro da plataforma.

6.3.3 Implicações no Planeamento Futuro

As decisões de priorização refletem-se no estado atual da ferramenta e no planeamento subsequente. O foco continuará a ser a consolidação e teste do módulo de *disentanglement*. A introdução de maior generalidade no backend e o suporte a outros formatos de dados são agora considerados trabalhos futuros, a serem abordados após a validação e estabilização da funcionalidade principal. A documentação técnica e os testes de usabilidade (Capítulo 5) ganham maior ênfase para garantir a qualidade da entrega atual.

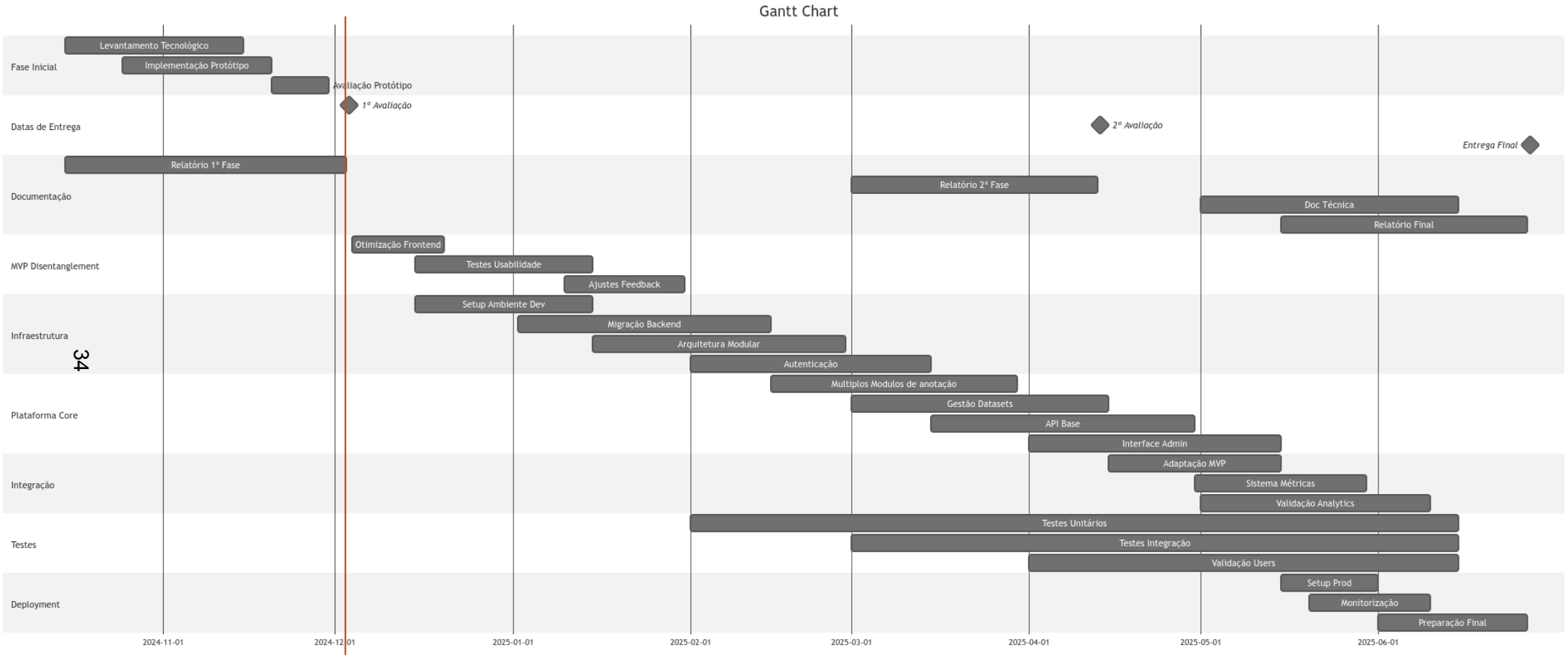

Figura 6.1: Cronograma detalhado do projeto (Gantt Chart) - Estado Atualizado

Referências Bibliográficas

About BRAT (2024). URL: `https://brat.nlplab.org/about.html` (acedido em 11/2024).

BRAT Rapid Annotation Tool (2024). URL: `https://github.com/nlplab/brat/tree/master` (acedido em 11/2024).

Doccano (2024). URL: `https://github.com/doccano/doccano/` (acedido em 11/2024).

Doccano Developer Guide (2024). URL: `https://doccano.github.io/doccano/developer_guide/` (acedido em 11/2024).

Elsner, M. e Charniak, E. (2010). "Disentangling Chat". Em: *Computational Linguistics*.

Lopes, F. (2024). Annotation tool for chat disentanglement. URL: `https://github.com/fabiofalopes/annotation_ui` (acedido em 11/2024).

Matos-Carvalho, J. P. (2024). The Lusófona LaTeX Template User's Manual. Lusófona University. URL: `https://github.com/jpmcarvalho/UL-Thesis`.