UNIVERSIDADE
LUSÓFONA

Ferramenta de Anotação

Trabalho Final de Curso

Relatório Final - 3ª Entrega

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

Lista de Figuras

Lista de Tabelas

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.

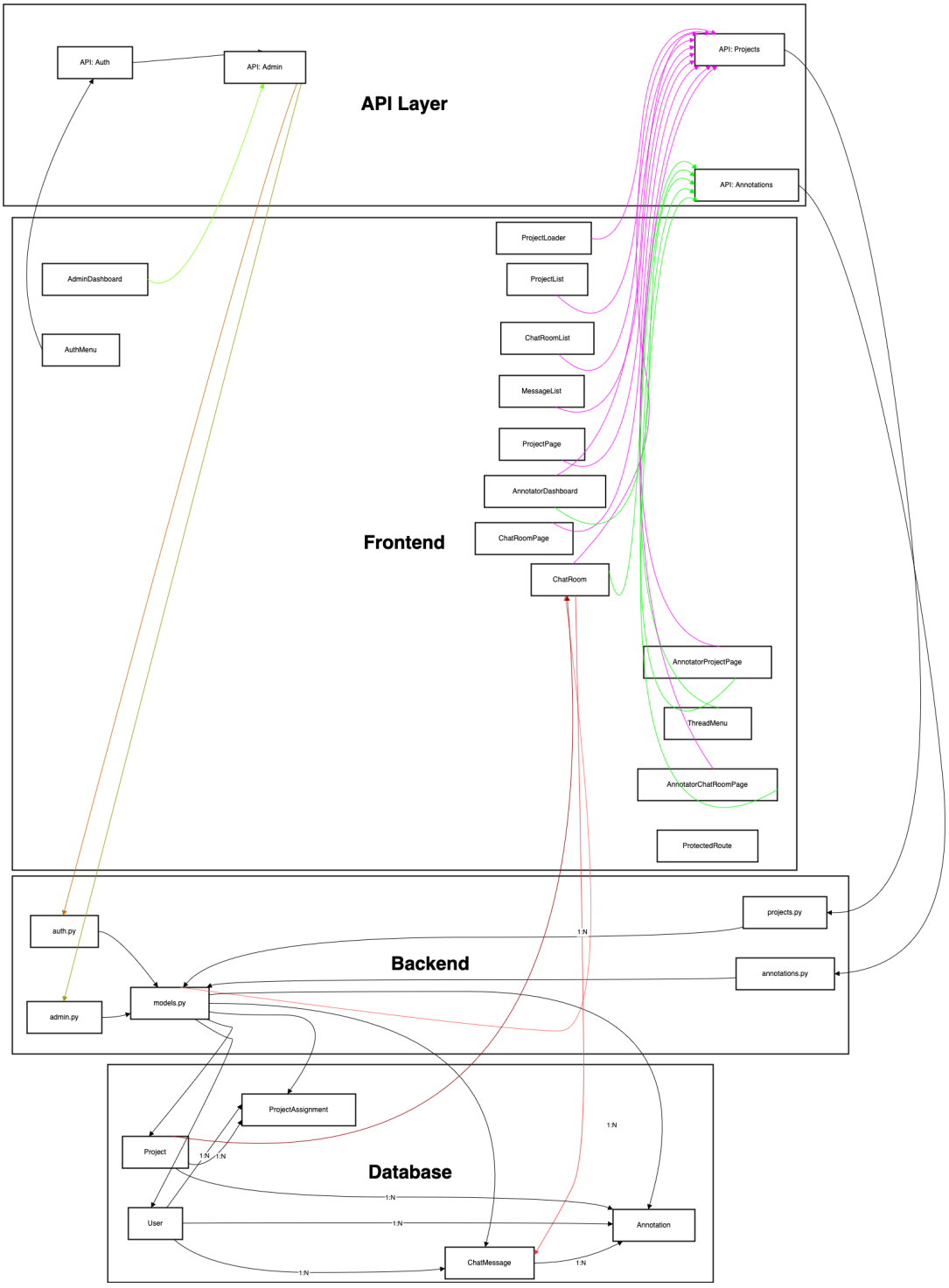

Figura 3.1: Modelo de Entidade-Relação final do Sistema.

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

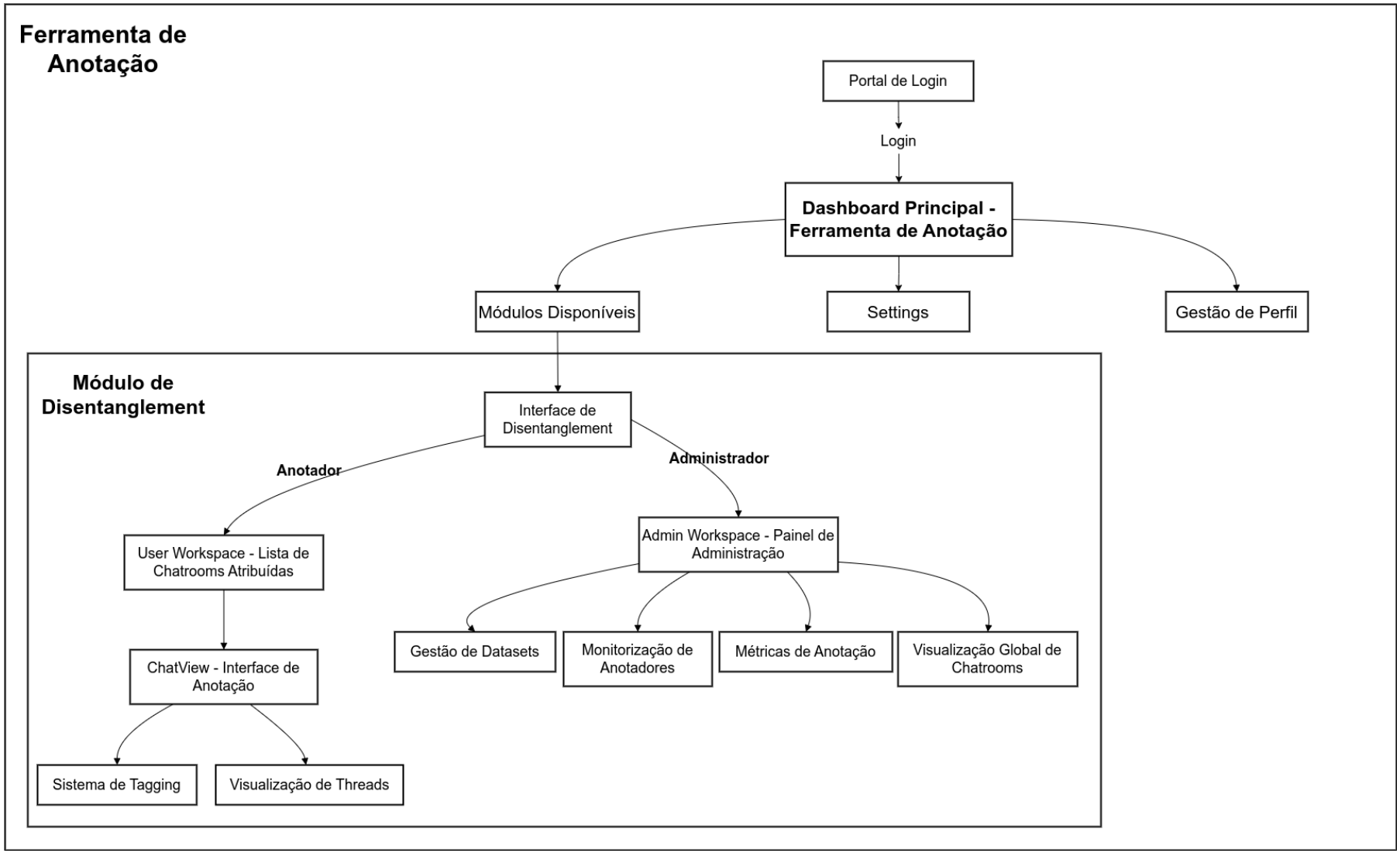

Figura 3.2: Mapa de Navegação do Sistema.

4 - Solução Proposta

Este capítulo detalha a solução técnica implementada para responder aos requisitos especificados. A apresentação abrange a arquitetura global do sistema, as tecnologias e ferramentas selecionadas para o seu desenvolvimento, e uma descrição dos principais componentes do frontend e do backend.

4.1 Arquitetura da Solução

A plataforma foi desenhada seguindo uma **arquitetura cliente-servidor desacoplada**, uma abordagem moderna que promove a separação de responsabilidades, a escalabilidade e a manutenibilidade. A solução é composta por dois sistemas independentes que comunicam através de uma API bem definida.

- **Frontend (Cliente):** Uma *Single-Page Application* (SPA) desenvolvida com a biblioteca **React**. A sua única responsabilidade é renderizar a interface do utilizador (UI) e gerir o estado da interação do utilizador. Toda a lógica de negócio, processamento de dados e autenticação são delegados ao backend através de chamadas a uma API RESTful. Esta abordagem de ”cliente puro”garante que o frontend permanece focado na experiência do utilizador.
- **Backend (Servidor):** Um servidor de API RESTful desenvolvido com a framework **FastAPI** em Python. Este componente é o cérebro da aplicação, responsável por:
  - Implementar toda a lógica de negócio.
  - Gerir a autenticação e autorização de utilizadores.
  - Executar todas as operações de base de dados (CRUD - Create, Read, Update, Delete).
  - Realizar os cálculos computacionalmente intensivos, como o Inter-Annotator Agreement (IAA).

A comunicação entre os dois componentes é feita exclusivamente através de requisições HTTP, com os dados a serem trocados no formato JSON.

4.2 Tecnologias e Ferramentas Utilizadas

A seleção de tecnologias foi guiada por critérios de performance, maturidade, ecossistema e adequação aos requisitos do projeto. A Tabela 4.1 resume as escolhas feitas.

4.3 Componentes da Solução

4.3.1 Frontend

O frontend foi estruturado para ser modular e de fácil manutenção, seguindo as melhores práticas do React.

- **Componentes de Página:** Componentes de alto nível que representam uma vista completa (e.g., `AdminDashboard.js`, `AnnotatorChatRoomPage.js`). São estes componentes que contêm a lógica de *data-fetching*, chamando a API para obter os dados necessários.

Componente	Tecnologia/Ferramenta	Justificação
Frontend	React 18	Framework líder para SPAs, com um vasto ecossistema e gestão de estado eficiente através de Hooks e Context API.
	JavaScript (ES6+)	Linguagem padrão para desenvolvimento web.
	Axios	Cliente HTTP para realizar as chamadas à API RESTful de forma fiável e com gestão de intercetores.
	CSS3	Estilização dos componentes para uma interface limpa e funcional.
Backend	Python 3.11	Linguagem robusta com um forte ecossistema para ciência de dados e desenvolvimento web.
	FastAPI	Framework web de alta performance com validação de dados automática e geração de documentação OpenAPI.
	SQLAlchemy	O principal ORM para Python, permitindo uma interação segura e abstrata com a base de dados.
	Alembic	Ferramenta para a gestão de migrações de esquema da base de dados.
	Pydantic	Biblioteca para validação de dados, utilizada pelo FastAPI para garantir a integridade dos dados.
	SciPy	Biblioteca para computação científica, utilizada para o cálculo do IAA (Algoritmo Húngaro).
	python-jose, passlib	Bibliotecas para a gestão de JWTs e hashing de passwords, garantindo a segurança.
Base de Dados	PostgreSQL	SGBD relacional robusto e pronto para produção.
	SQLite	SGBD leve e baseado em ficheiro, ideal para desenvolvimento e testes locais.
DevOps	Docker	Plataforma de contentorização para criar ambientes de desenvolvimento e produção consistentes.

Tabela 4.1: Tecnologias e Ferramentas Utilizadas na Solução

- **Componentes de UI:** Componentes mais pequenos e reutilizáveis (e.g., `MessageBubble.js`, `ProjectCard.js`) que são puramente presentacionais e recebem dados via *props*.
- **Gestão de API (`/src/utils/api.js`):** Módulo centralizador que exporta todas as funções de comunicação com o backend. Utiliza o `axios` e implementa intercetores para gerir automaticamente a injeção e o refrescamento de tokens de autenticação (JWT).
- **Gestão de Estado (`/src/contexts`):** Para o estado global, como a informação do utilizador autenticado, foi utilizado o Context API do React. O `AuthContext` disponibiliza o estado de autenticação a qualquer componente que necessite, enquanto o hook `useState` é usado para o estado local.

4.3.2 Backend

O backend está organizado por funcionalidades, seguindo as melhores práticas do FastAPI.

- **Routers da API (`/app/api`):** Os endpoints estão divididos em ficheiros modulares (`admin.py`, `projects.py`, etc.), cada um contendo um `APIRouter`, o que mantém o código organizado por domínio.
- **Lógica de Negócio e Acesso a Dados (`/app/crud.py`):** Este ficheiro contém toda a lógica que interage com a base de dados, executando as queries (via SQLAlchemy), realizando cálculos (como o IAA) e implementando as regras de negócio.
- **Modelos da Base de Dados (`/app/models.py`):** Define o esquema da base de dados através de classes Python que herdam do `Base` do SQLAlchemy.
- **Esquemas da API (`/app/schemas.py`):** Contém os modelos Pydantic que definem a ”forma”dos dados que entram e saem da API, garantindo a validação automática das requisições e a serialização das respostas.
- **Autenticação e Dependências (`/app/dependencies.py`):** Código de suporte para a segurança da API, implementando a lógica de validação de tokens JWT e criando dependências reutilizáveis para obter o utilizador atual ou verificar permissões.

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.

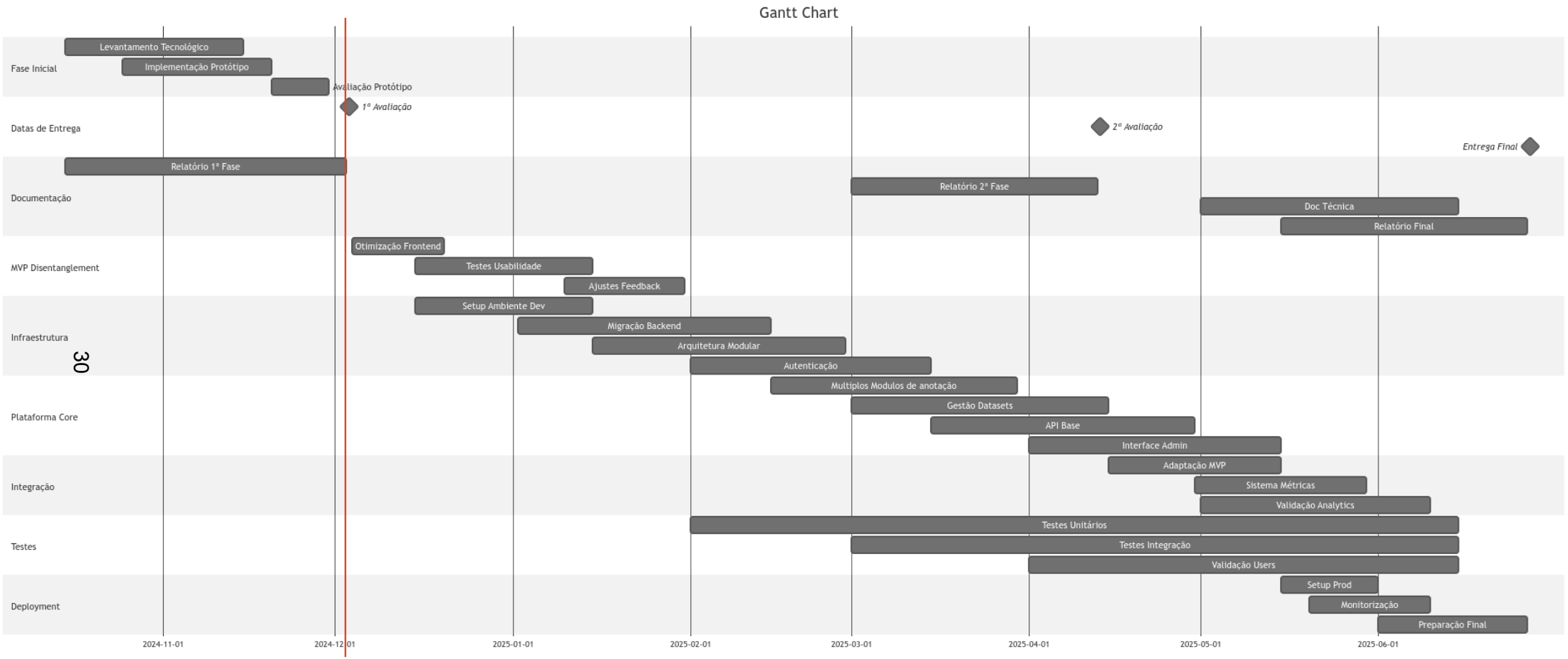

Figura 6.1: Cronograma detalhado do projeto (Gantt Chart) - Estado Atualizado

7 - Resultados

Este capítulo apresenta os resultados concretos obtidos com o desenvolvimento da ferramenta de anotação. A apresentação está dividida em três áreas principais: uma visão geral da plataforma final, uma análise detalhada das funcionalidades chave implementadas, com especial destaque para o cálculo automático de métricas de concordância, e a documentação técnica produzida.

7.1 Apresentação da Plataforma

A solução desenvolvida é uma aplicação web completa que serve como um ambiente integrado para a anotação de chatrooms, especificamente para a tarefa de *chat disentanglement*. A plataforma foi desenhada para servir dois perfis de utilizador distintos: o **Anotador**, focado na tarefa de anotação, e o **Administrador**, responsável pela gestão de projetos, utilizadores e pela análise dos dados.

O fluxo de utilização da aplicação segue um percurso lógico:

1. **Autenticação:** O utilizador acede através de uma página de Login.
2. **Dashboard:** Após o login, é apresentado um dashboard adaptado ao seu perfil.
    - **Anotador:** Vê uma lista dos projetos a que está atribuído (`AnnotatorDashboard`). Ao selecionar um projeto, vê as salas de chat disponíveis (`AnnotatorProjectPage`).
    - **Administrador:** Vê uma visão geral de todos os projetos e utilizadores no sistema (`AdminDashboard`).
3. **Anotação:** O anotador seleciona uma sala de chat e entra na interface de anotação (`AnnotatorChatRoomPage`), onde pode visualizar as mensagens e atribuir-lhes *thread IDs*.
4. **Análise (Admin):** O administrador pode aceder a uma página de projeto (`AdminProjectPage`) para gerir atribuições, importar dados e, crucialmente, visualizar a análise de concordância entre anotadores (`AdminChatRoomView`).

7.2 Resultados da Implementação

Esta secção detalha as funcionalidades mais relevantes que foram implementadas, demonstrando o cumprimento dos requisitos definidos e respondendo ao feedback recebido pelo júri.

7.2.1 Cálculo de Inter-Annotator Agreement (IAA)

Uma das funcionalidades centrais da plataforma é o cálculo automático da métrica de concordância entre anotadores (IAA), um requisito explícito do júri. A plataforma implementa o algoritmo **"1-to-1 agreement"**, que avalia a concordância estrutural entre os *threads* criados por diferentes anotadores, focando-se no conteúdo das conversas e não nos nomes arbitrários dos *threads*.

O processo de cálculo, disponível na visão do administrador para cada sala de chat, é o seguinte:

1. **Agregação de Anotações:** Para uma dada sala de chat, o sistema primeiro agrupa todas as anotações por cada anotador. O resultado é um conjunto de ”documentos de anotação”, um para cada utilizador, onde cada documento contém os *threads* que esse utilizador definiu.

2. **Cálculo Par a Par (Pairwise):** A concordância é então calculada para cada par de anotadores possível. Por exemplo, numa sala com três anotadores (A, B, C), o sistema calcula o IAA para (A, B), (A, C), e (B, C).

3. **Média Global:** O valor final de IAA apresentado para a sala de chat é a média aritmética de todos os scores de concordância calculados entre os pares.

O núcleo do algoritmo (`_calculate_one_to_one_accuracy` em `crud.py`) resolve um problema de atribuição. Para cada par de anotadores, ele constrói uma matriz de custo onde a dissimilaridade entre um *thread* do Anotador 1 e um *thread* do Anotador 2 é calculada com base no **Índice de Jaccard**. O índice mede a sobreposição de mensagens entre os dois *threads*:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

A dissimilaridade na matriz é, portanto, $1 - J(A, B)$. Com esta matriz, o **Algoritmo Húngaro** (via `scipy.optimize.linear_sum_assignment`) encontra a correspondência ótima ”um-para-um”entre os *threads* que maximiza a semelhança total.

O resultado é uma matriz de similaridade, apresentada na UI do administrador, e um score de concordância final.

7.2.2 Gestão de Projetos e Utilizadores

A plataforma fornece uma interface de administração robusta que permite a gestão completa do ciclo de vida de um projeto de anotação. As funcionalidades implementadas, acessíveis apenas ao administrador, incluem:

- **Gestão de Projetos:** Criação, listagem e remoção de projetos.
- **Gestão de Utilizadores:** Criação, listagem e remoção de utilizadores.
- **Atribuição a Projetos:** Atribuição granular de utilizadores a projetos específicos, o que garante o isolamento dos dados e a correta distribuição de tarefas.

7.2.3 Importação e Exportação de Dados

Para facilitar a integração com outros fluxos de trabalho, foram desenvolvidas funcionalidades de importação e exportação de dados:

- **Importação de Mensagens:** Os administradores podem iniciar um projeto importando uma chatroom completo a partir de um ficheiro CSV.
- **Importação de Anotações:** O sistema suporta a importação em lote de anotações a partir de um ficheiro JSON, que pode conter anotações de múltiplos utilizadores.
- **Exportação de Projetos:** Todos os dados de uma sala de chat (mensagens e a totalidade das anotações) podem ser exportados para um único ficheiro JSON.

7.3 Documentação Técnica da API

Uma vantagem inerente à escolha da framework FastAPI é a geração automática de uma especificação da API que segue o standard **OpenAPI 3.0**. Este processo resulta num ficheiro `openapi.json` que descreve todos os endpoints da aplicação, os seus parâmetros e os formatos de dados esperados.

O principal objetivo prático deste ficheiro no contexto do projeto foi facilitar os testes e a validação do backend. Ao importar esta especificação para ferramentas de desenvolvimento como o Postman, foi possível testar cada endpoint de forma sistemática e eficiente durante o ciclo de desenvolvimento, garantindo o seu correto funcionamento.

8 - Conclusão

8.1 Conclusão

O presente Trabalho Final de Curso teve como objetivo central o desenvolvimento de uma ferramenta de software especializada para a tarefa de anotação de *chat disentanglement*. O problema de base identificado foi a ausência de plataformas dedicadas que não só facilitassem o processo de anotação manual, mas que também integrassem mecanismos de análise de qualidade e concordância, forçando frequentemente os investigadores a recorrer a ferramentas genéricas e a processos de cálculo de métricas desligados da tarefa principal.

Em resposta a este desafio, foi concebida, desenvolvida e implementada uma aplicação web completa. A solução, construída sobre uma arquitetura moderna cliente-servidor (React e FastAPI), materializa-se numa plataforma funcional que permite a gestão de projetos de anotação, a atribuição de tarefas a múltiplos anotadores e, mais importante, oferece uma interface otimizada para a tarefa de *chat disentanglement*.

O principal resultado deste trabalho é uma ferramenta que cumpre os seus requisitos fundamentais. Destaca-se a implementação do cálculo automático do Inter-Annotator Agreement (IAA) através do algoritmo ”1-to-1 agreement”, que utiliza o Índice de Jaccard e o Algoritmo Húngaro para fornecer uma medida robusta da concordância estrutural entre anotadores. Esta funcionalidade, que foi uma indicação explícita dos coordenadores do TFC, transforma a plataforma de uma simples ferramenta de anotação num ambiente de análise, permitindo aos gestores de projeto aferir a qualidade e a consistência das anotações diretamente no sistema.

No entanto, é com rigor académico que se reconhecem as limitações do trabalho realizado. A principal limitação reside na ausência de uma fase de validação formal com utilizadores finais. Embora a ferramenta seja funcional e tecnicamente robusta, não foi conduzido um estudo empírico para avaliar o seu impacto real na qualidade das anotações ou na experiência do anotador. Adicionalmente, alguns requisitos não-funcionais, como a implementação de backups automáticos e testes de carga formais, foram considerados fora do âmbito da fase de desenvolvimento atual.

8.2 Trabalhos Futuros

As limitações identificadas abrem um caminho claro para trabalhos futuros, que poderiam elevar significativamente o impacto e a maturidade do projeto:

- **Estudo de Validação com Utilizadores:** O passo mais crítico seria a realização de um estudo formal. Este estudo envolveria um grupo de anotadores que realizaria a mesma tarefa de anotação utilizando a nossa ferramenta e um método de base (e.g., folha de cálculo), analisando métricas objetivas (tempo, scores de IAA) e subjetivas (inquéritos de satisfação e usabilidade).
- **Expansão de Métricas de Análise:** A plataforma poderia ser enriquecida com o cálculo de outras métricas de concordância, como o Krippendorff's Alpha, que é mais flexível em cenários com múltiplos anotadores e dados em falta.
- **Generalização da Ferramenta:** A arquitetura foi pensada de forma modular. Um próximo passo seria abstrair o processo de anotação para que a plataforma pu-

desse ser configurada para outros tipos de tarefas (e.g., anotação de entidades, análise de sentimento).

- **Melhorias de Infraestrutura e Performance:** Implementar as funcionalidades de backup e realizar testes de carga para otimizar o desempenho da API e da base de dados para um número elevado de utilizadores concorrentes, incluindo a migração para um driver de base de dados assíncrono.

Em suma, este projeto entregou com sucesso uma solução aplicacional concreta para um problema real no domínio do processamento de linguagem natural, estabelecendo uma base sólida sobre a qual futuras investigações e desenvolvimentos podem ser construídos.

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*.

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