



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](http://www.ulusofona.pt)

## **Direitos de cópia**

Ferramenta de Anotação, Copyright de Fábio Lopes, Universidade Lusófona. A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona (UL) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Este documento foi gerado com o processador (pdf/Xe/Lua) $\text{\LaTeX}$  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

<b>Resumo</b>	<b>2</b>
<b>Abstract</b>	<b>3</b>
<b>Conteúdo</b>	<b>4</b>
<b>Lista de Figuras</b>	<b>6</b>
<b>Lista de Tabelas</b>	<b>7</b>
<b>1 Introdução</b>	<b>8</b>
1.1 Enquadramento e Motivação . . . . .	8
1.2 Objetivos . . . . .	9
1.3 Estrutura do Documento . . . . .	9
<b>2 Pertinência e Viabilidade</b>	<b>11</b>
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
<b>3 Especificação e Modelação</b>	<b>14</b>
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
<b>4 Solução Proposta</b>	<b>22</b>
4.1 Arquitetura da Solução . . . . .	22
4.2 Tecnologias e Ferramentas Utilizadas . . . . .	22
4.3 Fluxo de Trabalho Suportado . . . . .	22
4.4 Componentes da Solução . . . . .	24
4.4.1 Frontend . . . . .	24
4.4.2 Backend . . . . .	24
<b>5 Testes e Validação</b>	<b>25</b>
5.1 Introdução . . . . .	25
5.2 Abordagem e Justificação dos Testes . . . . .	25
5.2.1 Estratégia de Validação . . . . .	25
5.2.2 Critérios de Aceitação . . . . .	25

5.2.3	Participantes e Perfis de Avaliação . . . . .	26
5.3	Cenários e Tarefas de Validação . . . . .	26
5.4	Validação Técnica e de Recursos . . . . .	26
5.4.1	Validação da Infraestrutura Técnica . . . . .	26
5.4.2	Validação da Importação e Gestão de Dados . . . . .	27
5.4.3	Avaliação da Interface de Anotação . . . . .	27
5.4.4	Validação das Métricas de Concordância . . . . .	27
5.5	Resultados da Validação . . . . .	27
5.5.1	Cumprimento dos Critérios de Aceitação . . . . .	27
5.5.2	Validação de Recursos Técnicos . . . . .	28
5.5.3	Adequação ao Fluxo de Trabalho . . . . .	28
5.5.4	Limitações Identificadas . . . . .	28
5.6	Conclusões da Validação . . . . .	28
<b>6</b>	<b>Método e Planeamento</b>	<b>29</b>
6.1	Metodologia de Desenvolvimento . . . . .	29
6.1.1	Princípios Metodológicos . . . . .	29
6.1.2	Organização do Trabalho . . . . .	29
6.2	Planeamento e Cronograma . . . . .	29
6.2.1	Fases do Projeto . . . . .	29
6.3	Análise Crítica ao Planeamento . . . . .	30
6.3.1	Progresso Realizado . . . . .	31
6.3.2	Desafios Encontrados e Adaptações ao Plano . . . . .	31
6.3.3	Implicações no Planeamento Futuro . . . . .	32
<b>7</b>	<b>Resultados</b>	<b>34</b>
7.1	Apresentação da Ferramenta . . . . .	34
7.1.1	Fluxo de Trabalho Suportado . . . . .	34
7.1.2	Navegação e Interface . . . . .	34
7.2	Resultados da Implementação . . . . .	35
7.2.1	Cálculo de Inter-Annotator Agreement (IAA) . . . . .	35
7.2.2	Gestão de Projetos e Utilizadores . . . . .	36
7.2.3	Importação e Exportação de Dados . . . . .	36
7.3	Contribuição Principal: Fluxo de Trabalho Integrado . . . . .	36
7.4	Documentação Técnica da API . . . . .	37
<b>8</b>	<b>Conclusão</b>	<b>38</b>
8.1	Conclusão . . . . .	38
8.1.1	Pipeline de Anotação Implementado . . . . .	38
8.2	Trabalhos Futuros . . . . .	39
	<b>Referências Bibliográficas</b>	<b>40</b>

# Lista de Figuras

3.1	Modelo de Entidade-Relação final do Sistema. . . . .	17
3.2	Mapa de Navegação do Sistema. . . . .	18
3.3	Fluxograma de Navegação por Perfil. . . . .	19
6.1	Cronograma detalhado do projeto (Gantt Chart) - Estado Atualizado . . . .	33

# Lista de Tabelas

2.1	Comparação técnica detalhada das soluções utilizadas no contexto do AI-SIC Lab . . . . .	13
3.1	Tabela de Requisitos e Estado de Implementação. . . . .	20
4.1	Tecnologias e Ferramentas Utilizadas na Solução . . . . .	23



# 1 - Introdução

Considerando os atuais desafios da área de Processamento de Linguagem Natural (PLN) e acompanhando as tendências emergentes, surgiu, no contexto do AISIC Lab (Artificial Intelligence, Social Interaction and Complexity), a necessidade de uma ferramenta especializada para a anotação de dados. Este projeto visa responder aos desafios específicos encontrados no laboratório, nomeadamente na análise de interações em ambientes de chat com múltiplos participantes, uma tarefa conhecida como *chat disentanglement*.

## 1.1 Enquadramento e Motivação

A anotação de dados é um processo crítico no treino de modelos de *Machine Learning*, sendo indispensável para transformar dados não estruturados — como o texto de uma conversa — em dados estruturados e anotados. São estes dados que servem como “verdade fundamental” (*ground truth*), permitindo que os algoritmos de PLN aprendam a reconhecer padrões e a executar tarefas complexas com precisão.

Uma das tarefas mais desafiadoras neste domínio é o **chat disentanglement**: o processo de separar um diálogo, que pode conter múltiplas conversas entrelaçadas, nos seus *threads* (fios de conversa) constituintes. A motivação principal deste projeto emerge da ausência de ferramentas que sejam, simultaneamente, especializadas nesta tarefa e que integrem mecanismos de controlo de qualidade, como a análise da concordância entre anotadores (*Inter-Annotator Agreement* - IAA).

Uma das motivações 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 controlo de qualidade
- Requisitos específicos para diferentes domínios de aplicação

### 2. Limitações das Ferramentas Existentes:

- Ferramentas estabelecidas como o BRAT<sup>1</sup> apresentam limitações tecnológicas e falta de evolução
- Soluções atuais como o Doccano<sup>2</sup>, 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

---

<sup>1</sup>BRAT Rapid Annotation Tool 2024; About BRAT 2024

<sup>2</sup>Doccano 2024; Doccano Developer Guide 2024

- Necessidade de implementação de métricas e análises específicas
- 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 garante a flexibilidade para implementar workflows customizados de pré-processamento, anotação e análise posterior dos dados.

O problema central que este trabalho se propõe a resolver é, portanto, o seguinte: **\*\*como criar um ambiente de software integrado que não apenas facilite a tarefa de anotação de *chat disentanglement*, mas que também forneça aos gestores de projeto as ferramentas para gerir o processo e avaliar a qualidade das anotações produzidas?\***

## 1.2 Objetivos

Para responder ao problema identificado, foram definidos e alcançados os seguintes objetivos específicos para o projeto:

1. **Desenvolver uma Ferramenta Dedicada:** Construir uma aplicação web completa, com um *frontend* reativo e um *backend* robusto, especificamente desenhada para a tarefa de *chat disentanglement*.
2. **Implementar Funcionalidades de Gestão:** Dotar a plataforma de um painel de administração para a gestão de múltiplos projetos e utilizadores (anotadores e administradores), com controlo de acesso.
3. **Automatizar o Cálculo de Métricas de Qualidade:** Implementar o cálculo automático do *Inter-Annotator Agreement* (IAA) como uma funcionalidade central, permitindo a análise da consistência entre anotadores diretamente na plataforma.
4. **Garantir a Interoperabilidade dos Dados:** Assegurar que os dados (mensagens e anotações) possam ser facilmente importados e exportados em formatos padrão (CSV e JSON), facilitando a integração da ferramenta em fluxos de trabalho de investigação mais vastos.

A principal contribuição deste trabalho é a entrega de uma ferramenta de anotação *open-source*, funcional e especializada, que preenche a lacuna identificada e suporta um fluxo de trabalho de anotação completo e integrado.

## 1.3 Estrutura do Documento

Este documento está organizado da seguinte forma:

- **Capítulo 2 - Pertinência e Viabilidade:** Apresenta uma análise do estado da arte e compara a solução proposta com ferramentas existentes, justificando a sua relevância.
- **Capítulo 3 - Especificação e Modelação:** Detalha os requisitos funcionais e não-funcionais, os casos de uso e a modelação da arquitetura e da base de dados.
- **Capítulo 4 - Solução Proposta:** Descreve em detalhe a arquitetura final da aplicação, as tecnologias utilizadas e os seus principais componentes.

- **Capítulo 5 - Testes e Validação:** Apresenta a estratégia de verificação técnica utilizada para garantir a robustez e o correto funcionamento do sistema.
- **Capítulo 6 - Método e Planeamento:** Detalha a metodologia de desenvolvimento e realiza uma análise crítica do planeamento face à execução real do projeto.
- **Capítulo 7 - Resultados:** Apresenta os resultados concretos da implementação, incluindo as funcionalidades da plataforma, o fluxo de trabalho suportado e a documentação técnica gerada.
- **Capítulo 8 - Conclusão:** Sintetiza as contribuições do trabalho, discute as suas limitações e aponta direções para trabalhos futuros.

## 2 - Pertinência e Viabilidade

### 2.1 Pertinência

O desenvolvimento de uma ferramenta especializada 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 funcionalidades específicas para a tarefa de *chat disentanglement*, adaptando-se às necessidades identificadas no laboratório.

### 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 implementada 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 utilização em outros contextos académicos e promete uma redução significativa nos custos operacionais relacionados à anotação de dados.

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

### 2.3 Análise Comparativa com Soluções Existentes

A análise comparativa apresentada foca-se nas ferramentas e métodos previamente apontados e utilizados no laboratório AISIC para tarefas similares de anotação, não constituindo uma análise exaustiva de todas as soluções disponíveis no mercado.

#### 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 baseada em Django 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 tecnológica.

## **Excel**

Folhas de cálculo representam uma abordagem manual frequentemente utilizada para tarefas de anotação. Embora acessível e familiar para muitos utilizadores, este método apresenta limitações significativas em termos de consistência, controlo de qualidade e cálculo automático de métricas de concordância entre anotadores.

### **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**

Entre as diversas tarefas disponíveis em NLP, este trabalho concentra-se especificamente nas exigências de anotação para a tarefa de *chat disentanglement*, permitindo uma implementação direta e eficiente deste tipo de tarefa.

## **2.5 Identificação de Oportunidade de Negócio**

O projeto apresenta potencial para aplicação em contextos académicos e de investigação, especialmente em grupos que trabalham com processamento de linguagem natural e *Machine Learning*.

Solução	BRAT <sup>1</sup>	Doccano <sup>2</sup>	Excel	Solução Proposta
<i>Frontend</i>	jQuery 1.7.1 + jQuery UI, SVG para visualização, JavaScript vanilla	Nuxt.js <i>framework</i> , Vue.js, Modern JavaScript/TypeScript	Interface de folha de cálculo tradicional	React 18, TypeScript, Componentes funcionais, <i>Hooks</i> customizados
<i>Backend</i>	Python 2.5+, CGI/FastCGI, Bibliotecas JSON	Python 3.8+, Django 4.0+, REST API architecture	N/A (aplicação <i>desktop</i> )	Python 3.x, FastAPI, Arquitetura REST API
Formato de Anotação	Ficheiros .ann para anotações, Ficheiros .txt para texto	REST API <i>endpoints</i> , JSON para comunicação	Células e fórmulas manuais	REST API para comunicação; anotações via base de dados; suporte a importação/exportação CSV e JSON
Dados	Sistema de ficheiros, Estrutura em diretórios, Sem base de dados	Base de dados (Django ORM), Suporte a PostgreSQL	Ficheiros .xlsx/.csv locais	Base de dados relacional SQLite gerida via ORM; armazenamento estruturado
<i>Deployment</i>	Apache/Lighttpd com CGI, Servidor Python autónomo	<i>Docker containers</i> , <i>Docker Compose</i>	Instalação local por utilizador	<i>Docker containers</i> , Configuração simplificada
Extensibilidade	Sistema via ficheiros .conf, Arquitetura modular, <i>Plugins</i> jQuery	Arquitetura Django modular, REST API extensível	Limitada a macros e fórmulas	Componentes React modulares, API REST extensível (FastAPI)
Controlo de Qualidade	Manual, sem métricas automáticas	Limitado, sem métricas de IAA	Manual, propenso a erros humanos	Cálculo automático de <i>Inter-Annotator Agreement</i> (1-to-1)
Colaboração	Limitada, baseada em ficheiros	Suporte básico multi-utilizador	Partilha manual de ficheiros	Gestão de múltiplos anotadores, controlo de acesso por <i>roles</i>
Estado de Manutenção	Última atualização 2012, Projeto inativo	Projeto ativo, Atualizações regulares	Software comercial ativo	Em desenvolvimento ativo, <i>Stack</i> moderna
Especialização	Genérica para anotação de texto	Genérica para múltiplas tarefas de NLP	Não especializada	Especializada em <i>chat disentanglement</i>

Tabela 2.1: Comparação técnica detalhada das soluções utilizadas no contexto do AISIC Lab

<sup>a</sup>BRAT *Rapid Annotation Tool* 2024; *About BRAT* 2024

<sup>b</sup>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 análise de requisitos identificou os requisitos funcionais (RF) e não-funcionais (RNF) para o projeto, juntamente com o estado final da sua implementação. A tabela completa de requisitos e o seu estado de implementação encontra-se detalhada no final deste capítulo (ver Tabela 3.1 na página 20).

### 3.2 Modelação da Solução

A modelação da solução foi um processo iterativo que resultou numa arquitetura cliente-servidor, um modelo de dados relacional bem definido e uma API RESTful estruturada.

#### 3.2.1 Arquitetura da Solução

A plataforma segue uma arquitetura cliente-servidor 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 SQLite. 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 concebida para se adaptar 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.

A ferramenta de anotação poderá ser estruturada de forma a acomodar diferentes tipos de tarefas no futuro, onde o dashboard principal permitirá acesso às várias funcionalidades disponíveis. Numa primeira fase, a funcionalidade de *disentanglement* constitui o foco principal da implementação.

### 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.



## Funcionalidade de Disentanglement

A funcionalidade principal implementada 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 da funcionalidade de *disentanglement* é a interface de anotação, que foi concebida 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 uma gravação automática do progresso, permitindo que os anotadores retomem o seu trabalho de forma contínua em qualquer momento.

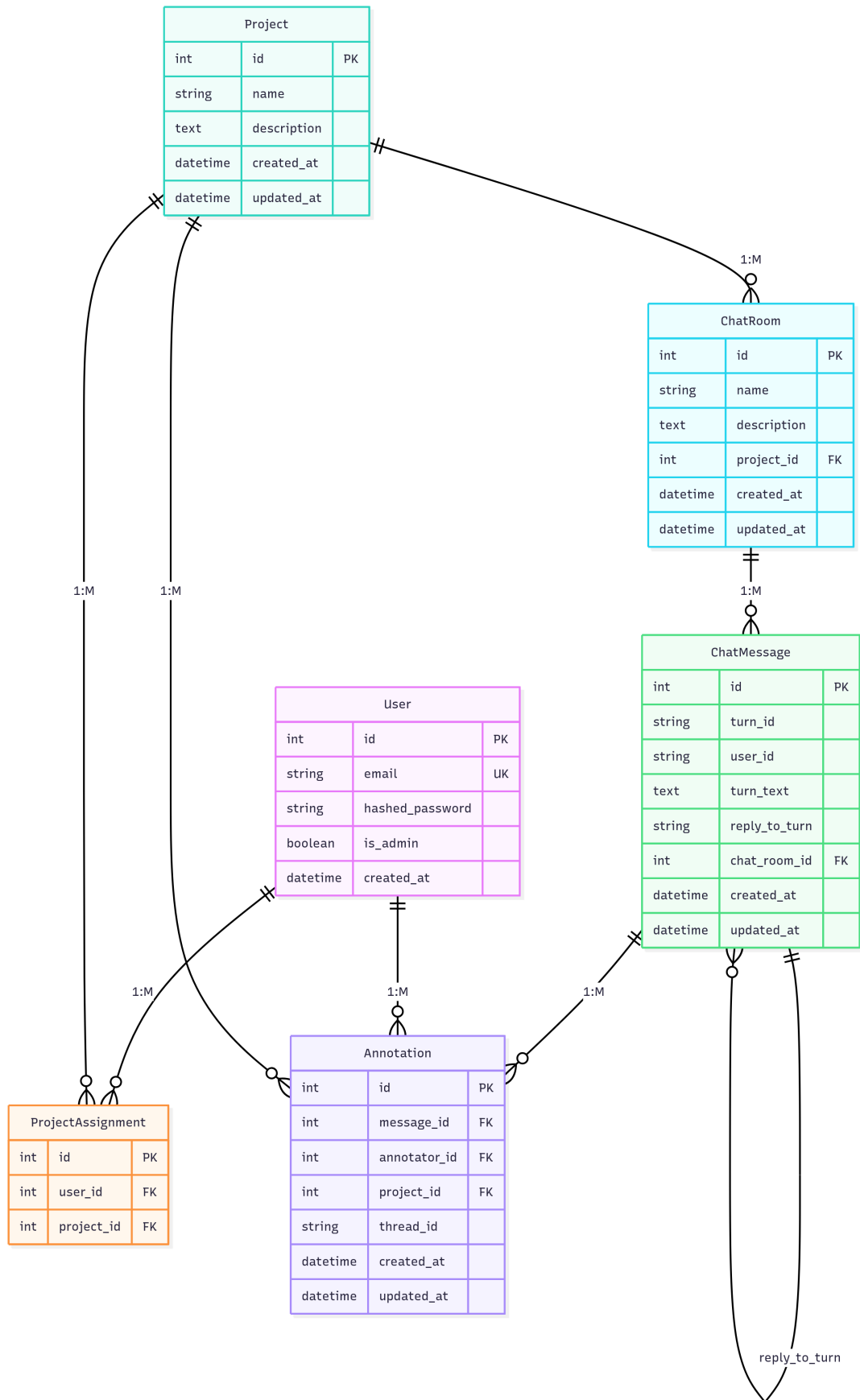


Figura 3.1: Modelo de Entidade-Relação final do Sistema.

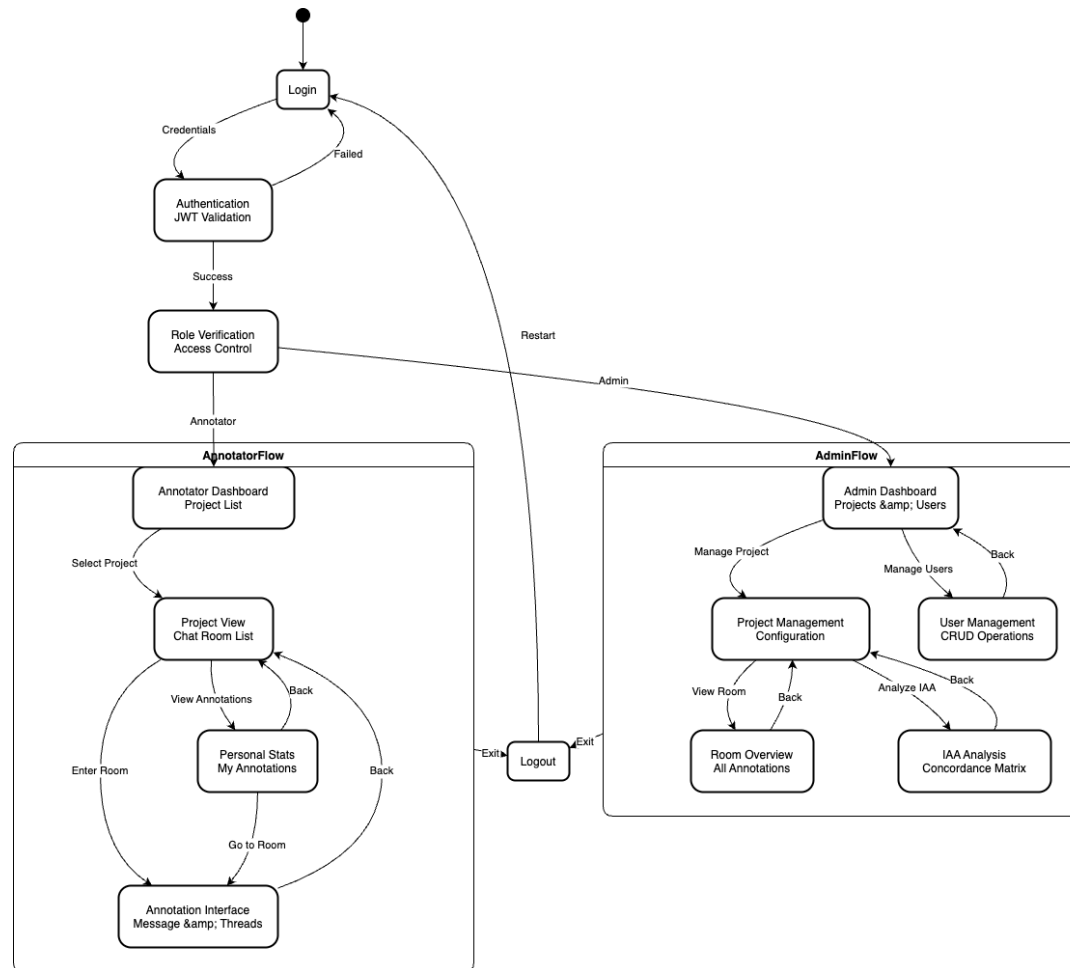


Figura 3.2: Mapa de Navegação do Sistema.

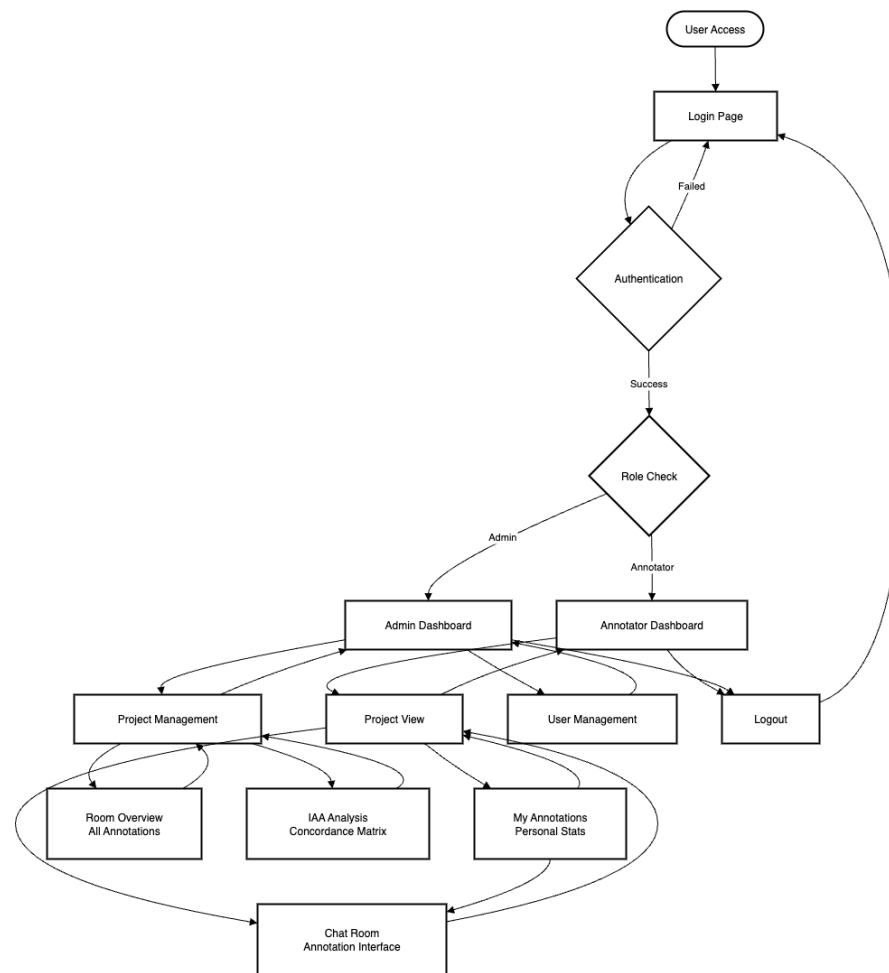


Figura 3.3: Fluxograma de Navegação por Perfil.

Tabela 3.1: Tabela de Requisitos e Estado de Implementação.

ID	Descrição	Estado	Notas
<b>Requisitos Funcionais</b>			
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 'User' contém o campo 'is_admin'.
RF13	Controlo de acesso baseado em permissões	Cumprido	Endpoints da API protegidos com base no role.
<b>Requisitos Não Funcionais</b>			
Continua na página seguinte...			

**Tabela 3.1 – continuação da página anterior**

<b>ID</b>	<b>Descrição</b>	<b>Estado</b>	<b>Notas</b>
RNF1	Tempo de resposta adequado	Cumprido	A API responde rapidamente a pedidos interativos.
RNF2	Processamento eficiente para múltiplos utilizadores	Parcialmente	A arquitetura suporta-o, mas não foram feitos testes de carga formais.
RNF3	Interface responsiva	Parcialmente	O design é funcional mas não foi totalmente otimizado para todos os tamanhos de ecrã.
RNF4	Feedback visual claro	Cumprido	A interface React fornece feedback sobre o estado (loading, error, success).
RNF5	Interface simples e intuitiva	Cumprido	O design focou-se na simplicidade para a tarefa de anotação.
RNF6	Backup automático de anotações	Não Cumprido	Não implementado. Requer uma solução de infraestrutura (e.g., cron jobs na base de dados).
RNF7	Logging de atividades críticas	Parcialmente	O backend utiliza logging básico, mas não um sistema de logging estruturado e exaustivo.

## 4 - Solução Proposta

Este capítulo detalha a solução técnica implementada para responder aos requisitos especificados. O conteúdo 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 concebida seguindo uma **arquitetura cliente-servidor desacoplada**, uma abordagem 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 responsabilidade é renderizar a interface do utilizador (UI) e gerir o estado da interação do utilizador. 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 responsável por:
  - Implementar a lógica de negócio.
  - Gerir a autenticação e autorização de utilizadores.
  - Executar 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 a troca de dados em 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 Fluxo de Trabalho Suportado

A solução implementa um fluxo de trabalho estruturado para o processo de anotação de *chat disentanglement*:

1. **Configuração do Projeto:** O administrador cria um projeto e define os anotadores que irão participar.
2. **Importação de Dados:** O administrador importa o corpus de mensagens (formato CSV) para o sistema.

Componente	Tecnologia/Ferramenta	Justificação
<b>Frontend</b>	React 18	Framework estabelecido 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 interceptores.
	CSS	Estilização dos componentes para uma interface limpa e funcional.
<b>Backend</b>	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.
<b>Base de Dados</b>	SQLite	SGBD relacional leve e baseado em ficheiro, adequado para o âmbito do projeto.
<b>DevOps</b>	Docker	Plataforma de contentorização para criar ambientes de desenvolvimento consistentes.

Tabela 4.1: Tecnologias e Ferramentas Utilizadas na Solução

3. **Atribuição de Tarefas:** O sistema organiza as mensagens em salas de chat e atribui-as aos anotadores.
4. **Processo de Anotação:** Os anotadores acedem às suas salas atribuídas e realizam a tarefa de classificação em *threads*.
5. **Cálculo de Métricas:** O administrador pode solicitar o cálculo automático do IAA para avaliar a concordância entre anotadores.
6. **Exportação de Resultados:** Os dados anotados e as métricas podem ser exportados para análise posterior.



## 4.4 Componentes da Solução

### 4.4.1 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`). Estes componentes contêm a lógica de *data-fetching*, chamando a API para obter os dados necessários.
- **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 as funções de comunicação com o backend. Utiliza o `axios` e implementa interceptores 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.

### 4.4.2 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 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 estrutura 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 foi testada e validada. O objetivo é verificar se a ferramenta funciona corretamente, se é adequada para a tarefa de *chat disentanglement*, e se representa uma melhoria face a métodos de anotação mais manuais. A validação pretende confirmar a aplicabilidade prática da solução desenvolvida e demonstrar o cumprimento dos objetivos propostos, nomeadamente o de contribuir para a solução de um problema real de investigação.

### 5.2 Abordagem e Justificação dos Testes

#### 5.2.1 Estratégia de Validação

A validação da ferramenta baseou-se numa abordagem de avaliação técnica interna realizada em conjunto com a equipa do laboratório AISIC. Esta estratégia foi escolhida por permitir:

- Uma análise técnica detalhada das funcionalidades implementadas.
- Avaliação por utilizadores com conhecimento especializado na área de processamento de linguagem natural.
- Validação em contexto real de investigação, com dados e cenários autênticos.
- Identificação de limitações e áreas de melhoria numa fase adequada do desenvolvimento.

#### 5.2.2 Critérios de Aceitação

Os testes foram estruturados para validar o cumprimento dos seguintes critérios de aceitação:

- **Funcionalidade Técnica:** A ferramenta deve importar, processar e exportar dados de conversas corretamente.
- **Adequação à Tarefa:** A interface deve facilitar a realização de tarefas de *chat disentanglement*.
- **Cálculo de Métricas:** O sistema deve calcular automaticamente métricas de concordância inter-anotador.
- **Gestão de Projetos:** O sistema deve permitir gestão adequada de utilizadores e projetos de anotação.
- **Eficiência Operacional:** A ferramenta deve representar uma melhoria face a métodos manuais.

### 5.2.3 Participantes e Perfis de Avaliação

A avaliação envolveu dois perfis principais de investigadores:

- **Administradores:** Investigadores familiarizados com a gestão de sistemas e projetos de investigação, que avaliaram as funcionalidades administrativas da ferramenta.
- **Anotadores:** Um grupo mais alargado de participantes (investigadores, estudantes) com experiência ou interesse na análise de dados textuais, que se focaram na tarefa central de anotação para *chat disentanglement*.

## 5.3 Cenários e Tarefas de Validação

As tarefas de validação procuraram simular o ciclo de vida completo da utilização da ferramenta:

- **Tarefas de Administração:** Avaliadas pelos investigadores com perfil administrativo, incluíram:
  - Criação e gestão de projetos de anotação.
  - Gestão de utilizadores (criação, atribuição de papéis e permissões).
  - Importação de dados (ficheiros de *chat*) para os projetos.
  - Verificação da correta importação e organização dos dados.
- **Tarefas de Anotação:** Avaliadas pelos investigadores com experiência em anotação, incluíram:
  - Navegação e visualização das conversas importadas.
  - 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.

## 5.4 Validação Técnica e de Recursos

### 5.4.1 Validação da Infraestrutura Técnica

Foi realizada uma validação detalhada dos recursos computacionais e técnicos da solução:

- **Armazenamento:** Validação da capacidade de armazenamento de dados na base SQLite, incluindo testes com conjuntos de dados de diferentes dimensões.
- **Desempenho:** Avaliação dos tempos de resposta para operações críticas (importação, cálculo de métricas, exportação).
- **Estabilidade:** Testes de funcionamento contínuo e robustez da aplicação durante sessões prolongadas de anotação.
- **Compatibilidade:** Verificação do funcionamento em diferentes sistemas operativos e navegadores web.

### 5.4.2 Validação da Importação e Gestão de Dados

Foi realizada uma validação técnica detalhada dos processos de importação de dados, verificando:

- Correta estruturação e organização dos dados na base de dados.
- Integridade dos dados durante o processo de importação.
- Funcionalidade de exportação dos dados anotados em formatos apropriados para análise posterior.

### 5.4.3 Avaliação da Interface de Anotação

A interface de anotação foi avaliada em reunião pela equipa do laboratório, focando:

- Adequação da visualização das conversas para a tarefa de disentanglement.
- Facilidade de identificação e marcação de diferentes *threads*.
- Eficiência do processo de atribuição de mensagens a *threads*.
- Capacidade de edição e correção de anotações.
- Clareza visual da organização das anotações realizadas.

### 5.4.4 Validação das Métricas de Concordância

Uma funcionalidade central da ferramenta é o cálculo automático de métricas de concordância inter-anotador (IAA - *Inter-Annotator Agreement*). Foi validado:

- Correto funcionamento do algoritmo de cálculo de concordância "one-to-one" Elsner e Charniak 2008.
- Adequação das métricas implementadas para a avaliação da qualidade das anotações.
- Capacidade de comparação entre múltiplos anotadores no mesmo conjunto de dados.
- Funcionalidade de exportação das métricas calculadas para análise externa.

## 5.5 Resultados da Validação

### 5.5.1 Cumprimento dos Critérios de Aceitação

A validação confirmou o cumprimento de todos os critérios de aceitação estabelecidos:

- **Funcionalidade Técnica:** Os processos de importação e exportação de dados funcionam de forma robusta.
- **Adequação à Tarefa:** A interface de anotação permite realizar eficientemente as tarefas de *chat disentanglement*.
- **Cálculo de Métricas:** As métricas de concordância são calculadas automaticamente e fornecem informação útil sobre a qualidade das anotações.
- **Gestão de Projetos:** O sistema de gestão de utilizadores e projetos adequa-se às necessidades de um ambiente de investigação.
- **Eficiência Operacional:** Confirmada melhoria significativa face a métodos manuais.

### 5.5.2 Validação de Recursos Técnicos

Os testes de infraestrutura confirmaram:

- Desempenho adequado para conjuntos de dados típicos de investigação.
- Compatibilidade com os principais navegadores web e sistemas operativos.
- Capacidade de armazenamento suficiente para múltiplos projetos simultâneos.

### 5.5.3 Adequação ao Fluxo de Trabalho

A avaliação em reunião pela equipa do laboratório indicou que as tarefas de anotação foram avaliadas como mais eficientes quando realizadas com a ferramenta, comparativamente a abordagens manuais ou anotações realizadas em outras ferramentas citadas neste trabalho. Especificamente:

- A visualização estruturada das conversas facilita a identificação de *threads*.
- O processo de atribuição de mensagens é mais intuitivo que métodos totalmente manuais.
- O cálculo automático de métricas de concordância elimina a necessidade de processamento manual posterior.
- A capacidade de exportação permite integração com fluxos de trabalho de análise existentes.

### 5.5.4 Limitações Identificadas

Durante a validação, foram também identificadas algumas limitações:

- A ferramenta é específica para tarefas de *chat disentanglement*, não sendo diretamente aplicável a outros tipos de anotação.
- A interface assume familiaridade básica com conceitos de anotação de dados textuais.

## 5.6 Conclusões da Validação

A validação realizada confirma que a ferramenta desenvolvida cumpre os objetivos propostos, fornecendo uma solução técnica adequada para tarefas de *chat disentanglement*. Todos os critérios de aceitação foram cumpridos, e a avaliação pela equipa do laboratório indica que a ferramenta representa uma melhoria significativa face a métodos manuais.

A validação técnica de recursos confirmou a adequação da infraestrutura para o contexto de investigação, com desempenho e estabilidade apropriados para as necessidades identificadas. A automatização do cálculo de métricas de concordância e a estruturação do processo de anotação constituem as principais contribuições para a melhoria do fluxo de trabalho.

Embora reconhecendo as limitações inerentes a uma validação técnica interna e a especificidade da ferramenta para *chat disentanglement*, os resultados fornecem indicações sobre a pertinência e eficácia da solução desenvolvida, confirmando a sua adequação ao contexto de investigação em processamento de linguagem natural e análise de conversas.

## 6 - Método e Planeamento

### 6.1 Metodologia de Desenvolvimento

O desenvolvimento deste projeto segue uma abordagem iterativa e incremental, adaptada 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 ferramenta especializada para anotação de *chat disentanglement*.

#### 6.1.1 Princípios Metodológicos

A metodologia adotada assenta em três princípios fundamentais:

- **Desenvolvimento Iterativo:** Ciclos de desenvolvimento flexíveis, permitindo adaptação às necessidades emergentes e feedback regular
- **Validação Contínua:** Envolvimento regular dos stakeholders do AISIC LAB para validação de funcionalidades
- **Desenvolvimento Incremental:** Construção progressiva da ferramenta, começando pelas funcionalidades essenciais de *disentanglement*

#### 6.1.2 Organização do Trabalho

O desenvolvimento está estruturado em fases de trabalho, com os seguintes elementos:

- **Planeamento:** Definição de objetivos e prioridades para cada fase de desenvolvimento
- **Desenvolvimento:** Implementação das funcionalidades priorizadas
- **Revisão:** Avaliação do progresso e demonstração aos stakeholders
- **Adaptação:** Análise do processo e ajustes necessários às circunstâncias

### 6.2 Planeamento e Cronograma

O planeamento do projeto, representado na Figura 6.1, está organizado em fases distintas que refletem a evolução da ferramenta 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 da ferramenta:

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

### **Ferramenta Principal (Março - Maio 2025)**

Desenvolvimento das funcionalidades principais:

- Funcionalidades de anotação especializadas
- Sistema de gestão de datasets
- API para comunicação frontend-backend
- Interface de administração

### **Finalização (Maio - Junho 2025)**

Preparação para disponibilização:

- Testes extensivos
- Validação com utilizadores
- Documentação técnica
- Preparação para deployment

## **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.

### 6.3.1 Progresso Realizado

Desde a validação inicial do conceito através do protótipo, o desenvolvimento focou-se na construção da infraestrutura base e das funcionalidades principais 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.
- Desenvolvimento do algoritmo de cálculo automático de métricas de concordância inter-anotador.

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, 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 especializado para *chat disentanglement*\*\***. Isto implicou que a implementação do *backend* e do modelo de dados fosse mais específica para esta tarefa, permitindo uma maior qualidade e adequação da solução final.
- **Foco em Formatos Essenciais:** Pelas mesmas razões de priorização e alinhamento com as necessidades imediatas do caso de uso, o suporte foi centrado nos formatos mais relevantes: importação através de CSV e exportação em JSON. O suporte a outros formatos de importação de dados 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. Este processo incluiu também a resposta a feedback e requisitos que emergiram durante o desenvolvimento.



- **Adaptação do Processo:** O processo de desenvolvimento teve de ser adaptado às circunstâncias académicas, com períodos de maior e menor intensidade de trabalho, ajustando-se às necessidades de outras disciplinas e compromissos académicos.

Estas adaptações, embora desviando 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 especialização para *chat disentanglement* permitiu uma maior qualidade e adequação da solução às necessidades específicas identificadas.

### 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 das funcionalidades 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) ganharam maior ênfase para garantir a qualidade da entrega atual.

O desenvolvimento demonstrou que uma abordagem pragmática, focada na entrega de valor específico para o domínio de *chat disentanglement*, é mais eficaz que uma tentativa de criar uma solução genérica sem validação adequada das necessidades específicas.

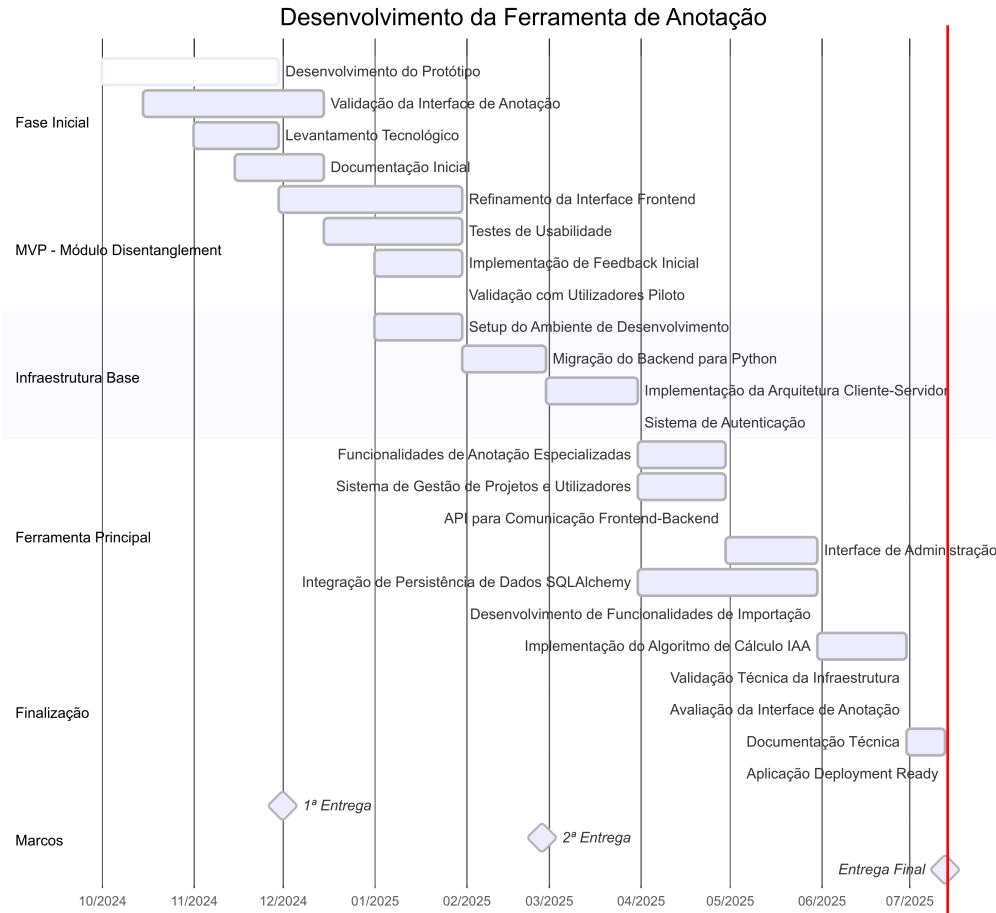


Figura 6.1: Cronograma detalhado do projeto (Gantt Chart) - Estado Atualizado

## 7 - Resultados

Este capítulo apresenta os resultados concretos obtidos com o desenvolvimento da ferramenta de anotação. O conteúdo está organizado em três áreas principais: uma visão geral da ferramenta 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 Ferramenta

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 ferramenta foi concebida 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.

#### 7.1.1 Fluxo de Trabalho Suportado

Uma das principais contribuições desta ferramenta é a definição e implementação de um fluxo de trabalho estruturado para anotação de *chat disentanglement*. Embora existam diversas ferramentas de anotação disponíveis, muitas com funcionalidades extensas, a proposta diferenciadora desta solução é a integração de um fluxo de trabalho completo e especializado.

O fluxo de trabalho implementado segue um percurso lógico em 6 etapas principais:

1. **Configuração de Projeto:** O administrador cria um novo projeto de anotação e define os utilizadores participantes.
2. **Importação de Dados:** Os dados de conversas são importados através de ficheiros CSV, estruturando-os na base de dados.
3. **Atribuição de Tarefas:** Os anotadores são atribuídos a projetos específicos, garantindo isolamento e distribuição adequada das tarefas.
4. **Processo de Anotação:** Os anotadores acedem às suas tarefas através de dashboards personalizados, selecionam salas de chat e realizam a anotação através de uma interface especializada.
5. **Análise de Concordância:** O sistema calcula automaticamente métricas de concordância inter-anotador (IAA), fornecendo feedback sobre a qualidade das anotações.
6. **Exportação de Resultados:** Os dados anotados são exportados em formato JSON para análise posterior ou integração com outros sistemas.

#### 7.1.2 Navegação e Interface

O percurso de utilização da aplicação implementa este fluxo de trabalho através das seguintes interfaces:

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 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 a respetiva anotação (*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, previamente, pelo júri.

### 7.2.1 Cálculo de Inter-Annotator Agreement (IAA)

Uma das funcionalidades centrais da ferramenta é o cálculo automático da métrica de concordância entre anotadores (IAA), um requisito explícito identificado durante o desenvolvimento. A ferramenta implementa o algoritmo **"1-to-1 agreement"** Elsner e Charniak 2008, que avalia a concordância estrutural entre os *threads* criados por diferentes anotadores, focando-se no índice de cada turno/mensagem anotada, e não nos labels das *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 agrupa 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 dos valores 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 ferramenta 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 completa 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:** Os dados de uma sala de chat (mensagens e anotações) podem ser exportados para um único ficheiro JSON.

## 7.3 Contribuição Principal: Fluxo de Trabalho Integrado

A principal contribuição desta ferramenta não reside apenas nas funcionalidades individuais implementadas, mas na integração dessas funcionalidades num fluxo de trabalho coeso e especializado para *chat disentanglement*. Esta abordagem distingue-se de ferramentas genéricas de anotação por oferecer:

- **Especialização:** Interface e funcionalidades otimizadas especificamente para a tarefa de disentanglement.
- **Automatização:** Cálculo automático de métricas de concordância, eliminando processamento manual posterior.
- **Gestão Integrada:** Desde a importação de dados até à exportação de resultados, o processo é gerido numa única ferramenta.
- **Controlo de Qualidade:** Métricas de concordância integradas no fluxo de trabalho para avaliação contínua da qualidade das anotações.

Esta integração permite que equipas de investigação tenham uma solução completa para projetos de anotação de *chat disentanglement*, reduzindo a necessidade de ferramentas múltiplas e processos manuais.

## 7.4 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 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*.

#### 8.1.1 Pipeline de Anotação Implementado

A ferramenta implementa uma arquitetura que suporta o ciclo de vida da anotação através de uma API REST e interfaces especializadas. O sistema integra as seguintes funcionalidades:

**Ingestão e Processamento de Dados:** A aplicação oferece múltiplas vias para importação de dados de conversas. Através de endpoints REST específicos (`/api/projects/{id}/import-messages`), a ferramenta processa ficheiros CSV contendo mensagens de chat. O sistema realiza validação, estruturação e persistência automática na base de dados SQLite. Esta funcionalidade suporta importação de grandes volumes de dados através de *scripting* ou integração programática. Isto permite que equipas de investigação automatizem a ingestão de *datasets* extensos sem intervenção manual.

**Interface de Anotação Especializada:** O frontend React implementa uma interface otimizada especificamente para *chat disentanglement*. As funcionalidades incluem visualização cronológica de mensagens, atribuição interativa de *thread IDs*, e gestão de estado em tempo real. A interface comunica com a API através de endpoints dedicados (`/api/annotations`) que garantem persistência imediata das anotações e sincronização entre múltiplos anotadores.

**Sistema de Análise Automática:** A ferramenta integra capacidades de análise quantitativa através da implementação do algoritmo "1-to-1 agreement" (Elsner e Charniak 2008). Este sistema processa automaticamente as anotações de múltiplos utilizadores, calculando métricas de concordância através de técnicas de otimização (Algoritmo Húngaro) e apresentando os resultados através de matrizes de similaridade e scores agregados. O cálculo é acionado dinamicamente através da API (`/api/admin/chatrooms/{id}/iaa`) e apresentado em tempo real na interface administrativa.

**Exportação e Integração:** A plataforma oferece capacidades de exportação estruturada através de endpoints REST que serializam dados de projetos completos em formato JSON. Esta funcionalidade permite integração com ferramentas de análise externa, facilitando a continuidade do trabalho de investigação e a interoperabilidade com outros sistemas de processamento de linguagem natural.

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" (Elsner e Charniak 2008), 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 pudesse 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).
- Doccoano (2024). URL: <https://github.com/doccoano/doccoano/> (acedido em 11/2024).
- Doccoano Developer Guide (2024). URL: [https://doccano.github.io/doccano/developer\\_guide/](https://doccano.github.io/doccano/developer_guide/) (acedido em 11/2024).
- Elsner, M. e Charniak, E. (2008). “You talking to me? a corpus and algorithm for conversation disentanglement”. Em: *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pp. 834–842.
- (2010). “Disentangling Chat”. Em: *Computational Linguistics*.
- Matos-Carvalho, J. P. (2024). The Lusófona L<sup>A</sup>T<sub>E</sub>X Template User’s Manual. Lusófona University. URL: <https://github.com/jpmcarvalho/UL-Thesis>.