



UNIVERSIDADE
LUSÓFONA

Ferramenta de Anotação

Trabalho Final de Curso

Relatório Intercalar 1º Semestre

Fábio Lopes
Bruno Saraiva
Zuil Filho

Trabalho Final de Curso | LEI | 2024/25

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) \LaTeX e o modelo ULThesis (v1.0.0) Matos-Carvalho 2024.

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 and 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.

Índice

Abstract	2
Índice	3
Lista de Figuras	5
Lista de Tabelas	6
1 Introdução	7
1.1 Enquadramento	7
1.2 Motivação e Identificação do Problema	7
1.3 Objetivos	8
1.4 Estrutura do Documento	8
2 Pertinência e Viabilidade	10
2.1 Pertinência	10
2.2 Viabilidade	10
2.3 Análise Comparativa com Soluções Existentes	10
2.3.1 Soluções Existentes	10
2.3.2 Análise de Benchmarking	11
2.4 Proposta de Inovação e Mais-Valias	11
2.5 Identificação de Oportunidade de Negócio	11
3 Especificação e Modelação	13
3.1 Análise de Requisitos	13
3.1.1 Requisitos Funcionais	13
3.1.2 Requisitos Não Funcionais	13
3.2 Modelação	14
3.2.1 Modelo de Dados	14
3.3 Protótipos de Interface	14
3.3.1 Mapa de Navegação	14
3.3.2 Principais Interfaces	15
4 Solução Proposta	19
4.1 Apresentação	19
4.2 Arquitectura	19
4.3 Tecnologias e Ferramentas Utilizadas	20
4.4 Ambientes de Teste e de Produção	21
4.5 Abrangência	21
4.6 Componentes	21
4.6.1 Frontend	21
4.6.2 Backend	22
4.6.3 Fluxo de Dados	23
4.7 Interfaces	24

5	Método e Planeamento	26
5.1	Metodologia de Desenvolvimento	26
5.1.1	Princípios Metodológicos	26
5.1.2	Organização do Trabalho	26
5.2	Planeamento e Cronograma	26
5.2.1	Fases do Projeto	26
5.3	Análise Crítica ao Planeamento	27
5.3.1	Progresso Atual	27
5.3.2	Desafios e Ajustes	28
5.3.3	Adaptações ao Plano Original	28
	Referências Bibliográficas	30

Lista de Figuras

3.1	Modelo de Entidade-Relação do Sistema	17
3.2	Mapa de Navegação do Sistema	18
4.1	Diagrama de Arquitectura do Sistema. O diagrama apresenta os principais componentes da solução, incluindo o frontend em React, backend em Python, sistema de persistência de dados e as interfaces de comunicação entre os diferentes módulos. A arquitectura modular permite a extensão futura com novos tipos de anotação além do módulo inicial de disentangle-ment.	25
5.1	Cronograma detalhado do projeto (Gantt Chart)	29

Lista de Tabelas

2.1 Comparação técnica detalhada das soluções 12

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 BRAT¹ apresentam limitações tecnológicas e falta de evolução
- Soluções atuais como o Doccano², 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

¹BRAT Rapid Annotation Tool 2024; About BRAT 2024

²Doccano 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 arquitetura proposta.
- **Capítulo 4 - Solução Desenvolvida:** Descreve a implementação, tecnologias utilizadas e componentes desenvolvidos.

- **Capítulo 5 - Testes e Validação:** Apresenta metodologia de testes, resultados e validação da solução.
- **Capítulo 6 - Método e Planeamento:** Detalha a abordagem metodológica e análise do processo de desenvolvimento.
- **Capítulo 7 - Resultados:** Discute os resultados obtidos e avalia o cumprimento dos objetivos.
- **Capítulo 8 - Conclusão:** Sintetiza as contribuições e apresenta perspectivas futuras.

Os anexos incluem documentação técnica adicional, manuais de utilização e 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, 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 dedisentanglement de chat 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	Express.js (protótipo), Migração planeada para Flask/FastAPI
Formato de Anotação	Arquivos .ann para anotações, Arquivos .txt para texto, JSON para comunicação	REST API endpoints, JSON para comunicação	REST API para comunicação e configuração, Suporte planeado para múltiplos formatos
Dados	Sistema de arquivos, Estrutura em diretórios, Sem banco de dados	Database (Django ORM), Suporte a PostgreSQL	Suporte planeado para múltiplos formatos (CSV, TXT, JSON, Markdown), Cache local opcional via SQLite
Deployment	Apache/Lighttpd com CGI, Standalone Python server	Docker containers, Docker Compose support, Production/Development configs	Container Docker único, Setup simplificado
Extensibilidade	Sistema via arquivos .conf, Arquitetura modular, Plugins jQuery	Modular Django architecture, REST API extensibility, Frontend component system	Componentes React modulares, Hooks customizáveis, API extensível
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+ (atual), Python 3.x (planeado), Navegador moderno, Docker (opcional)
Segurança	Autenticação básica, Controle via arquivo	Django authentication, Modern security practices	Autenticação básica baseada em roles (administrador/annotador)

Table 2.1: Comparação técnica detalhada das soluções

^aBRAT Rapid Annotation Tool 2024; About BRAT 2024

^bDoccano 2024; Doccano Developer Guide 2024

3 - Especificação e Modelação

3.1 Análise de Requisitos

3.1.1 Requisitos Funcionais

Requisitos Gerais de Anotação

- **RF1:** Gestão de diferentes formatos de dados (CSV, TXT, JSON, etc.)
- **RF2:** Sistema de organização hierárquica de datasets
- **RF3:** Exportação de dados anotados em múltiplos formatos
- **RF4:** Interface genérica para anotação de dados
- **RF5:** Suporte a múltiplos anotadores por dataset
- **RF6:** Distribuição automática de tarefas entre anotadores disponíveis

Requisitos Específicos - Módulo Disentanglement

- **RF7:** Interface especializada para chat disentanglement
- **RF8:** Sistema de tagging para classificação de mensagens em threads
- **RF9:** Visualização sequencial dos turnos (mensagens)
- **RF10:** Calculo de métricas de anotação

Gestão de Utilizadores

- **RF11:** Autenticação de utilizadores
- **RF12:** Definição de papéis (administrador/anotador)
- **RF13:** Controle de acesso baseado em permissões

3.1.2 Requisitos Não Funcionais

Performance

- **RNF1:** Tempo de resposta adequado para operações interativas
- **RNF2:** Processamento eficiente para múltiplos utilizadores simultâneos

Usabilidade

- **RNF3:** Interface responsiva e adaptável
- **RNF4:** Ferramenta interativa dando feedback visual claro das ações ao utilizador
- **RNF5:** Interface simples, minimalista e intuitiva

Segurança

- **RNF6:** Backup automático de anotações
- **RNF7:** Logging de atividades críticas

3.2 Modelação

3.2.1 Modelo de Dados

O modelo de dados da plataforma é organizado em três componentes principais: gestão de utilizadores, gestão de dados e anotações. A estrutura completa, pode ser visualizada na Figura 3.1, que apresenta em detalhe as entidades e seus relacionamentos.

Componentes Principais

Gestão de Utilizadores O sistema mantém informações básicas dos utilizadores, incluindo credenciais e roles (administrador/anotador). Cada utilizador pode ter acesso a diferentes módulos de anotação, dependendo das permissões atribuídas.

Gestão de Dados Para o módulo de disentanglement, os dados são organizados em chatrooms, onde cada chatroom contém uma sequência de turnos (mensagens). Cada turno possui atributos como timestamp, autor e conteúdo. O conjunto total de chatrooms forma a base de dados a ser anotada. A ferramenta permite que diferentes módulos possam trabalhar com diversos formatos de dados (CSV, TXT, JSON, etc.).

Anotações As anotações são armazenadas independentemente dos dados originais, mantendo referências para a chatroom, turno, utilizador e timestamp. No contexto do disentanglement, cada anotação associa turnos a threads específicas, permitindo a reconstrução das conversas paralelas.

Fluxo de Dados

O fluxo de dados na plataforma segue o seguinte padrão: 1. Importação de dados através da interface administrativa 2. Distribuição automática de tarefas aos anotadores 3. Processo de anotação no módulo específico 4. Armazenamento contínuo das anotações via backend 5. Cálculo de métricas e análise das anotações

Armazenamento

A solução proposta contempla: - Sistema de arquivos estruturado para gestão de diferentes formatos de dados - Base de dados para gestão de utilizadores e estado da aplicação - Sistema de cache para otimização de performance - Armazenamento independente de anotações para facilitar análise e processamento

Esta estrutura visa suportar diferentes tipos de dados e formatos, mantendo a flexibilidade necessária para implementação de novos módulos de anotação.

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 único módulo desta arquitetura modular, no entanto o objetivo 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.

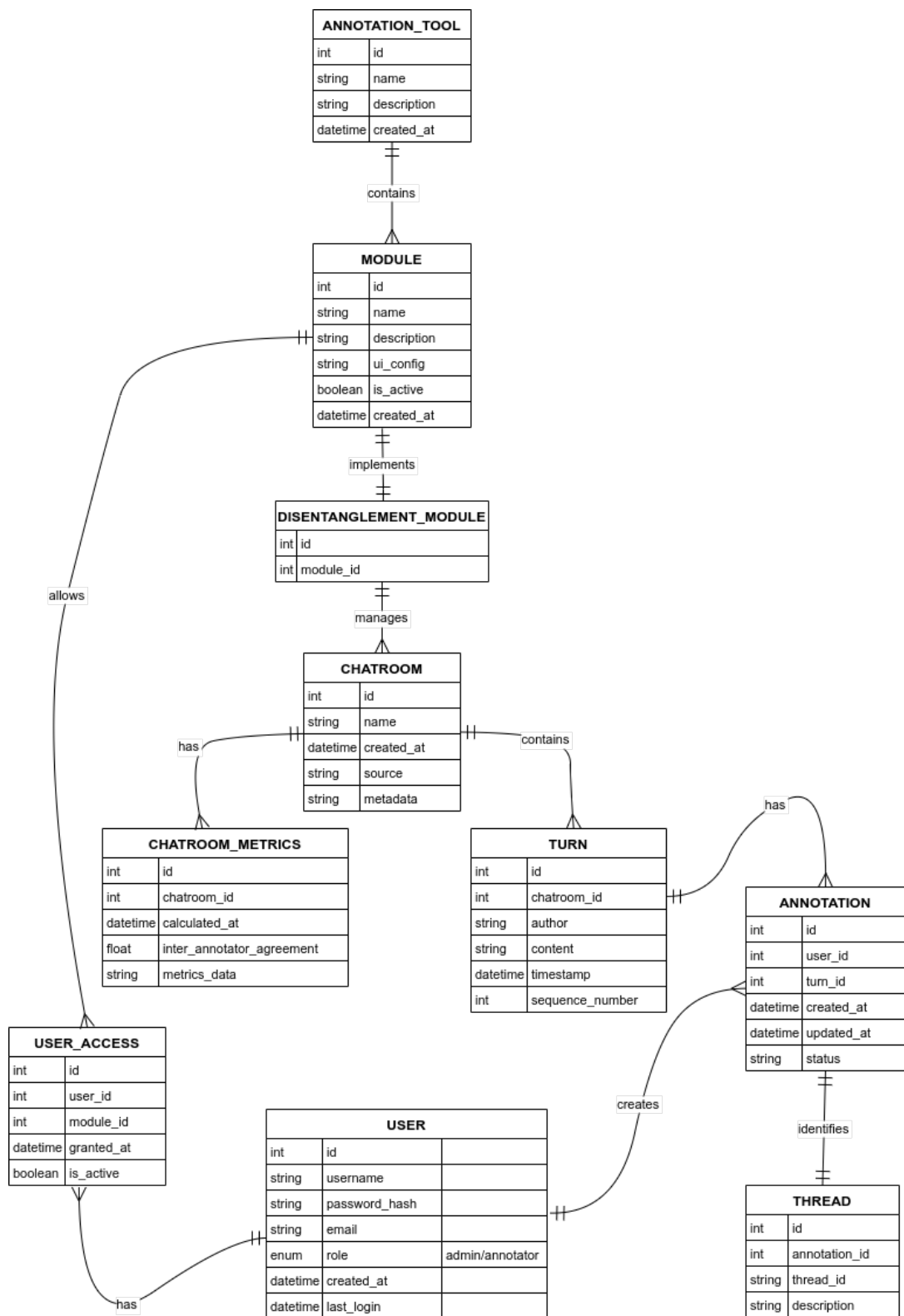


Figure 3.1: Modelo de Entidade-Relação do Sistema

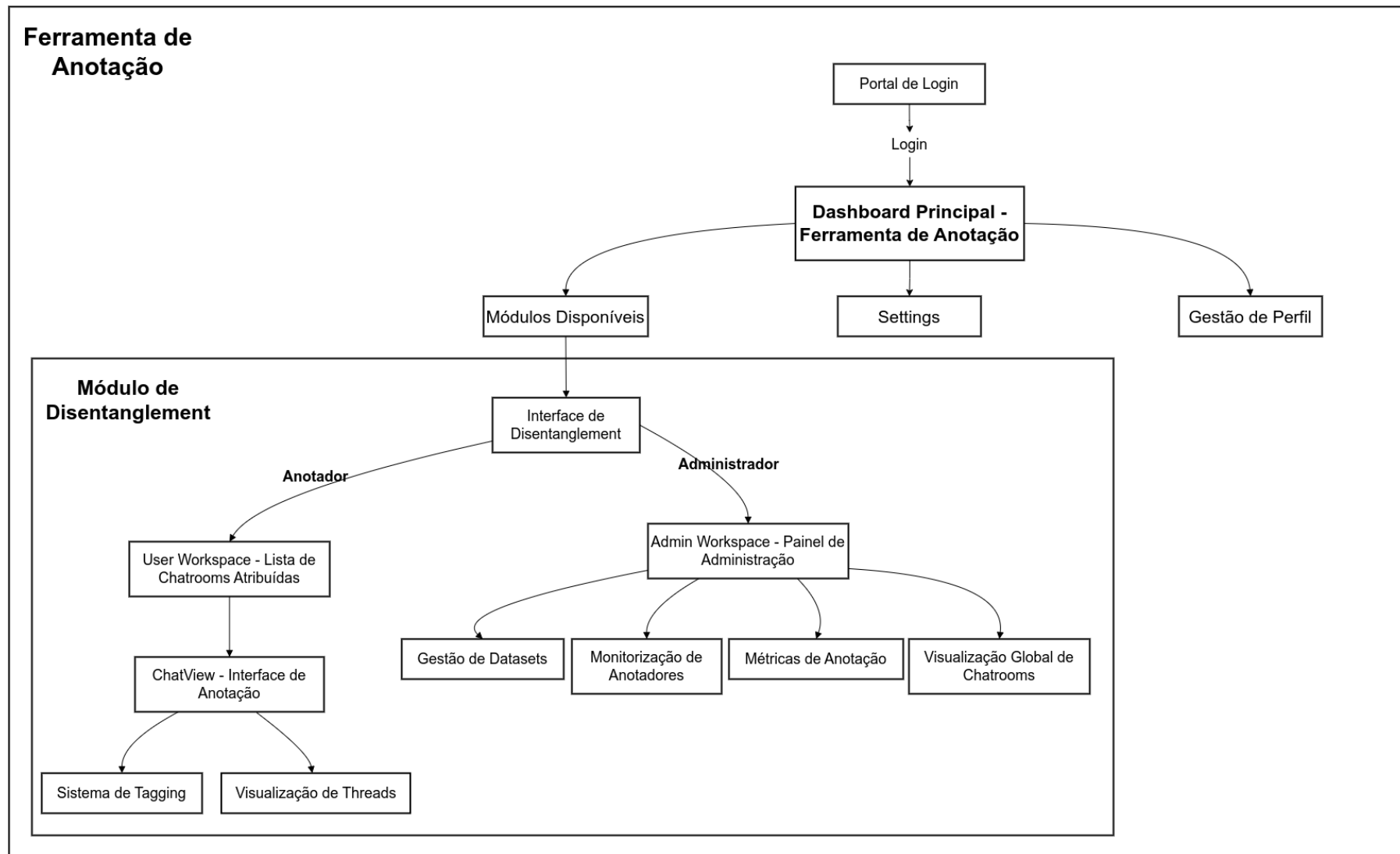


Figure 3.2: Mapa de Navegação do Sistema

4 - Solução Proposta

4.1 Apresentação

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

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

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

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

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

Insights do Protótipo O desenvolvimento exploratório inicial proporcionou descobertas importantes:

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

4.2 Arquitectura

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

Operacionalização O primeiro foca-se na operacionalização através de:

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

Modularidade O segundo centra-se na modularidade através de:

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

Evolução Planeada A arquitectura suporta evolução em várias dimensões:

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

4.3 Tecnologias e Ferramentas Utilizadas

A implementação assenta em tecnologias web estabelecidas, escolhidas conforme estado da arte e conforme os requisitos do projeto. O frontend utiliza React para a interface de anotação, enquanto o backend em Python gere o processamento de dados, permitindo integração eficiente com ferramentas de análise e processamento.

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

- Gestão eficiente de estado através de Hooks
- Componentes reutilizáveis para consistência da interface
- Rendering otimizado para operações frequentes
- Extensa documentação e comunidade activa

Backend O desenvolvimento em Python permite:

- Integração com bibliotecas de processamento de dados
- Implementação eficiente de métricas e análises
- Alinhamento com o ecossistema do departamento
- Extensibilidade para funcionalidades futuras

4.4 Ambientes de Teste e de Produção

O desenvolvimento segue uma abordagem iterativa com dois ambientes distintos:

Ambiente de Desenvolvimento Suporta o desenvolvimento activo e testes:

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

Ambiente de Produção Garante estabilidade e performance:

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

4.5 Abrangência

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

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

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

Extensibilidade A arquitectura suporta expansão através de:

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

4.6 Componentes

4.6.1 Frontend

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

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

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

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

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

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

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

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

4.6.2 Backend

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

Arquitetura do Backend O backend segue uma arquitetura em camadas:

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

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

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

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

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

4.6.3 Fluxo de Dados

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

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

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

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

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

Distribuição As conversas são disponibilizadas através da interface web:

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

4.7 Interfaces

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

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

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

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

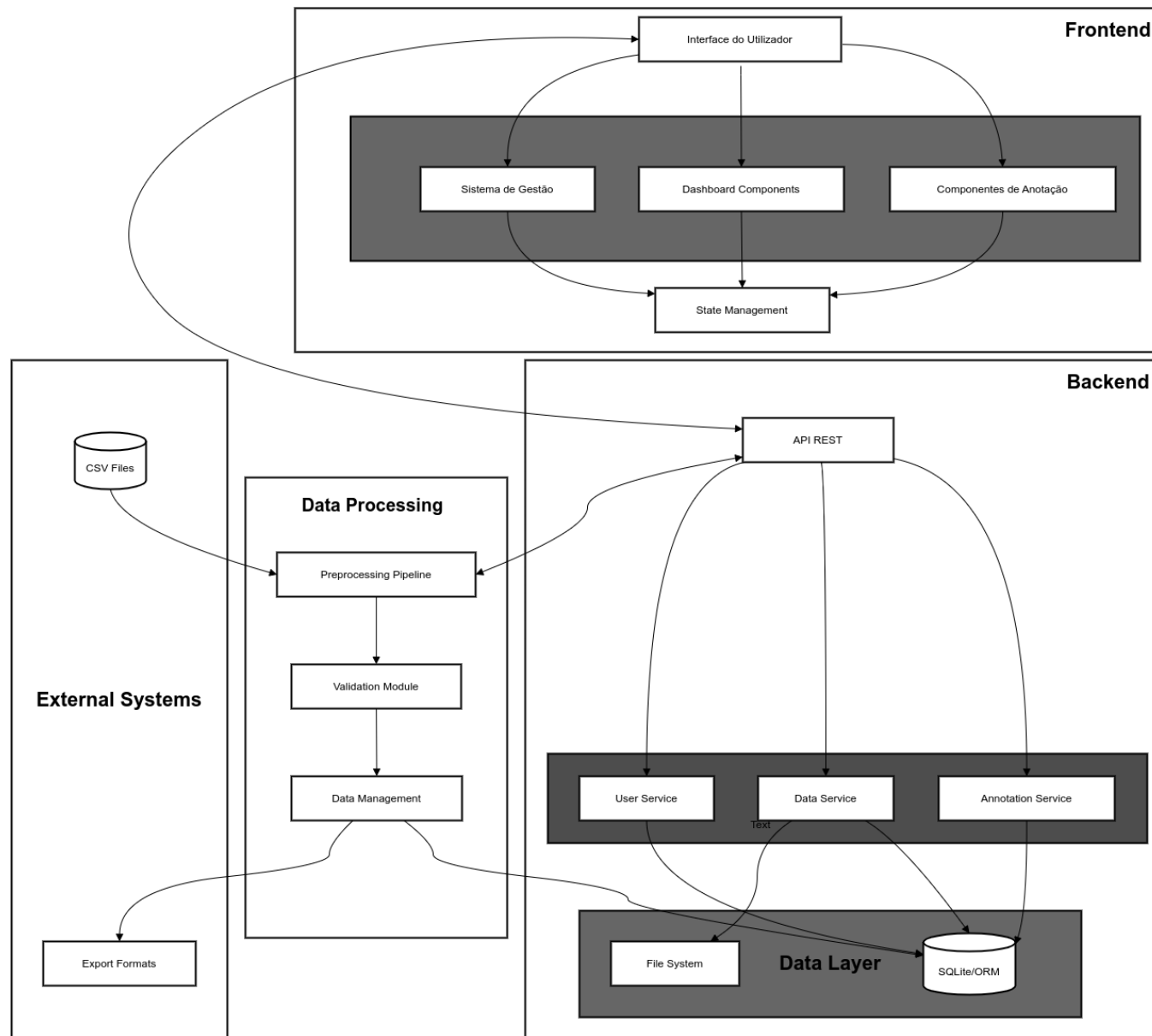


Figure 4.1: Diagrama de Arquitectura do Sistema. O diagrama apresenta os principais componentes da solução, incluindo o frontend em React, backend em Python, sistema de persistência de dados e as interfaces de comunicação entre os diferentes módulos. A arquitectura modular permite a extensão futura com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

5 - Método e Planeamento

5.1 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.

5.1.1 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

5.1.2 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
- **Retrospectiva:** Análise do processo e identificação de melhorias

5.2 Planeamento e Cronograma

O planeamento do projeto, representado na Figura 5.1, está organizado em fases distintas que refletem a evolução da plataforma desde o protótipo inicial até à solução final.

5.2.1 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

5.3 Análise Crítica ao Planeamento

5.3.1 Progresso Atual

O desenvolvimento inicial centrou-se na validação do conceito através de um protótipo funcional, que permitiu:

- Validar a viabilidade técnica da solução proposta
- Confirmar a adequação da interface para tarefas de disentanglement
- Identificar pontos de melhoria no fluxo de trabalho
- Recolher feedback inicial dos utilizadores

5.3.2 Desafios e Ajustes

Desafios Técnicos

- Complexidade na implementação da arquitetura modular
- Migração do backend para Python mantendo funcionalidade
- Otimização do processamento para datasets extensos

Desafios Operacionais

- Balanceamento entre funcionalidades desejadas e prazos
- Adaptação a requisitos emergentes do AISIC LAB

5.3.3 Adaptações ao Plano Original

Com base na experiência do desenvolvimento inicial, foram identificados ajustes necessários:

- Maior ênfase em testes de usabilidade
- Simplificação inicial de algumas funcionalidades
- Foco aumentado na documentação técnica
- Integração contínua de feedback dos utilizadores

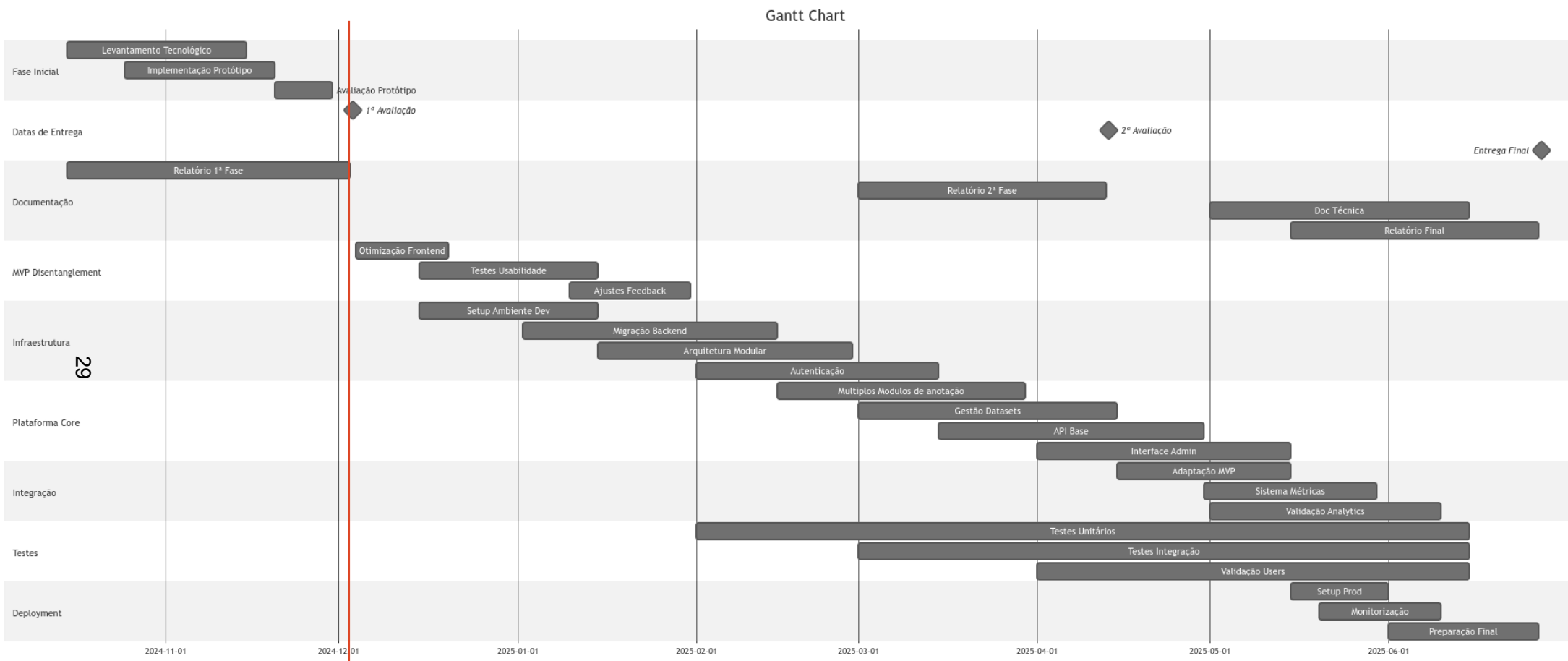


Figure 5.1: Cronograma detalhado do projeto (Gantt Chart)

Referências Bibliográficas

- About BRAT (2024). URL: <https://brat.nlplab.org/about.html> (visited on 11/2024).
- BRAT Rapid Annotation Tool (2024). URL: <https://github.com/nlplab/brat/tree/master> (visited on 11/2024).
- Doccano (2024). URL: <https://github.com/doccano/doccano/> (visited on 11/2024).
- Doccano Developer Guide (2024). URL: https://doccano.github.io/doccano/developer_guide/ (visited on 11/2024).
- Elsner, M. and Charniak, E. (2010). “Disentangling Chat”. In: *Computational Linguistics*.
- Lopes, F. (2024). Annotation tool for chat disentanglement. URL: https://github.com/fabiofalopes/annotation_ui (visited on 11/2024).
- Matos-Carvalho, J. P. (2024). The Lusófona L^AT_EX Template User’s Manual. Lusófona University. URL: <https://github.com/jpmcarvalho/UL-Thesis>.