

## **SADo - Sistema de Avaliação Docente**

### **Sistema de Avaliação Docente (SAD)**

Levantamento de Requisitos

Versão:1.1.2

Data: 10 Julho, 2025

Identificador do documento: LR

Localização: [https://github.com/MarcioJrAlmeida/tads\\_eng\\_softw](https://github.com/MarcioJrAlmeida/tads_eng_softw)

## Histórico de revisões do modelo

Versão	Data	Autor	Descrição	Localização
1.0.0	24/ABR/2025	MarcioJunior	Repositório criado com <i>CHANGELOG.md</i> para controle de versões. Documentos como Cronograma, Proposta e dúvidas ao Dono do Produto adicionados em <i>/documents</i> .	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>
1.0.1	25/ABR/2025	MarcioJunior	Pastas adicionadas para organizar o projeto: <i>/front</i> , <i>/back</i> , <i>/banco</i> e <i>/documents</i> .	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>
1.0.2	27/ABR/2025	Gabriel Henrique	Documentos <i>Aprimoramento dos Requisitos</i> e <i>Passo a Passo Requisitos</i> adicionados em <i>/documents</i> .	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>
1.0.3	02/MAI/2025	MarcioJunior	Adicionado em <i>/banco/modelos</i> o arquivo <i>modelo_ER.png</i> , primeira versão do Modelo Entidade-Relacionamento. Documento <i>Levantamento de Requisitos.docx</i> adicionado à pasta <i>/documents</i> .	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>
1.0.4	23/MAI/2025	Gabriel Henrique	Adicionado à pasta <i>/documents</i> o arquivo de apresentação <i>Sistema-de-Avaliacao-Doctent e-SADo.pptx</i> . Arquivos <i>Cronograma - Projeto ES.xlsx</i> e <i>PROPOSTA DE PROJETO.docx</i> foram atualizados com novas versões.	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>
1.0.5	25/MAI/2025	Jão Pedro Neve	README atualizado com a descrição da pasta <i>/documents</i> .	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>

1.0.6	26/MAI/2025	Jão Pedro Neves	Pastas hierárquicas e arquivos de configuração adicionados para o Frontend: <i>/app</i> com subpastas <i>/assets</i> , <i>components</i> , <i>pages</i> , <i>script</i> e <i>streamlit</i> ; arquivos <i>login.py</i> , <i>config.yaml</i> , <i>requirements.txt</i> e <i>run_app.py</i> .	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>
1.0.7	29/MAI/2025	MarcioJunior	Pasta <i>/banco</i> atualizada com o modelo Conceitual revisado após reunião com o cliente e o professor de Banco de Dados. Modelo Lógico também adicionado.	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>
1.0.8	04/JUN/2025	Gabriel Henrique	Na pasta <i>/banco</i> , modelo Conceitual revisado após reunião com cliente e professor, modelo Lógico adicionado, além de novos modelos em PDF.	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>
1.0.9	05/JUN/2025	MarcioJunior	Na pasta <i>/banco</i> , criadas as pastas <i>/ddl</i> (com <i>create_table.sql</i> ) e <i>/dml</i> (com <i>insert_table.sql</i> e <i>select_table.sql</i> contendo dados fictícios).	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>
1.1.0	26/JUN/2025	Jão Pedro Neves	Alterações no <i>/front</i> : criado <i>backend_app.py</i> e atualizado <i>forms.py</i> para consultas via API. Em <i>/front/app</i> , desenvolvidas rotas de API: <i>configuracao_service.py</i> e <i>perguntas_service.py</i> . No <i>/back</i> : adicionadas pastas <i>/config</i> (configuração modo desenvolvimento e dados do banco), <i>/database</i> (conexão	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>

			e queries rápidas) e <i>/services</i> (teste de consulta ao banco).	
1.1.1	28/JUN/2025	Marcílio Correia	Alterações em <i>/front/app/api</i> : novas funções de salvar, atualizar e excluir perguntas em <i>perguntas_service.py</i> ; funções para obter modelo de avaliação e listar avaliações em <i>avaliacao_service.py</i> . Em <i>/front</i> : atualizados <i>forms.py</i> e criado <i>edicao_forms.py</i> ; botão "Editar Formulários" ajustado para <i>edicao_forms.py</i> ; consultas agora verificam modo BD ou CSV. <i>requirements.txt</i> atualizado com Flask. No <i>/banco</i> , criada pasta <i>/data/</i> com tabelas em CSV para consultas offline.	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>
1.1.2	13/JUL/2025	Marcílio Correia	Na pasta <i>/pages</i> , realizados ajustes e finalizações no Frontend, Backend e NLP.	<a href="https://github.com/MarcioJrAlmeida/tads_eng_softw">https://github.com/MarcioJrAlmeida/tads_eng_softw</a>

## Aprovadores

Nome	Função
Marcílio Correia	Scrum Master
MarcioJunior	Desenvolvedor
Gabriel Henrique	Desenvolvedor
Jãoo Pedro Neves	Desenvolvedor

## 1. Introdução

### 1.1. Propósito

Desenvolver uma API para Avaliação Docente que colete e analise dados de formulários, gere gráficos e aplique análise de sentimento em respostas abertas, auxiliando a gestão acadêmica na tomada de decisões sobre o desempenho dos professores.

### 1.2. Público Alvo

O sistema é destinado à gestão acadêmica da instituição, especialmente ao diretor do campus, que utilizará os dados para avaliar o desempenho dos docentes. Também pode atender coordenadores e setores pedagógicos interessados na melhoria da qualidade do ensino.

### 1.3. Escopo

O projeto abrange o desenvolvimento de uma API capaz de processar dados de formulários de avaliação docente, gerar visualizações gráficas e realizar análise de sentimento. Está incluída a integração com banco de dados, uso de bibliotecas de IA e uma interface em Streamlit para visualização dos resultados.

### 1.4. Definições e Abreviações

API: Interface de Programação de Aplicações, que permite a comunicação entre sistemas.

PLN: Processamento de Linguagem Natural, área da IA usada para interpretar textos escritos.

RF: Requisito Funcional – funcionalidades que o sistema deve oferecer.

RNF: Requisito Não-Funcional – características como desempenho, segurança e usabilidade.

Diretor: Responsável por analisar os relatórios gerados para tomada de decisões acadêmicas.

Docente: Professor avaliado por meio dos formulários aplicados.

Usuário: Aluno responsável por preencher o formulário de avaliação.

Streamlit: Framework usado para construir a interface interativa do sistema.

PyODBC: Biblioteca usada para conectar a API ao banco de dados.

Transformers: Biblioteca de modelos de linguagem usados para análise de sentimento.

Kaggle Hub: Plataforma utilizada para importar modelos de machine learning.

### 1.5. Referências

[1] Documentação oficial do Flask. Disponível em: <https://flask.palletsprojects.com/>

[2] Scikit-learn: Machine Learning in Python. Disponível em: <https://scikit-learn.org/>

[3] Hugging Face Transformers. Disponível em: <https://huggingface.co/transformers/>  
[4] Streamlit Documentation. Disponível em: <https://docs.streamlit.io/>  
[5] Kaggle Models Hub. Disponível em: <https://www.kaggle.com/models>  
[6] Padrão de Homologação de Sistemas – MDS/TCE-PE. Localização:  
/documentos/mds\_tce\_pe

## 1.6. Visão geral do documento

A seção 2 apresenta a visão geral do sistema.  
A seção 3 descreve as premissas e restrições.  
A seção 4 enumera os requisitos funcionais.  
A seção 5 lista os requisitos não-funcionais.

## 2. Visão Geral do Produto

O sistema é uma API de Avaliação Docente que coleta e processa dados de formulários respondidos pelos alunos, gerando gráficos e relatórios para análise do desempenho dos professores. Além disso, aplica técnicas de processamento de linguagem natural para analisar o sentimento das respostas abertas e conta com uma interface em Streamlit que facilita a visualização dos dados, auxiliando a gestão acadêmica na tomada de decisões estratégicas.

### 2.1. Descrição dos usuários

Administrador: responsável pela gestão técnica da API, manutenção do sistema e controle dos formulários.

Diretor do campus: utiliza os relatórios e análises para monitorar e tomar decisões sobre a avaliação dos professores.

Alunos: preenchem os formulários de avaliação, fornecendo informações essenciais para o funcionamento da API.

## 3. Premissas e Restrições

Premissa 1: Os alunos irão responder aos formulários de avaliação de forma sincera e frequente.

Premissa 2: A infraestrutura da faculdade suportará o funcionamento contínuo da API e da interface em Streamlit.

Premissa 3: As bibliotecas e ferramentas utilizadas (Flask, pyodbc, Streamlit, Scikit-learn, Transformers) estarão atualizadas e compatíveis durante o desenvolvimento e uso.

Premissa 4: O diretor e o administrador terão acesso à internet e dispositivos adequados para utilizar a interface web.

Restrição 1: A API depende dos dados fornecidos pelos alunos; avaliações insuficientes

podem impactar a qualidade das análises.

Restrição 2: Limitação no processamento de linguagem natural devido ao volume e qualidade das respostas abertas.

Restrição 3: O sistema está restrito ao ambiente da faculdade, sem suporte para usuários externos ou multiplataformas.

Restrição 4: Segurança e privacidade dos dados devem seguir as políticas institucionais, limitando acesso e uso das informações.

#### **4. Requisitos Funcionais**

RF001 - Login de usuários: O sistema deve permitir login por e-mail e senha para alunos, gestores e diretores. Docentes não têm acesso ao sistema, apenas são avaliados. O administrador é um usuário técnico. Senhas devem ser armazenadas com hash seguro, evitando texto puro.

RF002 - Visualização de docentes pelo aluno: O aluno deve visualizar a lista de docentes vinculados ao seu curso ou aos quais teve aula no período. Deve ser possível filtrar docentes para garantir avaliações relevantes.

RF003 - Resposta ao formulário de avaliação: O aluno deve responder um formulário obrigatório de múltipla escolha para cada docente avaliado, com perguntas definidas pelo administrador.

RF004 - Comentário opcional na avaliação: O aluno pode deixar comentário textual opcional em cada avaliação. Os comentários exibidos a gestores/diretores devem ser anônimos para proteger a identidade do aluno.

RF005 - Prevenção de avaliações duplicadas: O sistema deve impedir que um aluno avalie o mesmo docente mais de uma vez no mesmo período letivo. Deve permitir edição da avaliação dentro do prazo, se aplicável.

RF006 - Acesso do gestor às médias das avaliações: O gestor deve visualizar as médias das avaliações dos docentes vinculados ao seu curso, respeitando o anonimato e limite mínimo de respostas para exibição.

RF007 - Acesso do diretor às médias gerais: O diretor pode acessar médias agregadas de todos os docentes, independentemente do curso, com filtros e garantia de anonimato.

RF008 - Dashboard visual para gestores e diretores: O sistema deve oferecer um dashboard com visualizações gráficas (gráficos, indicadores, comparativos) dos resultados das avaliações.

RF009 - Gerenciamento das perguntas pelo administrador:O administrador deve poder criar, editar e remover perguntas do formulário de avaliação, mantendo o questionário dinâmico.

RF010 - Associação docente-curso:O sistema deve manter o vínculo entre docentes e cursos, permitindo associação múltipla e uso desse vínculo para filtrar docentes exibidos aos alunos e gestores.

## **5. Requisitos Não Funcionais**

RNF001 - Acesso via navegador:O sistema deve ser acessível por navegadores modernos e compatível com diferentes dispositivos, preferencialmente em ambiente interno (intranet).

RNF002 - Implementação em formato MVP:O sistema será desenvolvido inicialmente como um MVP, priorizando funcionalidades essenciais (login, avaliação e médias), com possibilidade de evolução posterior.

RNF003 - Armazenamento seguro de senhas:As senhas dos usuários devem ser armazenadas com hash seguro (ex: bcrypt ou Argon2), nunca em texto puro ou com criptografia reversível.

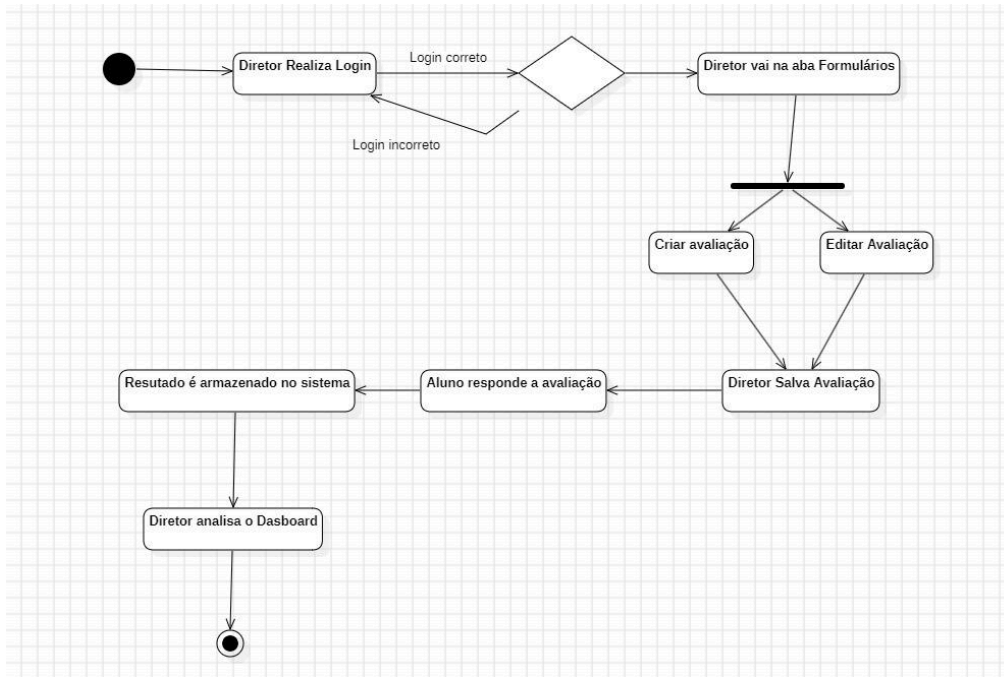
RNF004 - Uso de banco de dados relacional:O sistema deve utilizar um banco de dados relacional, garantindo integridade dos dados e suporte eficiente a consultas estruturadas.

RNF005 - Interface responsiva e acessível:A interface deve ser responsiva, adaptando-se a diferentes telas (desktop e mobile), e seguir boas práticas de acessibilidade.

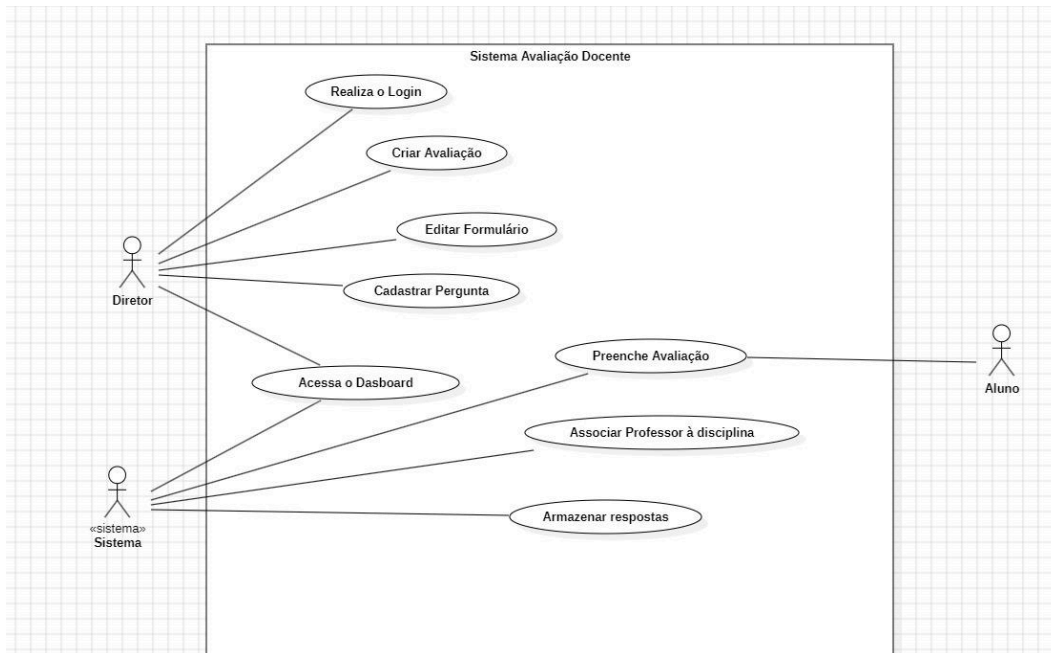
RNF006 - Tempo de resposta:O sistema deve responder às ações principais em até 2 segundos, sob condições normais de uso, garantindo boa experiência para o usuário.



## 6. Diagramas



### Diagrama Atividades



### Diagrama Caso de uso

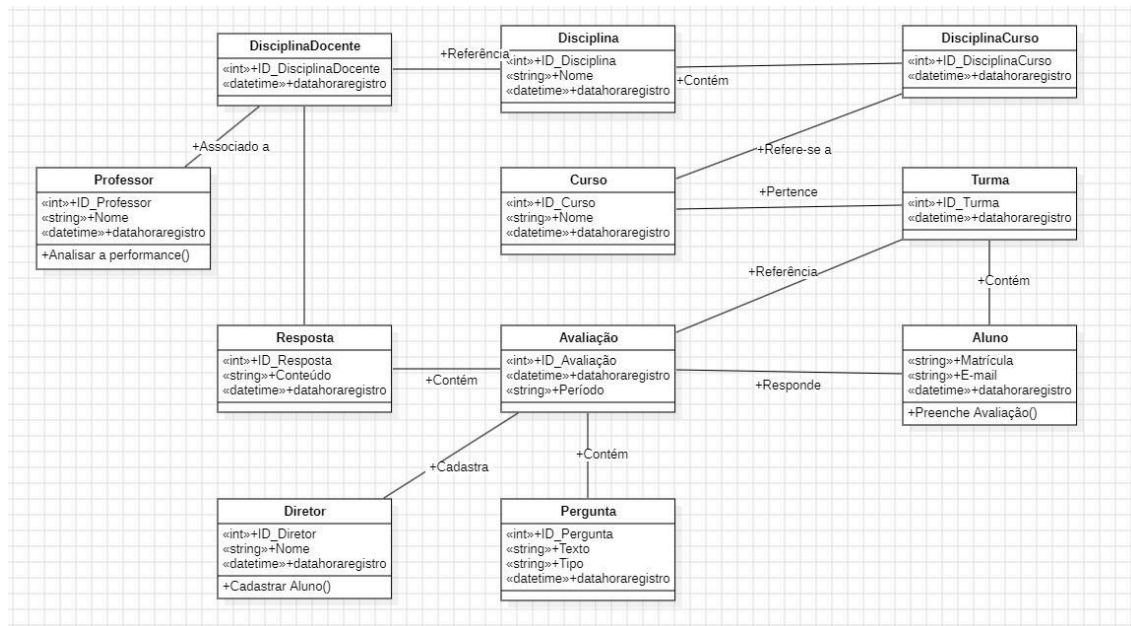


Diagrama Classe objeto