



Hackathon Unisanta 2025

Desafio técnico da T2S para a competição

[INTRODUÇÃO](#)

[EQUIPES](#)

[VISÃO GERAL](#)

[REQUISITOS FUNCIONAIS](#)

[REQUISITOS NÃO FUNCIONAIS](#)

[ORIENTAÇÕES](#)

[PREMIAÇÃO](#)

INTRODUÇÃO

Este documento tem por objetivo estabelecer critérios e requisitos do desafio proposto pela T2S para o Hackathon Unisanta 2025, competição organizada pela Universidade Santa Cecília aos alunos dos cursos de tecnologia da informação

EQUIPES

São as equipes sorteadas para realização do desafio da T2S e seus respectivos integrantes:
[TODO]

VISÃO GERAL

A proposta para o desafio da T2S consiste no desenvolvimento de uma plataforma SaaS para avaliação automatizada de projetos de software utilizando Inteligência Artificial. A persona alvo para esta ferramenta é um analista de qualidade de software.

Cada equipe desenvolverá sua própria ferramenta avaliadora. A fonte de dados para a avaliação serão os repositórios públicos no GitHub de todas as outras equipes participantes. É desejável que a ferramenta seja capaz de clonar e analisar os arquivos de um projeto.

O objetivo é que a plataforma forneça uma análise inteligente e estruturada, servindo como uma ferramenta de auxílio para o júri técnico, otimizando o processo de revisão de código e documentação.

REQUISITOS FUNCIONAIS

São requisitos funcionais do portal:

1. Dashboard de Avaliação:

- 1.1. A ferramenta deve possuir uma **interface web** onde o usuário poderá submeter as URLs dos repositórios do GitHub a serem avaliados (ou paths locais dos projetos clonados do GitHub);
- 1.2. O processo de análise deve ser **assíncrono**, permitindo que o usuário acompanhe a evolução parcial da avaliação em tempo real na interface;
- 1.3. A plataforma deve apresentar um dashboard principal listando os projetos avaliados e suas pontuações resumidas em um ranking;

2. Relatório de Análise:

-
- 2.1. Para cada projeto avaliado, a ferramenta deve gerar um relatório detalhado em **formato Markdown**.
 - 2.2. Este relatório deve ser exibido no frontend e ser estruturado de forma que possa ser facilmente copiado para um arquivo de documentação (ex: `TECH_REVIEW.MD`).
 - 2.3. O relatório deve conter a pontuação para cada critério de avaliação e uma justificativa textual gerada pela IA para a nota atribuída.
 3. **Critérios de Avaliação (Motor da IA):** A IA deverá avaliar os projetos com base em dois eixos principais, com pontuação igualmente distribuída:
 - 3.1. **Qualidade de Engenharia de Software (Adaptado de ISO/IEC 25010):**
 - 3.1.1. **Adequação Funcional:** Análise do `README.md` e da estrutura do projeto para verificar a coerência entre o que foi proposto e o que parece ter sido implementado.
 - 3.1.2. **Manutenibilidade:** Avaliação da clareza do código, uso de comentários, padronização (linters) e complexidade do código.
 - 3.1.3. **Confiabilidade:** Verificação da existência de suítes de testes (ex: arquivos de teste, frameworks como Jest, PyTest, etc.).
 - 3.1.4. **Usabilidade (Clareza):** Análise da qualidade e completude da documentação (`README.md`, guias de instalação, etc.).
 - 3.1.5. **Desempenho:** Análise da velocidade de execução e do consumo de recursos computacionais.
 - 3.2. **Qualidade de Aplicação de IA:**
 - 3.2.1. **Origem e Tratamento dos Dados:** Análise sobre como a IA é alimentada (fonte de dados, pré-processamento).
 - 3.2.2. **Técnicas Aplicadas:** Identificação das técnicas de IA utilizadas (ex: engenharia de prompt, RAG, fine-tuning).
 - 3.2.3. **Estratégia de Validação e Escolha de Modelos:** Verificação se há justificativa para a escolha do LLM e como a solução é validada.
 - 3.2.4. **Métricas de Avaliação, Custo e Desempenho:** Análise de como a equipe mede o sucesso de sua IA e considerações sobre custo/eficiência.
 - 3.2.5. **Segurança e Governança:** Avaliação de riscos (injeção de prompt) e implementação de mecanismos de segurança (guardrails).

REQUISITOS NÃO FUNCIONAIS

São requisitos não-funcionais do portal:

1. Tecnologias como linguagem de programação, sistema gerenciador de banco de dados, frameworks, intermediários e modelos de linguagem são de livre escolha.

2. A aplicação pode ser monolítica ou multicamadas, mas a manipulação de dados precisa ser dinâmica e assíncrona, por meio de uma API RESTFul trafegando dados em formato [JSON](#).
3. O código fonte da aplicação deve ser mantido, desde o início do projeto até sua publicação, em repositório público no GitHub.

ORIENTAÇÕES

- O prazo é curto, mas a proposta é para entrega de um [MVP](#). Então recomenda-se ser muito pragmático e utilizar [metodologias ágeis](#). Lembre-se que o “perfeito” é inimigo do “feito”.
- A equipe é livre para escolher ferramentas e tecnologias, mas recomendamos que sejam de código aberto.
- Vocês estão desenvolvendo um produto, então é importantíssimo se preocupar com a apresentação do produto do ponto de vista comercial. Use e abuse de técnicas e [copywriting](#).
- Vocês estão vivendo a era dos grandes modelos de linguagem, então usem e abusem deles para enriquecer os conteúdos, agilizar o processo de desenvolvimento e aumentar a produtividade de vocês. **Vocês estão livres para usar chat de IA ou plugins integrados à IDE.**

PREMIAÇÃO

A T2S oferecerá à equipe vencedora:

- 1 livro “[Engenharia de Prompt para Devs](#)”, da editora Casa do Código, para cada integrante da equipe;
- Uma oportunidade de estágio na [T2S LABS](#).