Lucas Cabral Soares e Maria Eduarda Amaral Muniz
{ maria.amaral, lucas.cabral } @sga.pucminas.br

Documento de Visão para o Sistema XXXX

21 de Setembro de 2025

Proposta dos alunos Lucas Cabral Soares e Maria Eduarda Amaral Muniz ao curso de Engenharia de Software como projeto de Trabalho de Conclusão de Curso (TCC) sob orientação de conteúdo do professor Cleiton Silva Tavares e orientação acadêmica do professor YYY.

OBJETIVOS

O presente documento tem como objetivo descrever a visão do sistema Plataforma de Geração e Análise de Grafos de Teste, que será desenvolvido como parte do Trabalho de Conclusão de Curso. A plataforma busca apoiar engenheiros de software, pesquisadores e estudantes na prática de testes estruturais (caixa branca) e funcionais (caixa preta), fornecendo meios automatizados e interativos para geração de Grafos de Fluxo de Controle (GFC) e Grafos de Causa-Efeito (GCE).

O sistema visa atender às necessidades de reduzir a complexidade da geração manual de grafos de teste, auxiliar na identificação de casos de teste mais eficientes para diferentes critérios de cobertura (comandos, decisões, condições e caminhos), apoiar a derivação de tabelas de decisão e casos de teste funcionais a partir de grafos de causa-efeito e fornecer relatórios visuais e métricas que facilitem a análise da qualidade dos testes.

ESCOPO

A plataforma oferecerá recursos que possibilitam ao usuário realizar tanto testes estruturais quanto funcionais em um ambiente único e integrado. No contexto de testes estruturais, será possível importar código-fonte em linguagens suportadas, como Python, Java e C#, para que a aplicação gere automaticamente o Grafo de Fluxo de Controle. A partir desse grafo, o sistema calculará métricas de cobertura de comandos, decisões, condições e caminhos, destacando

trechos não exercitados pelo conjunto de testes existente. Já no contexto de testes funcionais, a plataforma permitirá que o usuário modele graficamente as causas e efeitos extraídos da especificação, resultando em um Grafo de Causa-Efeito. Esse grafo será convertido em uma tabela de decisão e, posteriormente, em casos de teste funcionais. O sistema também fornecerá integração entre os dois enfoques, permitindo a comparação entre a cobertura estrutural e funcional de um mesmo sistema e gerando relatórios consolidados em formatos como PDF, CSV e JSON. Como diferencial em relação a sistemas concorrentes, que geralmente tratam apenas de cobertura de código (por exemplo, JaCoCo ou Cobertura) ou apenas de geração de tabelas de decisão (como Tcases), esta plataforma unificará as duas abordagens em uma solução acadêmica e acessível, favorecendo tanto o ensino quanto a prática profissional.

FORA DO ESCOPO

A plataforma não terá como objetivo executar diretamente os testes automatizados em frameworks como JUnit, PyTest ou Selenium, pois sua função principal será a geração de casos de teste e não a sua execução. Também não será oferecido suporte a linguagens de programação de baixo nível, como Assembly e C puro, uma vez que o foco inicial está em linguagens modernas e amplamente utilizadas. Outro ponto fora do escopo será a integração direta com IDEs complexas, como Eclipse ou IntelliJ, já que a interação proposta será realizada exclusivamente por meio da interface web da plataforma.

GESTORES, USUÁRIOS E OUTROS INTERESSADOS

Nome	Cleiton Silva Tavares
Qualificação	Gestor/Orientador
Responsabilidades	Auxiliar no acompanhamento do desenvolvimento da plataforma e fornecer orientações acadêmicas.

Nome	Danilo de Quadros Maia Filho
Qualificação	Gestor/Orientador
Responsabilidades	Auxiliar no acompanhamento do desenvolvimento da plataforma e fornecer orientações acadêmicas.

Nome	Leonardo Vilela Cardoso

Qualificação	Gestor/Orientador
Responsabilidades	Auxiliar no acompanhamento do desenvolvimento da plataforma e fornecer orientações acadêmicas.

Nome	Raphael Ramos Dias Costa
Qualificação	Gestor/Orientador
Responsabilidades	Auxiliar no acompanhamento do desenvolvimento da plataforma e fornecer orientações acadêmicas.

Nome	Lucas Cabral Soares
Qualificação	Estudante/Desenvolvedor
Responsabilidades	Responsável pelo desenvolvimento da plataforma ao longo da disciplina de TCC I.

Nome	Maria Eduarda Amaral Muniz
Qualificação	Estudante/Desenvolvedor
Responsabilidades	Responsável pelo desenvolvimento da plataforma ao longo da disciplina de TCC I.

Qualificação	Estudante de Engenharia de Software
Responsabilidades	Participa de disciplinas relacionadas a teste de software, aprendendo fundamentos de teste estrutural e funcional. Utiliza a plataforma como ferramenta de apoio ao aprendizado, explorando exemplos práticos.
Como poderá usar o sistema?	Gerando grafos de fluxo de controle a partir de pequenos trechos de código fornecidos em aula e construindo grafos de causa-efeito a partir de especificações, a fim de compreender a derivação de casos de teste.

Qualificação	Professor Universitário de Teste de Software

Responsabilidades	Leciona disciplinas de qualidade e teste de software, criando exercícios, exemplos e avaliações. Precisa de ferramentas didáticas para ilustrar conceitos de grafos de fluxo de controle e grafos de causa-efeito.
Como poderá usar o sistema?	Utilizando a plataforma para gerar grafos de fluxo de controle a partir de trechos de código de exemplo e construir grafos de causa-efeito a partir de especificações, de modo a ilustrar em sala de aula como casos de teste podem ser derivados.

Qualificação	Engenheiro de Software Profissional
Responsabilidades	Responsável por validar qualidade de código e especificações em projetos reais. Atua no planejamento e execução de testes em nível de unidade, integração e sistema.
Como poderá usar o sistema?	Importando código do projeto para gerar automaticamente o grafo de fluxo de controle, avaliando a cobertura dos testes já implementados, e utilizando o módulo funcional para derivar casos de teste adicionais a partir de requisitos.

LEVANTAMENTO DE NECESSIDADES

1. Automatização da geração de grafos de fluxo de controle. Atualmente, professores e engenheiros realizam esse processo de forma manual, o que demanda tempo e aumenta as chances de falhas. Ao automatizar a construção desses grafos, o sistema reduz o esforço e garante maior precisão.

2. Derivação de casos de teste funcionais a partir de especificações. A geração manual de tabelas de decisão é suscetível a erros e pode não contemplar todas as combinações necessárias. O uso de grafos de causa-efeito e a derivação automática de tabelas proporcionam maior confiabilidade no processo.

3. Consolidação das métricas de cobertura estrutural e funcional em um único ambiente. Hoje, essas análises são feitas em ferramentas diferentes, dificultando a visão integrada dos resultados. Com a plataforma, será possível avaliar de forma unificada a cobertura de código e os casos de teste funcionais.

4. Visualização interativa e didática das estruturas de teste. A ausência de ferramentas que ilustram de maneira clara os grafos de fluxo de controle e de causa-efeito dificulta o aprendizado e a análise prática. A plataforma permitirá que esses elementos sejam explorados de forma visual e compreensível.

FUNCIONALIDADES DO PRODUTO

Necessidade: Automatização da geração de grafos de fluxo de controle	
Funcionalidade	**Categoria**
1. Importar código-fonte e realizar parsing automático para construção do GFC	Crítico
2. Identificar blocos de comandos e decisões, criando nós e arestas do GFC	Crítico
3. Calcular cobertura estrutural: comandos, decisões, condições e caminhos	Crítico
4. Gerar relatório de elementos não cobertos, com referência a arquivos e linhas	Importante

Necessidade: Derivação de casos de teste funcionais a partir de especificações	
Funcionalidade	**Categoria**
1. Modelar causas e efeitos por meio de interface gráfica dedicada	Crítico
2. Suportar operadores lógicos (AND/OR/NOT) e restrições (E, I, O, R, M) no grafo	Crítico
3. Converter o grafo de causa-efeito em tabela de decisão	Crítico
4. Gerar automaticamente casos de teste funcionais a partir da tabela de decisão	Importante
5. Validar consistência do modelo (causas sem uso, efeitos inalcançáveis)	Útil

Necessidade: Consolidação das métricas de cobertura estrutural e funcional	
Funcionalidade	**Categoria**

1. Exibir dashboard com métricas integradas (estrutural × funcional)	Importante
2. Mapear casos de teste aos elementos cobertos do GFC (nós/arestas)	Importante
3. Destacar no grafo elementos parcialmente cobertos (mapa de calor)	Útil
4. Exportar métricas consolidadas em CSV/JSON para análise externa	Útil

Necessidade: Visualização interativa e didática das estruturas de teste	
Funcionalidade	**Categoria**
1. Fornecer layouts automáticos de grafos	Importante
2. Adicionar anotações/legendas automáticas (rótulos de nós, decisões, caminhos)	Importante
3. Permitir foco em subgrafos e filtragem por critério (ex.: somente decisões)	Útil
4. Exportar os grafos como imagem (PNG/SVG) para uso didático	Importante
5. Suporte a temas claro/escuro e ajustes de contraste/acessibilidade	Útil

INTERLIGAÇÃO COM OUTROS SISTEMAS

O sistema poderá ser integrado a repositórios de código, como GitHub ou GitLab, para facilitar a importação de trechos de programas. Além disso, haverá compatibilidade com ferramentas de exportação, permitindo que os resultados sejam compartilhados em formatos padrão que possam ser reutilizados em relatórios, artigos acadêmicos ou em pipelines de software.

RESTRIÇÕES

Restrições de produto

- **Requisitos de eficiência:**
    - A geração do **Grafo de Fluxo de Controle (GFC)** para trechos de até **500 linhas** deve ocorrer em **até 2 segundos** em ambiente de referência.

    - A construção do **Grafo de Causa-Efeito (GCE)** com até **20 causas/efeitos** deve ocorrer em **até 2 segundos** em ambiente de referência.
- **Requisitos de confiabilidade:**
    - As métricas de cobertura (comandos, decisões, condições, caminhos) devem ser apresentadas com **precisão de duas casas decimais**.
    - A geração dos grafos deve ser **determinística**: a mesma entrada resulta no mesmo grafo e mesmas métricas.
- **Requisitos de segurança e privacidade:**
    - O código colado pelo usuário é processado **apenas em memória** e **descartado por padrão**; qualquer persistência requer **consentimento explícito**.
    - Logs não devem armazenar conteúdo do código submetido; apenas metadados técnicos (timestamp, tamanho aproximado, rótulos de artefato).
- **Requisitos de usabilidade e acessibilidade:**
    - Interface **responsiva** e com **contraste adequado**, suporte a **navegação por teclado** e textos em **PT-BR**.

Restrições organizacionais

- **Requisitos de implementação:**
    - A aplicação deverá ser desenvolvida com **TypeScript**, com **Angular** no front-end e **Spring Boot** no back-end.
    - Devem ser adotadas bibliotecas **open source** com licenças compatíveis (**MIT**, **Apache-2.0** ou equivalentes) para visualização de grafos e exportação de relatórios.

Restrições externas

- **Requisitos de portabilidade:**
    - Execução em navegadores modernos (**Chrome/Edge/Firefox** versões recentes). Não há dependência de IDEs.
- **Requisitos de interoperabilidade:**
    - Exportação em **PDF, CSV e JSON**; importação via **colagem de código** e metadados fornecidos pelo usuário (projeto/artefato/unidade).
- **Requisitos legais/licenciamento:**
    - Observância às licenças de terceiros e **proibição de armazenamento** de código confidencial sem autorização do usuário.

DOCUMENTAÇÃO

Detalhamento da documentação prevista para o sistema, exemplos: manuais do usuário, help online, guia de instalação, arquivos readme.