

---

# Documentação de Projeto

para o sistema

## Sistema de Geração e Análise de Grafos de Teste

**Versão 1.0**

Projeto de sistema elaborado pelo(s) aluno(s) Lucas Cabral Soares e Maria Eduarda Amaral Muniz e apresentado ao curso de **Engenharia de Software** da **PUC Minas** como parte do Trabalho de Conclusão de Curso (TCC) sob orientação de conteúdo dos professores Danilo de Quadros Maia Filho, Leonardo Vilela Cardoso, Raphael Ramos Dias Costa, orientação acadêmica do professor Cleiton Silva Tavares e orientação de TCC II do professor (a ser definido no próximo semestre).

**25/09/2025**

# Tabela de Conteúdo

<b>Tabela de Conteúdo</b>	<b>ii</b>
<b>Histórico de Revisões</b>	<b>ii</b>
<b>1. Modelo de Requisitos</b>	<b>1</b>
1.1 Descrição de Atores	1
1.2 Modelo de Casos de Uso	1
<b>2. Modelo de Projeto</b>	<b>1</b>
2.1 Diagrama de Classes	1
2.2 Diagramas de Sequência	1
2.3 Diagramas de Comunicação	1
2.4 Arquitetura Lógica: Diagramas de Pacotes	1
2.5 Diagramas de Estados	1
2.6 Diagrama de Componentes	1
<b>3. Projeto de Interface com Usuário</b>	<b>2</b>
3.1 Interfaces Comuns a Todos os Atores	2
3.2 Interfaces Usadas pelo Ator <A>	2
3.3 Interfaces Usadas pelo Ator <B>	2
<b>4. Modelo de Dados</b>	<b>2</b>
<b>5. Modelo de Teste</b>	<b>2</b>

## Histórico de Revisões

Nome	Data	Razões para Mudança	Versão
Lucas Cabral Soares e Maria Eduarda Amaral Muniz	28/09/2025	<ul style="list-style-type: none"><li>● Criação do documento;</li><li>● Adicionando a Seção 1. Introdução;</li><li>● Adicionando a Seção 2.3. Modelo de Casos de Uso e Histórias de Usuário;</li><li>● Adicionando a Seção 2.3.1. Diagrama de Casos de Uso;</li><li>● Adicionando a Seção 2.3.2. Histórias de Usuários</li><li>● Adicionando a Seção 4. Projeto de Interface com Usuário</li><li>● Adicionando a Seção 4.1 Esboço das Interfaces Comuns a Todos os Atores</li></ul>	1.0

## 1. Introdução

Este documento agrega: 1) a elaboração e revisão de modelos de domínio e 2) modelos de projeto para o Sistema de Geração e Análise de Grafos de Teste. A referência principal para a descrição geral do problema, domínio e requisitos do sistema é o documento de especificação que descreve a visão de domínio do sistema. Tal especificação acompanha este documento. Anexo a este documento também se encontra o Glossário.

## 2. Modelos de Usuário e Requisitos

### 2.1 Descrição de Atores

Estudante de Engenharia de Software: usa a plataforma para colar pequenos trechos de código, gerar o GFC, visualizar cobertura (comandos/decisões/condições/caminhos) e modelar causas-efeitos para obter tabela de decisão e casos de teste, exporta imagens/relatórios para trabalhos.

Professor de Teste de Software: utiliza a ferramenta em demonstrações de aula para gerar GFC a partir de código colado e construir grafos de causa-efeito a partir de especificações.

Engenheiro de Software: analisa funções/métodos colando o código para obter mapa de calor do GFC, identificar lacunas de cobertura e derivar casos funcionais via tabela de decisão, exporta CSV/JSON para integração manual com a suíte de testes do projeto.

### 2.2 Modelos de Usuários

Esta subseção tem como objetivo descrever os modelos de usuários desenvolvidos por meio da implementação de personas. Para a construção das personas, foram considerados os diferentes perfis que utilizarão a plataforma: estudantes de Engenharia de Software, professores universitários de teste de software e engenheiros de software profissionais.

A Tabela 1 descreve a persona do usuário Julia Alves Silva, uma estudante de Engenharia de Software. Nela é possível perceber que, apesar de interessada no aprendizado, ela encontra dificuldades em compreender os conceitos teóricos de grafos quando apresentados apenas em aula expositiva. Por isso, deseja contar com uma ferramenta interativa que permita visualizar e manipular os grafos de forma prática, associando-os diretamente à derivação de casos de teste.

<b>Julia Alves Silva</b>	
<b>Descrição</b>	Julia tem 21 anos, cursa o quinto período de Engenharia de Software em Belo Horizonte e está atualmente matriculada na disciplina de Teste de Software. Apesar de interessada pela área, sente dificuldade em compreender conceitos abstratos de cobertura estrutural apenas a partir de slides. Ela vê na plataforma uma forma de praticar a geração de grafos de fluxo de controle e de causa-efeito, relacionando-os com casos de teste concretos.
<b>Dores</b>	<ul style="list-style-type: none"> <li>• Dificuldade em visualizar conceitos abstratos apenas em teoria.</li> <li>• Falta de ferramentas práticas que auxiliem no aprendizado em sala.</li> </ul>
<b>Objetivos</b>	<ul style="list-style-type: none"> <li>• Praticar a geração de grafos de fluxo de controle a partir de pequenos trechos de código.</li> <li>• Entender a derivação de casos de teste funcionais por meio de grafos de causa-efeito.</li> </ul>

Tabela 1. Persona Julia Alves Silva

A Tabela 2 descreve a persona do usuário Carlos Eduardo Menezes, um engenheiro de software profissional. Sua descrição mostra que ele tem experiência prática com testes, mas frequentemente precisa analisar grandes volumes de código e validar a qualidade dos testes existentes. Ele sente como dor a fragmentação entre ferramentas de cobertura estrutural e funcional, o que dificulta a visão integrada da qualidade de um sistema.

<b>Carlos Eduardo</b>	
<b>Descrição</b>	Carlos tem 34 anos, atua como engenheiro de software em uma empresa de desenvolvimento de sistemas corporativos. Trabalha diariamente com testes unitários e de integração, utilizando ferramentas de cobertura como JaCoCo. No entanto, enfrenta dificuldade para conectar métricas de cobertura estrutural com derivação de testes funcionais, precisando recorrer a diferentes ferramentas e relatórios manuais.
<b>Dores</b>	<ul style="list-style-type: none"> <li>• Necessidade de alternar entre ferramentas para avaliar cobertura estrutural e funcional.</li> <li>• Dificuldade em identificar rapidamente lacunas nos testes implementados.</li> </ul>

<b>Objetivos</b>	<ul style="list-style-type: none"><li>● Importar código do projeto e gerar automaticamente o grafo de fluxo de controle.</li><li>● Avaliar cobertura de testes já existentes e identificar pontos não exercitados.</li><li>● Derivar casos de teste funcionais a partir de requisitos.</li></ul>
------------------	--

*Tabela 2. Persona Carlos Eduardo*

A Tabela 3 descreve a persona do usuário Mariana Rocha Almeida, professora universitária de Teste de Software. Sua rotina exige o preparo de aulas, exercícios e exemplos práticos. Ela sente como dor a ausência de ferramentas didáticas que integrem conceitos teóricos e visuais, capazes de apoiar o ensino e a avaliação dos estudantes em disciplinas de teste

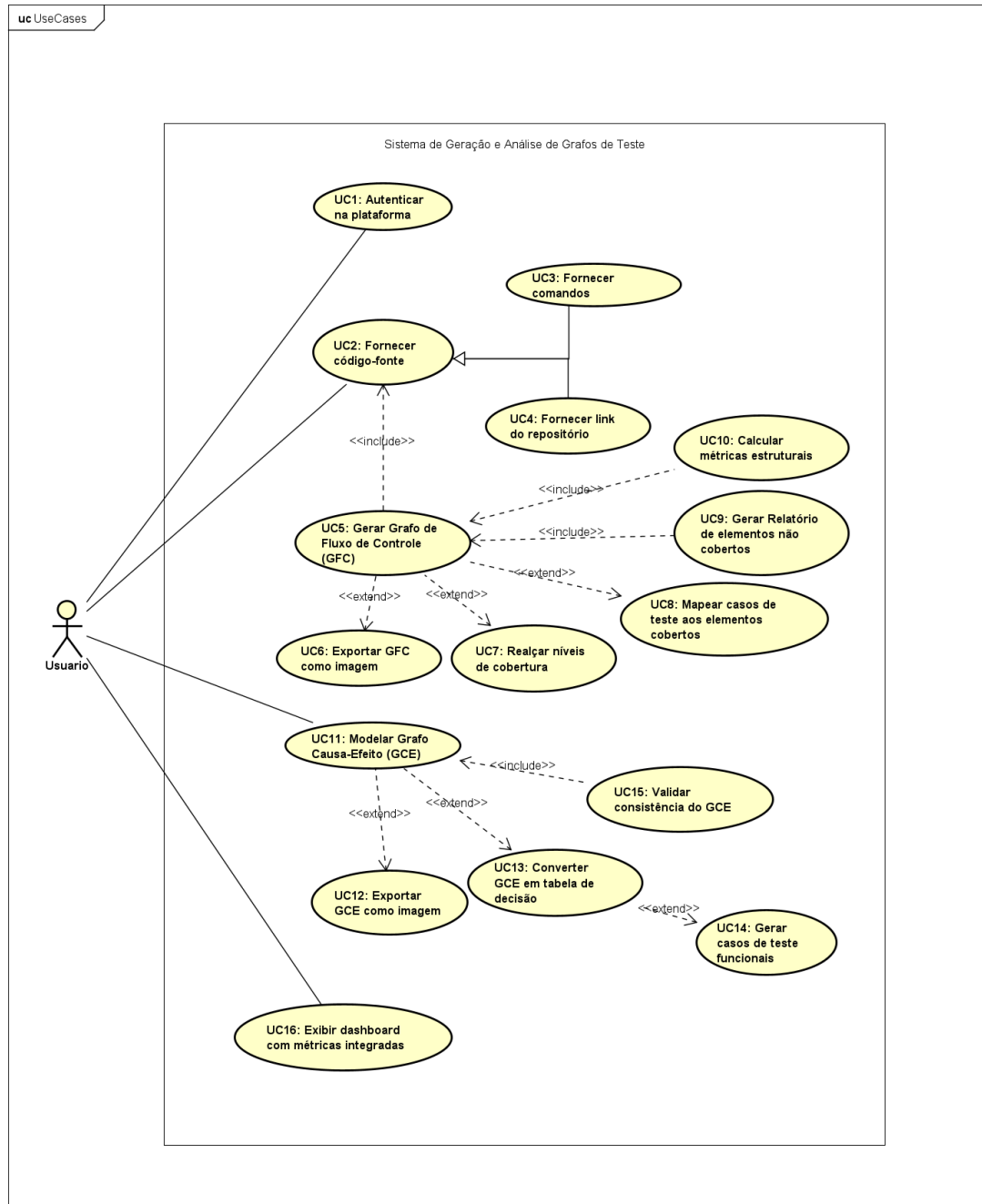
<b>Mariana Rocha Almeida</b>	
<b>Descrição</b>	Mariana tem 42 anos, é doutora em Ciência da Computação e leciona disciplinas de Qualidade e Teste de Software em uma universidade. Costuma preparar exercícios com base em exemplos de código simples e em especificações de sistemas. Entretanto, a construção manual de grafos e tabelas de decisão é trabalhosa e nem sempre engaja os alunos.
<b>Dores</b>	<ul style="list-style-type: none"><li>● Esforço elevado para criar exemplos manuais de grafos e tabelas de decisão.</li><li>● Falta de ferramentas que ilustrem de forma clara e didática o processo de geração de casos de teste.</li></ul>
<b>Objetivos</b>	<ul style="list-style-type: none"><li>● Utilizar a plataforma em sala de aula para demonstrar a geração de grafos e casos de teste.</li><li>● Explorar visualmente a relação entre requisitos, fluxos de controle e cobertura estrutural/funcional.</li></ul>

*Tabela 3. Persona Mariana Rocha Almeida*

## 2.3 Modelo de Casos de Uso e Histórias de Usuários

Esta subseção tem como objetivo descrever os casos de uso e histórias de usuário previstos para o projeto. Para isso é apresentado um diagrama de casos de uso onde todos são listados. Dessa mesma forma, também são apresentadas as histórias de usuário relacionadas às funcionalidades previstas para o sistema a ser desenvolvido.

### 2.3.1 Diagrama de Casos de Uso



### **2.3.2 Histórias de usuário**

Nesta seção são listadas as histórias de usuários elicitadas para o sistema proposto. Para fins de organização, utiliza-se identificadores no formato US#ID, em que US refere-se a User Story. Assim, as histórias de usuário identificadas para o sistema são:

US1. Como usuário, eu gostaria de gerar automaticamente um Grafo de Fluxo de Controle (GFC) a partir do código-fonte de modo que eu reduza o esforço manual e minimize erros na modelagem.

US2. Como usuário, eu gostaria de importar código diretamente de repositórios GitHub/GitLab de modo que eu agilize a geração do GFC sem precisar copiar trechos manualmente.

US3. Como usuário, eu gostaria de visualizar métricas de cobertura de comandos, decisões, condições e caminhos de modo que eu avalie a eficácia dos testes existentes no meu projeto.

US4. Como usuário, eu gostaria de gerar relatórios de elementos não cobertos no código (arquivo/linha) de modo que eu identifique lacunas que precisam de novos testes.

US5. Como usuário, eu gostaria de modelar graficamente causas e efeitos a partir da especificação de modo que eu tenha uma representação clara para derivar casos de teste funcionais.

US6. Como usuário, eu gostaria de aplicar operadores (AND, OR, NOT) e restrições (E, I, O, R, M) no Grafo de Causa-Efeito (GCE) de modo que eu capture regras complexas de negócio.

US7. Como usuário, eu gostaria de validar a consistência do GCE (causas não utilizadas, efeitos inalcançáveis) de modo que eu garanta a qualidade do modelo funcional.

US8. Como usuário, eu gostaria de converter o GCE em tabela de decisão de modo que eu obtenha uma visão sistemática para derivar casos de teste funcionais.

US9. Como usuário, eu gostaria de gerar automaticamente casos de teste funcionais a partir da tabela de decisão de modo que eu reduza erros e aumente a cobertura funcional.

US10. Como usuário, eu gostaria de consolidar métricas estruturais (GFC) e funcionais (GCE) em um dashboard de modo que eu tenha uma visão integrada da qualidade de testes.

US11. Como usuário, eu gostaria de mapear casos de teste aos elementos do GFC (nós/arestas) de modo que eu consiga verificar a relação entre cobertura estrutural e funcional.

US12. Como usuário, eu gostaria de visualizar no grafo elementos parcialmente cobertos por meio de heatmap de modo que eu identifique rapidamente áreas de baixa cobertura.

US13. Como usuário, eu gostaria de visualizar grafos com layouts automáticos, rótulos, anotações e filtragem de subgrafos de modo que eu compreenda melhor a estrutura e facilite o uso didático.

US14. Como usuário, eu gostaria de exportar grafos como imagens (PNG/SVG) de modo que eu os utilize em relatórios, artigos ou aulas.

US15. Como usuário, eu gostaria de visualizar um dashboard com métricas integradas (estrutural x funcional) de modo que eu acompanhe a qualidade dos testes em uma única tela.

## 2.4 Diagrama de Sequência do Sistema e Contrato de Operações

Nesta subseção é apresentado o diagrama de sequência do sistema e os Contratos de Operações.

Formato para cada contrato de operação

<b>Contrato</b>	
<b>Operação</b>	
<b>Referências cruzadas</b>	
<b>Pré-condições</b>	
<b>Pós-condições</b>	

## 3. Modelos de Projeto

### 3.1 Diagrama de Classes

Diagrama de classes do sistema

### 3.2 Diagramas de Sequência

Diagramas de sequência para realização de casos de uso.

### 3.3 Diagramas de Comunicação

Diagramas de comunicação para realização de casos de uso.

### 3.4 Arquitetura

Pode ser descrita com um diagrama apropriado da UML ou C4 Model



### 3.5 Diagramas de Estados

Diagramas de estados do sistema.

### 3.6 Diagrama de Componentes e Implantação.

Diagramas de componentes do sistema. Diagrama de implantação mostrando onde os componentes estarão alocados para a execução.

## 4. Projeto de Interface com Usuário

### 4.1 Esboço das Interfaces Comuns a Todos os Atores

A Figura 1 representa a tela de Dashboard, exibida como ponto de entrada da aplicação. Nela o usuário pode colar trechos de código diretamente na área de entrada para posterior análise. A barra lateral esquerda contém a navegação principal da plataforma, com as opções: Dashboard, GFC, GCE, Relatórios e Tabela de decisão. A presença dessa navegação fixa garante acesso rápido às funcionalidades principais, independentemente do perfil do usuário.

Na parte superior direita da tela encontra-se o campo de pesquisa, permitindo ao usuário localizar rapidamente artefatos ou projetos já analisados. Ao lado do campo de entrada de código, estão presentes os botões “Gerar GFC” e “Novo GCE”, que dão acesso direto às operações mais frequentes: geração de Grafo de Fluxo de Controle e construção de Grafo de Causa-Efeito.

A seção central do Dashboard exibe os resultados gráficos mais recentes. O Grafo de Fluxo de Controle aparece com destaque visual, ao lado do trecho de código correspondente, ressaltando a associação direta entre o artefato de entrada e a representação gráfica. Logo abaixo, é apresentado o Grafo de Causa-Efeito mais recentemente modelado, permitindo ao usuário visualizar a relação entre causas, operadores lógicos e efeitos derivados.

Por fim, a parte inferior da tela mostra cartões de artefatos já processados. Cada cartão contém informações como nome do artefato, data de geração e métricas de cobertura (por exemplo, comandos e decisões). Esse componente atende tanto alunos que precisam visualizar rapidamente seus exercícios quanto engenheiros que desejam acompanhar a evolução da cobertura em diferentes versões do código.

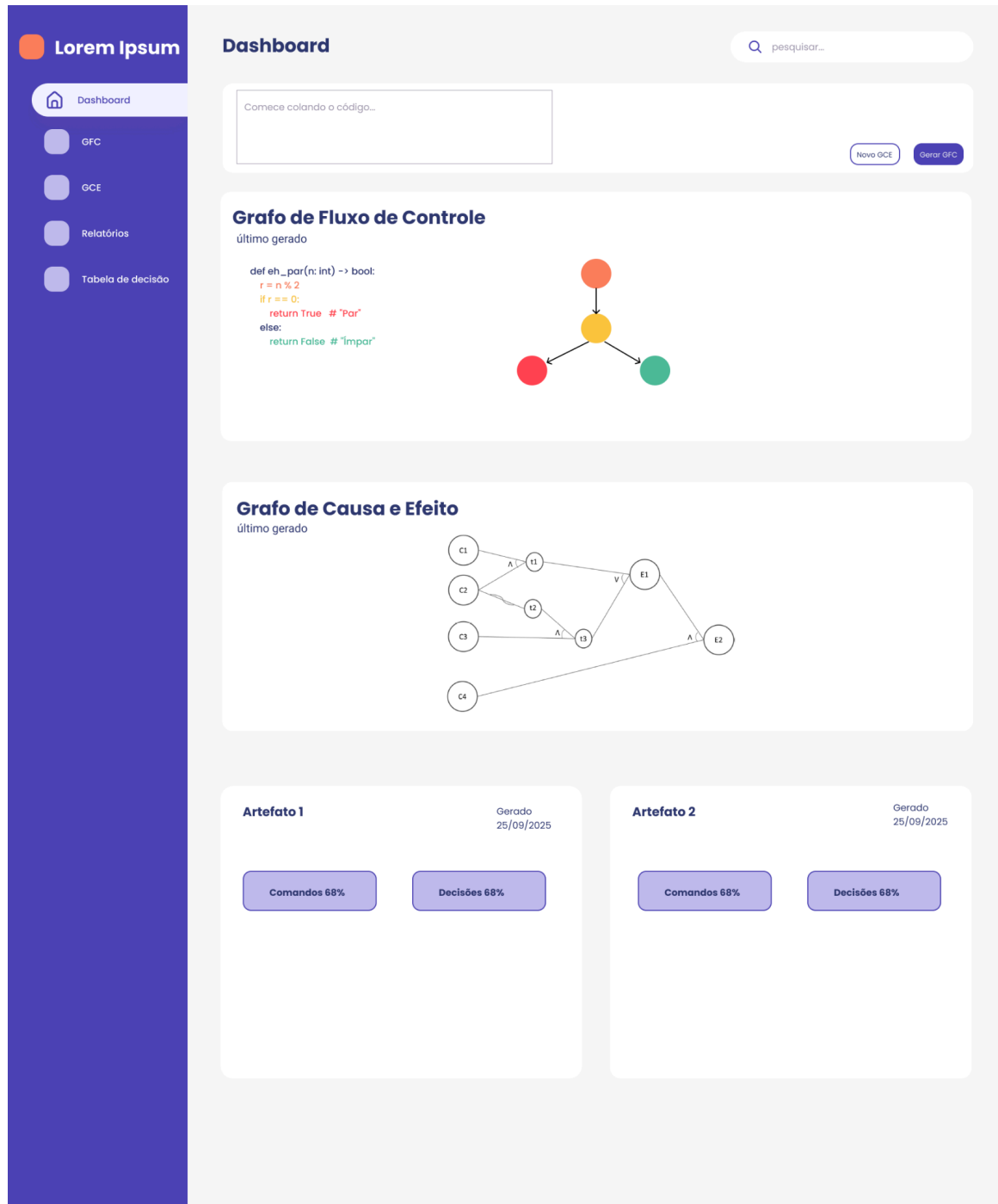


Figura 1. Protótipo da Tela de Dashboard da Plataforma

## 4.2 Esboço das Interfaces Usadas pelo Ator <A>

Wireframe/mockup/storyboard das interfaces exclusivas do ator <A>

### **4.3 Esboço das Interfaces Usadas pelo Ator <B>**

Wireframe/mockup/storyboard das interfaces exclusivas do ator <B>

## **5. Glossário e Modelos de Dados**

Deve-se apresentar o glossário para o sistema. Também apresente esquemas de banco de dados e as estratégias de mapeamento entre as representações de objetos e não-objetos.

## **6. Casos de Teste**

Uma descrição de casos de teste para validação do sistema.

## **7. Cronograma e Processo de Implementação**

Uma descrição do cronograma para implementação do sistema e do processo que será seguido durante a implementação.