
Documentação de Projeto

para o sistema

Plataforma de Apoio às Avaliações de Projetos GitHub

Versão 1.0

Projeto de sistema elaborado pelo(s) aluno(s) Guilherme Gabriel Silva Pereira e Lucas Ângelo Oliveira Martins Rocha e apresentado ao curso de **Engenharia de Software** da **PUC Minas** como parte do Trabalho de Conclusão de Curso (TCC) sob orientação de conteúdo do professor José Laerte Pires Xavier Junior, orientação acadêmica do professor Cleiton Silva Tavares e orientação de TCC II do professor (a ser definido no próximo semestre).

5 de março de 2023.

Tabela de Conteúdo

Tabela de Conteúdo	ii
Histórico de Revisões	ii
1. Modelo de Requisitos	1
1.1 Descrição de Atores	1
1.2 Modelo de Casos de Uso	1
2. Modelo de Projeto	1
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
3. Projeto de Interface com Usuário	2
3.1 Interfaces Comuns a Todos os Atores	2
3.2 Interfaces Usadas pelo Ator <A>	2
3.3 Interfaces Usadas pelo Ator 	2
4. Modelo de Dados	2
5. Modelo de Teste	2

Histórico de Revisões

Nome	Data	Razões para Mudança	Versão
Criação do documento.	05/03/2023	Criação do documento, atualização do título e datas, escrita da seção 1. Introdução, adição do diagrama de casos de uso e históricos de usuário.	1.0

1. Introdução

Este documento agrega: 1) a elaboração e revisão de modelos de domínio e 2) modelos de projeto para o sistema Plataforma de Apoio às Avaliações de Projetos GitHub. 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.

O sistema proposto neste trabalho consiste em uma plataforma web de apoio às avaliações de trabalhos no GitHub que será utilizado pelos professores das disciplinas de “Trabalho Interdisciplinar”. Nessa plataforma os docentes poderão cadastrar métodos avaliativos para cada oferta de disciplina e visualizar as informações resultantes dos repositórios dos trabalhos conforme o método avaliativo selecionado. A necessidade de uma plataforma com essas funções origina-se da carência de nenhuma outra aplicação que auxilie docentes a avaliarem repositórios de código e os artefatos de documentação de trabalhos, por meio de filtros temporais e por integrantes.

O fluxo de funcionamento da plataforma iniciará com o cadastro da oferta de uma disciplina em um semestre específico, com isso, o professor dessa disciplina irá selecionar são os repositórios dessa oferta. A partir disso, deverá ser cadastrado um método avaliativo para essa disciplina, informando quais são os arquivos e seus respectivos diretórios onde deverão estar criados e preenchidos. Por exemplo, na plataforma deverá ser cadastrada a oferta da disciplina de “Trabalho Interdisciplinar 5: Aplicações Distribuídas 1/2023” e seu método avaliativo contendo uma regra de consistência que exija a existência do arquivo “Documentacao/documento_de_arquitetura.md” contendo no mínimo 750 caracteres, a partir disso, todos os repositórios dessa oferta de disciplina verificarão essa regra. Quando essa regra não for seguida por algum repositório de um trabalho com esse método avaliativo, haverá uma opção para abrir uma *issue* padronizada no repositório respectivo.

Em relação ao controle temporal, também poderão ser cadastradas sprints para ofertas da disciplina, o que possibilitará avaliar contribuições de alunos em trabalhos em diferentes períodos temporais. Seguindo o mesmo exemplo da oferta da disciplina de “Trabalho Interdisciplinar 5: Aplicações Distribuídas 1/2023”, poderá ser cadastradas seis sprints para esse primeiro semestre de 2023, cada sprint com períodos pré-definidos de tempo, o que poderá ser utilizado para filtrar contribuições de integrantes em trabalhos para cada uma dessas seis sprints. Por exemplo, para regra de consistência que exija a existência do arquivo “Documentacao/documento_de_arquitetura.md”, poderá ser associada uma sprint que determinará que até a data final da sprint esse documento deverá estar criado ou abrirá uma *issue* padronizada.

A partir da coleta dos dados dos repositórios do GitHub, resultados das validações das regras de consistências dos métodos avaliativos, contribuições de arquivos, linhas de código, *issues* e qualidade de código, a plataforma informará as contribuições no trabalho, por aluno e prazos de sprints para cada trabalho. Além disso, apresentará uma visualização de histórico de *commits* com filtro para alunos e sprint. Ademais, para cada repositório será apresentada a quantidade de *issues* abertas e fechadas por cada integrante de cada trabalho, também sendo possível filtrar essa informação por sprint.

Esta plataforma contará com uma opção para efetuar uma inspeção da qualidade dos códigos SonarQube para cada repositório, essa inspeção calculará a qualidade do código do trabalho. Outrossim, com objetivo de facilitar o uso por meio dos docentes, o sistema de autenticação e

autorização utilizará o OAuth do próprio GitHub. Dessa forma, os docentes serão auxiliados na avaliação qualitativa e quantitativa das entregas de artefatos, documentação e qualidade de código dos trabalhos desenvolvidos pelas equipes de alunos.

2. Modelos de Usuário e Requisitos

2.1 Descrição de Atores

Nesta subseção é apresentado descrição de cada um dos atores que interagem com o sistema.

2.2 Modelos de Usuários

Apresentação de personas, perfis ou mapas de empatia.

2.3 Modelo de Casos de Uso e Histórias de Usuários

2.3.1 Casos de Uso

A Figura 1 ilustra o Diagrama dos Casos de Uso do sistema. Nesse diagrama é visível 2 atores da plataforma, sendo o ator Professor e o ator Scheduler que é um usuário de sistema. O ator Professor representa os docentes que acessaram o sistema para efetuar as operações básicas de cadastro, atualização, deleção e visualização das funcionalidades de trabalhos, métodos avaliativos, regras de consistência, análise estática de código e *issues* padronizadas. Já o usuário de sistema Scheduler será responsável por executar tarefas cronometradas de busca e atualização da base de dados do sistema por meio da Interface de Programação de Aplicação (API, do inglês Application Programming Interface) do GitHub.

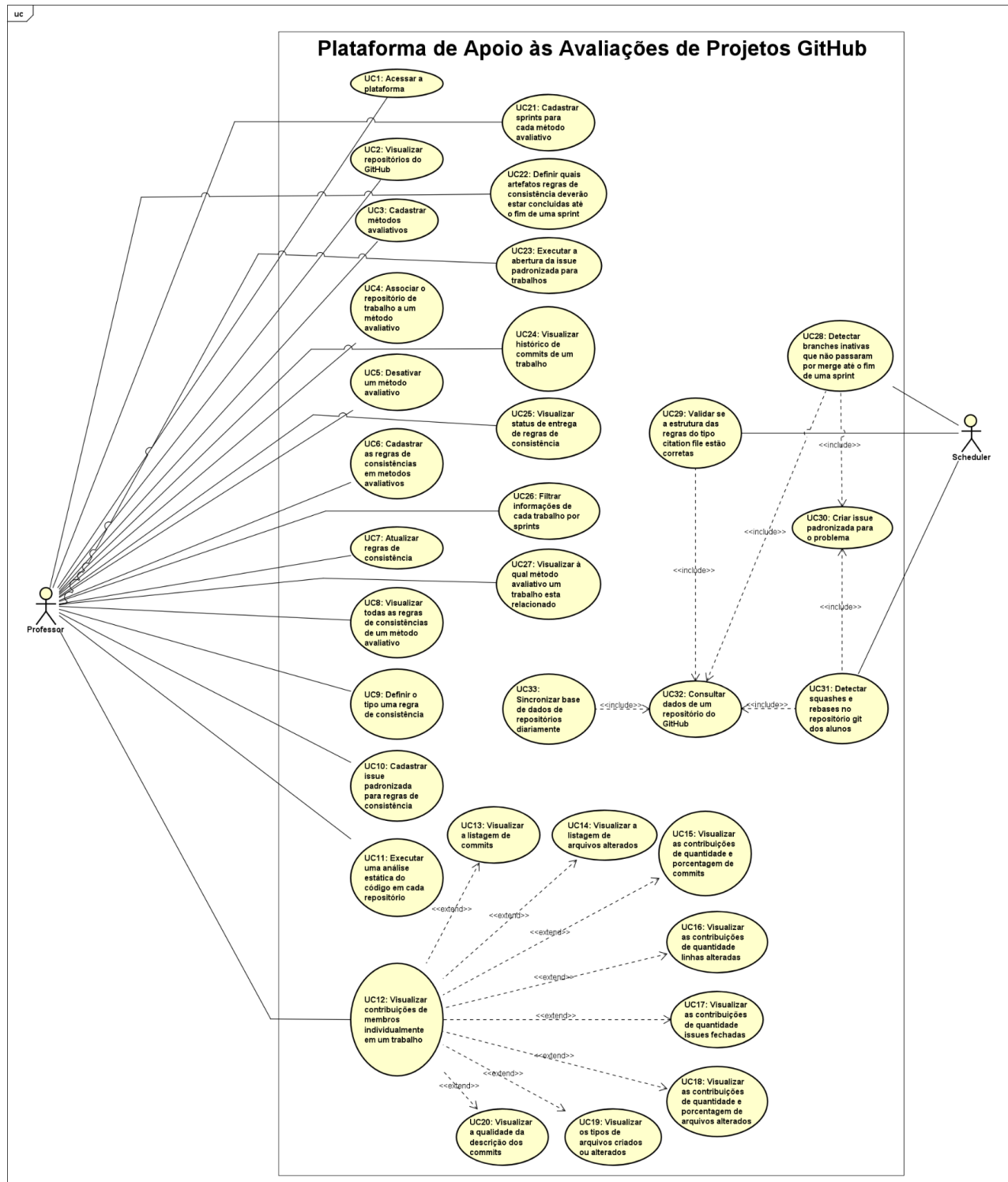


Figura 1. Diagrama de Caso de Uso

2.3.1 Histórias de Usuários

A seguir são listadas todas as histórias de usuários do sistema.

- US01. Como professor, desejo me autenticar usando a conta do Github, para usar as funcionalidades do sistema;
- US02. Como professor, desejo ver os repositórios que sou responsável, para poder avaliá-los;
- US03. Como professor, desejo vincular um repositório, que sou responsável, a um método avaliativo, para aquele repositório seja avaliado de acordo com aquele método;
- US04. Como professor, desejo criar um método avaliativo para vincular aos repositórios que avaliarei;
- US05. Como professor, desejo inativar um método avaliativo obsoleto para que ele não possa mais ser vinculado a novos repositórios;
- US06. Como professor, desejo criar uma regra de consistência de um artefato para que ele possa ser vinculado a um método avaliativo;
- US07. Como professor, desejo criar uma regra de consistência do conteúdo de um artefato para que ele seja vinculado a um método avaliativo;
- US08. Como professor, desejo vincular regras de consistência (pré-definidas ou criadas, do conteúdo ou não) a um método avaliativo, para que ele seja usado na avaliação dos repositórios vinculados;
- US09. Como professor, desejo editar as regras de consistência para que eu possa corrigir informações anteriormente cadastradas;
- US10. Como professor, desejo ver todas as regras de consistências cadastradas para um método avaliativo, para que eu possa saber como o conjunto de repositórios vinculados será avaliado;
- US11. Como professor, desejo definir as sprints de um método avaliativo, para agrupar as entregas dos repositórios vinculados;
- US12. Como professor, desejo ver quais as regras de consistência relacionadas a uma sprint, para poder saber o que deve ser entregue pelos alunos.
- US13. Como professor, desejo ver quais regras de consistência foram cumpridas no prazo da sprint estipulados em um determinado repositório, para usar essa informação para o cálculo da nota do grupo;
- US14. Como professor, desejo ver quais são os contribuidores do repositório, para descobrir os alunos que fazem parte daquele grupo;
- US15. Como professor, desejo conferir os indicadores de contribuição (*commits*, linhas alteradas, *issues* fechadas) de um repositório, para poder avaliar o empenho geral do grupo;
- US16. Como professor, desejo filtrar os indicadores de contribuição por contribuidor, para poder avaliar o empenho geral do integrante do grupo;
- US17. Como professor, desejo conferir os indicadores de contribuição relativos por contribuidor (% de *commits*, % de linhas alteradas, % de *issues* fechadas), para poder avaliar o empenho relativo do integrante do grupo;
- US18. Como professor, desejo filtrar os indicadores de contribuição por sprint, para poder avaliar o empenho naquela entrega;
- US19. Como professor, desejo ver quais os tipos de arquivos foram alterados por cada integrante, em quais sprints, para avaliar ter uma visão mais específica do tipo contribuição que o integrante está fazendo;
- US20. Como professor, desejo ver quais arquivos foram alterados por cada integrante, em quais sprints, para avaliar ter uma visão mais específica do tipo contribuição que o integrante está fazendo;

US21. Como professor, desejo disparar a execução de uma ferramenta de análise estática de código (SonarQube) no código do repositório, para avaliar a qualidade do código que está sendo produzidos pelos alunos;

US23. Como professor, desejo ver os resultados da execução de uma ferramenta de análise estática de código (SonarQube) no código do repositório, para avaliar a qualidade do código que está sendo produzidos pelos alunos;

US24. Como professor, desejo criar um *template* de *issue* em um método avaliativo para uma determinada regra de consistência, para ser usado na criação de *issues* nos repositórios;

US25. Como professor, desejo executar a criação de *issues* de *template*, para os alunos serem alertados de uma pendência na entrega;

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

Contrato	
Operação	
Referências cruzadas	
Pré-condições	
Pós-condições	

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

Wireframe/mockup/storyboard das interfaces que são comuns a todos os atores do sistema.

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 **

Wireframe/mockup/storyboard das interfaces exclusivas do ator

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.