
Documentação de Projeto

para o sistema

Plataforma de Apoio às Avaliações de Projetos GitHub

Versão 1.3

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
1. Criação.	05/03/2023	<ul style="list-style-type: none"> • Criação do documento; • Adicionando título e datas; • Escrita da Seção 1. Introdução; • Adição do Diagrama de Casos de Uso; • Adição das histórias de usuário. 	1.0
2. Correções.	10/03/2023	<ul style="list-style-type: none"> • Corrigindo Diagrama de Casos de Uso, com adição dos atores secundários GitHub e Aluno; • Removendo verbos no futuro. 	1.1
3. Adições.	13/03/2023	<ul style="list-style-type: none"> • Adicionando Interfaces de Usuário. 	1.2
4. Adições	14/03/2023	<ul style="list-style-type: none"> • Adicionando as Seções 2, 2.1, 2.2 e 2.3 	1.3

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 professores 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 professores 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 inicia com o cadastro da oferta de uma disciplina em um semestre específico, com isso, o professor dessa disciplina deve selecionar quais são os repositórios dessa oferta. A partir disso, deve ser cadastrado um método avaliativo para essa disciplina, informando quais são os arquivos e seus respectivos diretórios onde devem estar criados e preenchidos. Por exemplo, na plataforma deve 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 verificam essa regra. Quando essa regra não for seguida por algum repositório de um trabalho com esse método avaliativo, é possível utilizar a opção de abrir uma *issue* padronizada no respectivo repositório.

Em relação ao controle temporal, também pode-se cadastrar sprints para ofertas da disciplina, o que possibilita 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”, pode ser cadastrada seis sprints para esse primeiro semestre de 2023, cada sprint com períodos pré-definidos de tempo, o que pode 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”, pode ser associada uma sprint que determina 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 informa as contribuições no trabalho, por aluno e prazos de sprints para cada trabalho. Além disso, apresenta uma visualização de histórico de *commits* com filtro para alunos e sprint. Ademais, para cada repositório é apresentada a quantidade de *issues* abertas e fechadas por cada integrante de cada trabalho, também sendo possível filtrar essa informação por sprint.

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

autorização utiliza o OAuth do próprio GitHub. Dessa forma, os professores são 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

Nesta seção são apresentados os modelos de usuário e requisitos do sistema. Mais especificamente, a Seção 2.1 descreve os atores do sistema, a Seção 2.2 discorre sobre o modelo de usuário, a Seção 2.3 sobre os casos de uso e histórias de usuário e a Seção 2.4, por fim, apresenta o diagrama de sequência do sistema e o contrato de operações.

2.1 Descrição de Atores

Sobre os atores que interagem com a plataforma, pode-se dividi-los em primários e secundários, O primário é aqueles que interage diretamente com a plataforma, enquanto os secundários são aqueles que impactam ou são impactados indiretamente pelas ações realizadas dentro da plataforma. Portanto, pode-se definir que os atores da plataforma são os professores de disciplinas de trabalho interdisciplinar, os alunos das mesmas e o GitHub. A seguir são detalhados, individualmente, seu papel na plataforma.

O professor é o ator primário da plataforma, ou seja, o usuário que o utiliza diretamente. Ele é responsável por lecionar as disciplinas de trabalho interdisciplinar e, conseqüentemente, avaliar os repositórios dos alunos que cursam a disciplina. Seu objetivo principal na plataforma é acompanhar as entregas dos alunos.

O GitHub é um ator secundário do sistema. Ele é o repositório de dados que armazena os trabalhos dos alunos que cursam a disciplina. O próprio GitHub também é impactado pelo sistema, com a criação de *issues* a partir da detecção de erros nos repositórios dos alunos na plataforma.

O aluno é um ator secundário, impactado indiretamente pelas ações da plataforma. Eles cursam as disciplinas de trabalho interdisciplinar e realizam as entregas através do GitHub, sendo avaliados pelos professores. Além de alimentar os dados usados nas avaliações dentro da plataforma, ele recebe *feedbacks* através da criação de *issues* no Github.

2.2 Modelos de Usuários

Nesta subseção são apresentados os modelos de usuários do sistema. O modelo foi construído usando a estratégia de personas, no qual são especificados a biografia, personalidade, necessidades e frustrações. As personas são descrições de pessoas fictícias que representam o público alvo do sistema, apresentadas nas Figuras 1 e 2.

Vanessa



IDADE	28
FORMAÇÃO	Mestre
CARGO	Professora
LOCALIZAÇÃO	Belo Horizonte

Personalidade

Entusiasta

Curiosa

Bio

Residente de Belo Horizonte, possui um mestrado na área de engenharia de software. Há menos de 2 anos faz parte do corpo docente de Engenharia de Software na PUC Minas, lecionando a disciplina de Trabalho Interdisciplinar IV.

Necessidades

- Precisa acompanhar e avaliar as entregas dos alunos na disciplina que leciona
- Continuamente aprimorar sua didática

Frustrações

- A tarefa de avaliar entregas é árdua e manual;
- O tempo gasto para coferir manualmente as entregas poderia ser usado para estudar estratégias para melhorar sua didática

Figura 1. Persona que representa um professor



Figura 2. Persona que representa um professor e coordenador

2.3 Modelo de Casos de Uso e Histórias de Usuários

Esta subseção descreve os casos de uso e histórias de usuário que contemplam as possíveis ações de usuários da plataforma desenvolvida neste projeto. Para descrever os casos de uso, foi criado o Diagrama de Caso de Uso, o qual é responsável por descrever as interações entre a plataforma deste projeto e seus usuários ou outros sistemas, ele mostra as ações que um usuário ou sistema pode realizar em relação à plataforma em questão. As histórias de usuário tem objetivo de apresentar as necessidades do usuário em relação à plataforma de uma maneira clara e concisa para facilitar o entendimento das funcionalidades.

2.3.1 Casos de Uso

A Figura 3 ilustra o Diagrama dos Casos de Uso do sistema. Nesse diagrama é visível 3 atores da plataforma, sendo o ator primário Professor e os atores secundários GitHub e Aluno. O ator Professor representa os professores que acessam 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 ator secundário GitHub é utilizado pela execução de 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, também é utilizado para autenticação de usuários. Por fim, o ator Aluno é utilizado para análises de contribuições individuais nos repositórios de trabalhos.

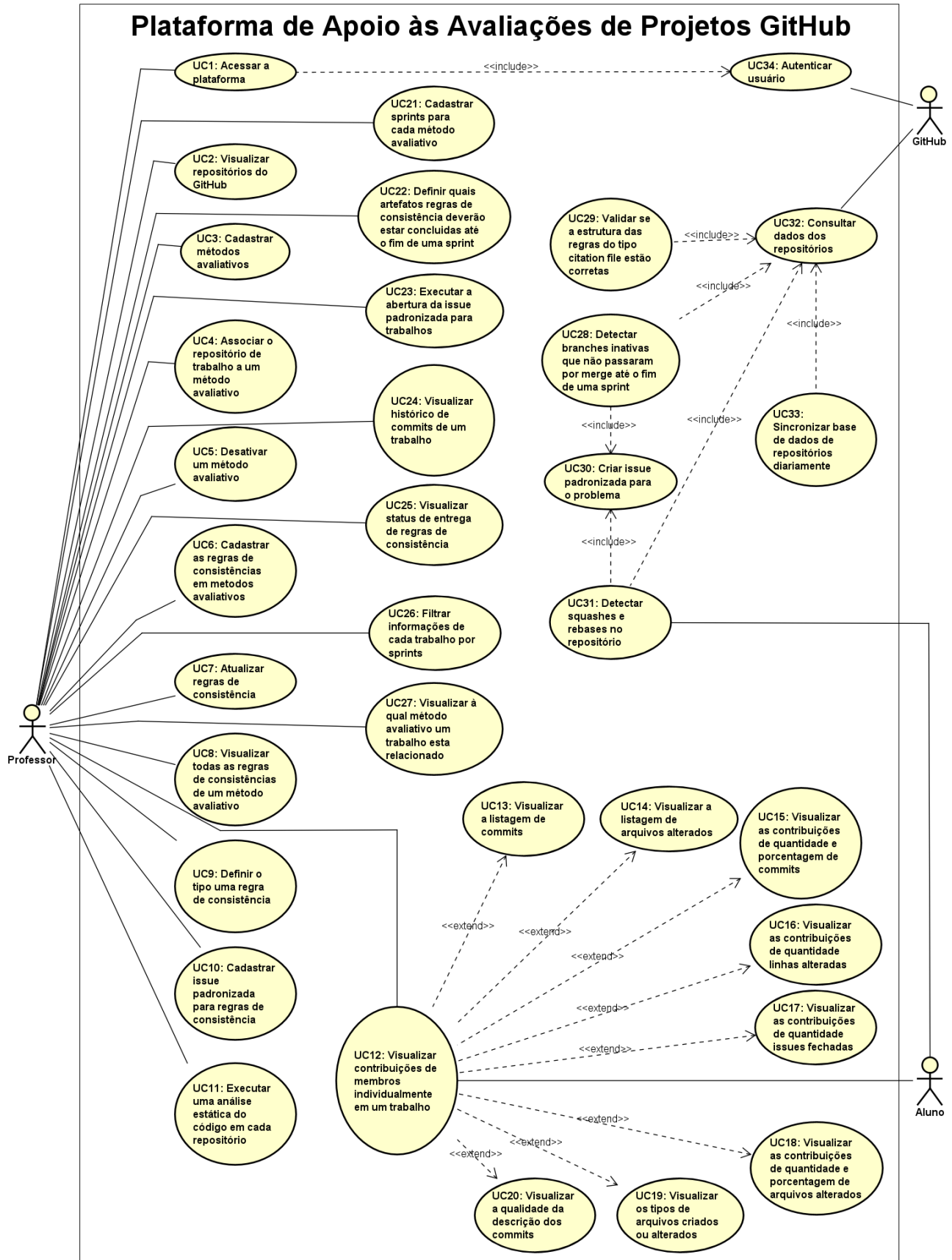


Figura 3. Diagrama de Caso de Uso

2.3.2 Histórias de Usuários

As histórias de usuário (US, do inglês *User Stories*) são uma forma de representar os requisitos do cliente. Se trata de uma descrição simples, que informa quem irá fazer uma determinada ação, qual será essa ação, e a razão que ela é realizada. A seguir, são descritas as histórias de usuário levantadas para a plataforma, identificadas pela sigla US e uma numeração.

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 criar um método avaliativo para vincular aos repositórios que avaliarei;

US04. Como professor, desejo vincular um repositório, que sou responsável, a um método avaliativo, para aquele repositório seja avaliado de acordo com aquele método;

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

Esta seção visa apresentar e descrever as principais interfaces de usuário da plataforma desenvolvida neste trabalho. Para isso, foram desenvolvidos *mockups* de alta fidelidade por meio da ferramenta Figma e as bibliotecas de *design* Primer e ChartJs. Essas bibliotecas foram escolhidas, por permitirem a criação de elementos gráficos semelhantes ao GitHub. Diante disso, as interfaces foram projetadas relacionando-as com os casos de uso definidos na Seção 2.3.1 com objetivo de mapear todas as funcionalidades necessárias para cumprimento dos requisitos especificados.

A Figura 4 representa a página inicial da plataforma que será aberta para os professores logo após a autenticação do usuário. Essa e todas as páginas do sistema possuem um *header* padrão com o ícone da plataforma, uma barra de pesquisa para buscar repositório de um trabalho pelo nome e os botões para visualizar a lista de repositórios de trabalhos e lista de métodos avaliativos, também possui um *footer* padrão com o link do repositório do GitHub deste TCC. Além disso, nessa página são listados os repositórios pertencentes a organização ICEI-PUC-Minas-PPLES-TI do GitHub, possibilitando os professores visualizarem os nomes dos repositórios e dos seus respectivos métodos avaliativos. Por fim, também há os botões de Sincronizar para atualizar a base de dado de cada repositório com do GitHub e o Abrir que possibilitar abrir a visão geral específica do repositório de um trabalho. Essa interface contempla os casos de uso UC2, UC4 e UC27.

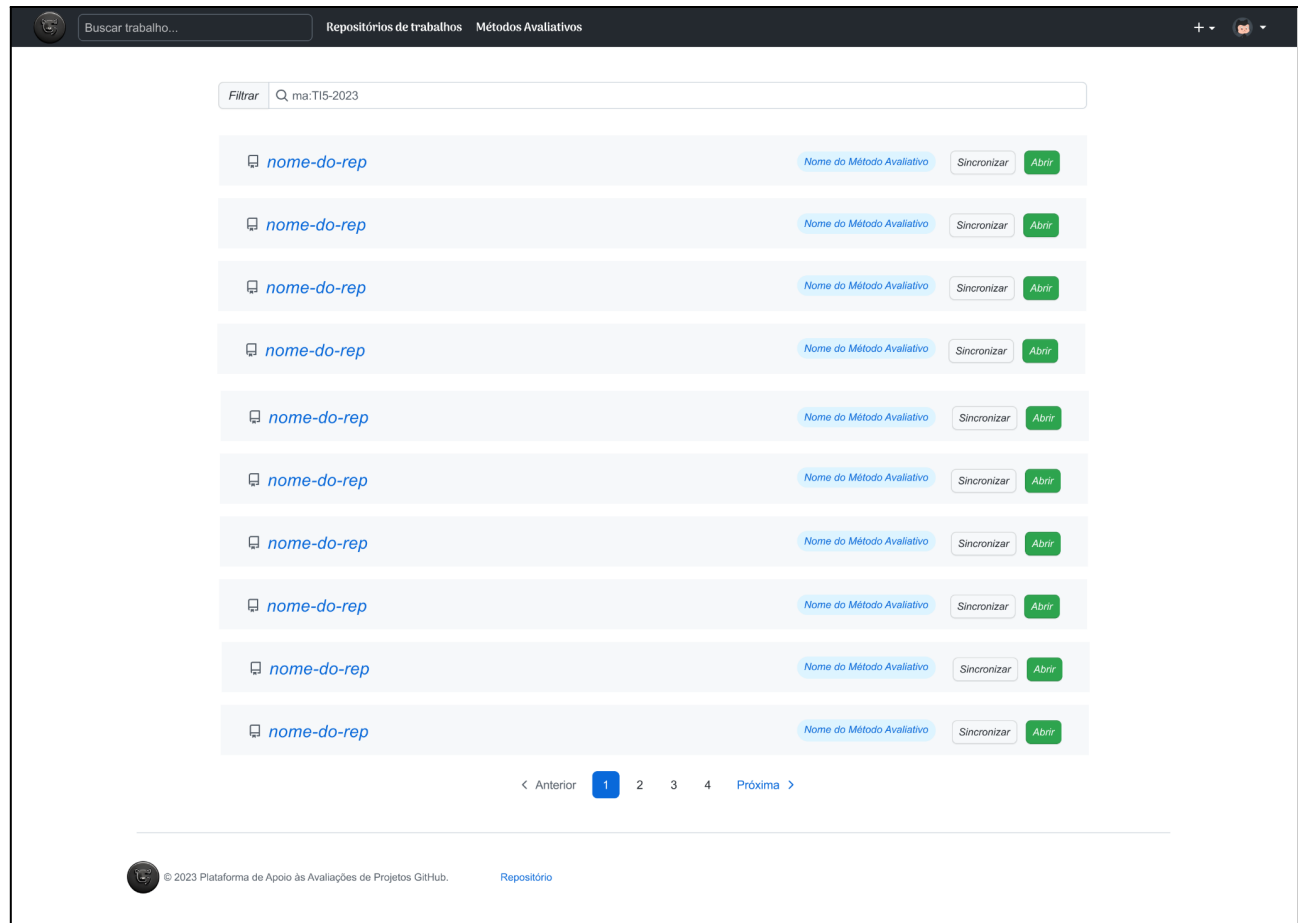


Figura 4. Lista de repositórios de trabalhos

A Figura 5 representa a tela de visão geral das informações de um trabalho. Nessa tela os professores podem visualizar as métricas por tipo de informação, *branches*, sprints de cada contribuidor do repositório ou do repositório inteiro. Além disso, poderá ativar e desativar a sincronização automática, sincronizar manualmente, redirecionar para o repositório, visualizar contribuidores e método avaliativo, executar análise de código, ver os problemas detectados conforme as regras de consistência do método avaliativo, acessar histórico de *commits* e alterar configurações do repositório na plataforma. Essa interface contempla os casos de uso UC11, UC24, UC25, UC26, UC12, UC13, UC14, UC15, UC16, UC17, UC18, UC19 e UC20.

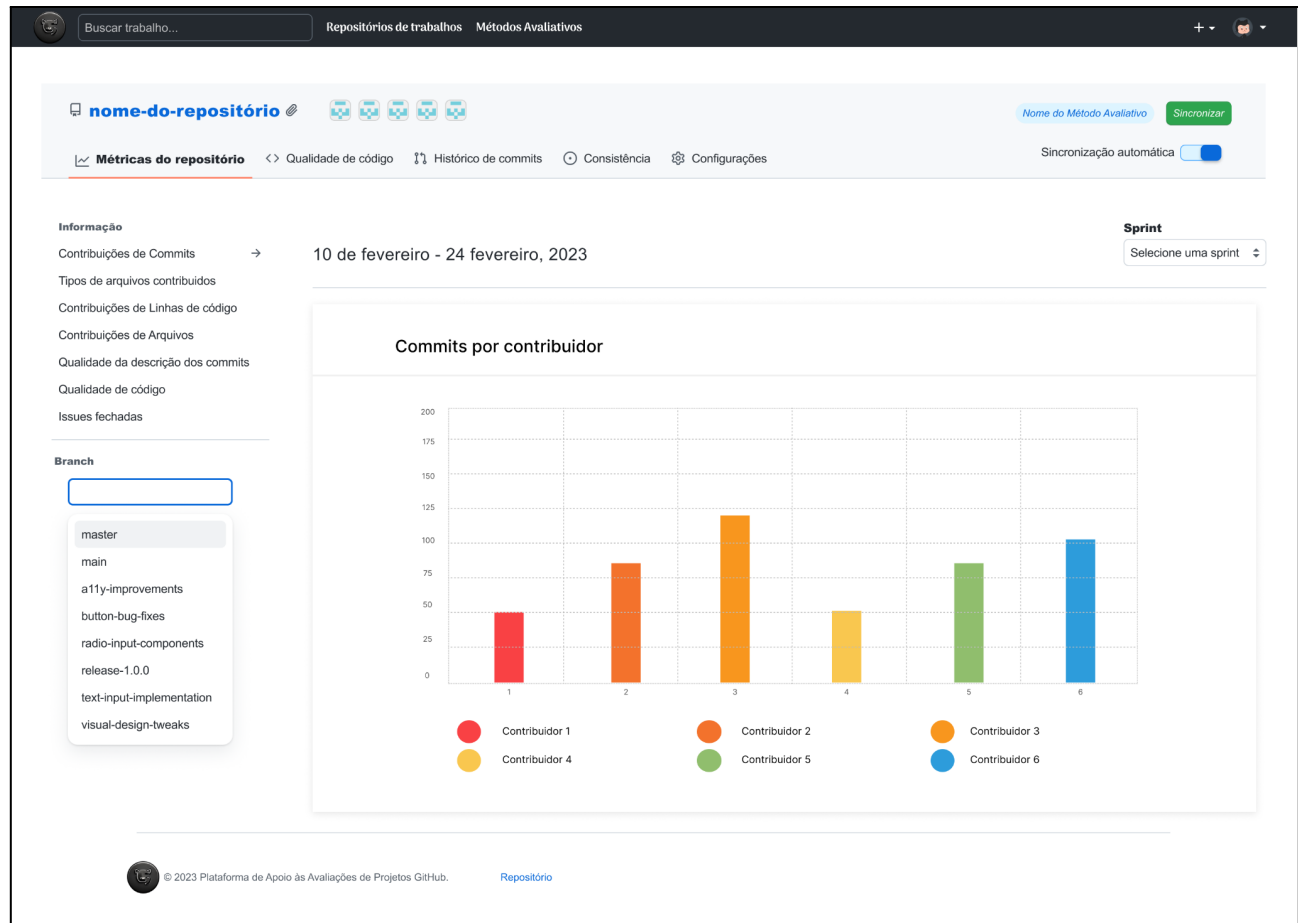


Figura 5. Visualização de trabalho

A Figura 6 é a página de listagem de métodos avaliativos cadastrados na plataforma. Nessa página é possível filtrar pelo nome do método avaliativo, abri-los e duplicá-los para facilitar na reutilização das configurações em vários semestres letivos. Essa interface contempla o caso de uso UC3.

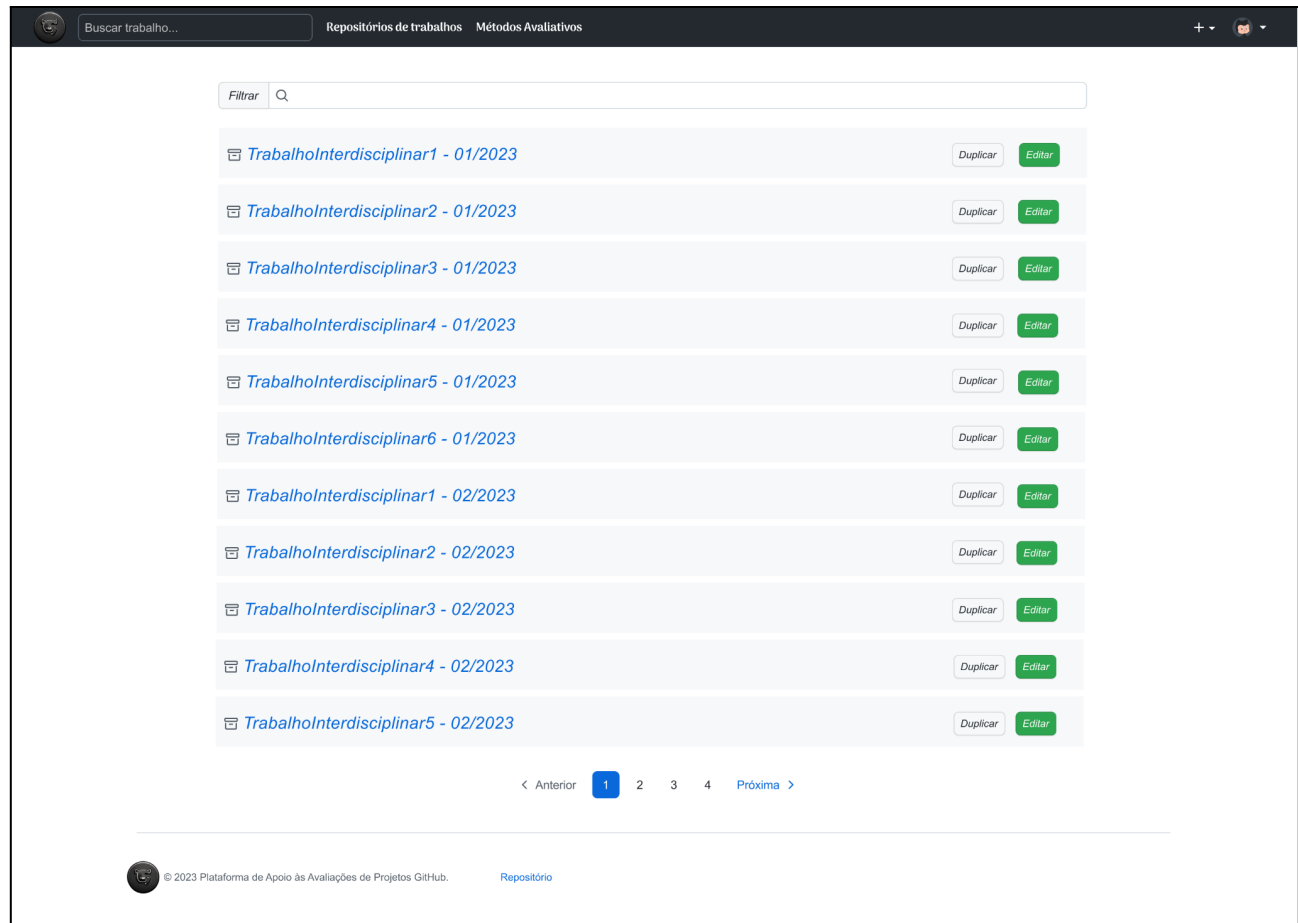


Figura 6. Lista de métodos avaliativos

A Figura 7 representa a página de gerenciamento de um método avaliativo. Nessa página é possível adicionar e filtrar quais repositórios estão sendo avaliados por esse método, duplicar o método avaliativo, cadastrar regras de consistência, sprints e *issues* padronizadas para as regras de consistência. Essa interface contempla os casos de uso UC5, UC6, UC7, UC8, UC9, UC10, UC21, UC22 e UC23.

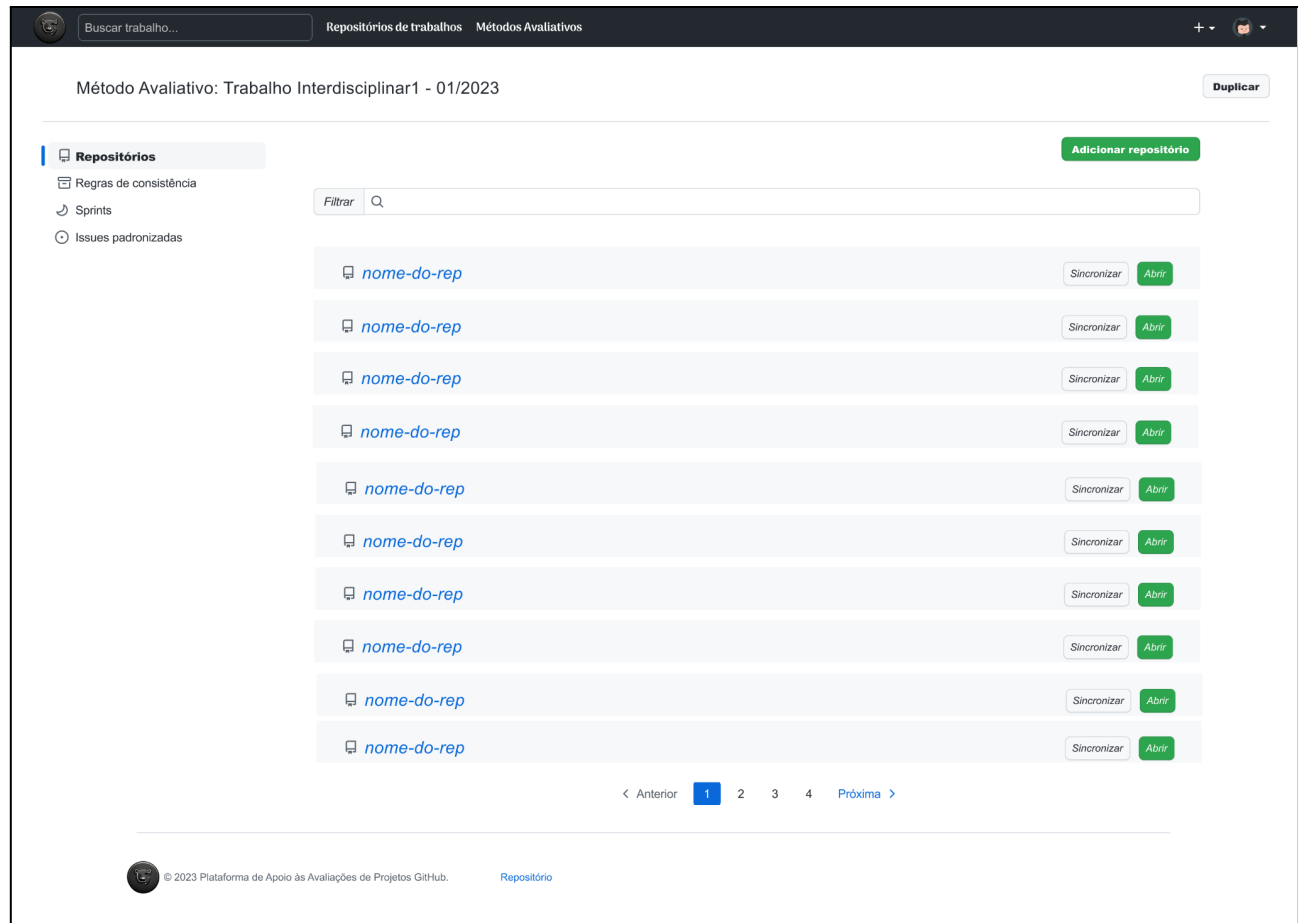


Figura 7. Visualização de método avaliativo

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.