



BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO
Análise e Projeto de Software (IC2A_CM)
Professor: Dr. Lucio Geronimo Valentin

Aluno: Eduardo Kurek
RA: 2458586

Aluno: Natanael Aparecido Tagliaferro Galafassi
RA: 2504537

Aluno: Vinícius Kurek
RA: 2515776

Sumário

| | |
|--|----|
| 1. Descrição do Sistema | 4 |
| 1.1. Introdução | 4 |
| 1.2Etapas | 4 |
| 1.3Campos..... | 5 |
| 1.4Relação de campos e etapas | 6 |
| 1.5Admin e Proprietário..... | 6 |
| 1.6Propriedade | 7 |
| 1.7Projetos..... | 7 |
| 1.8Interessados | 7 |
| 1.9Tecnologias | 8 |
| 2.Diagrama de casos de uso | 9 |
| 3.Descrição textual de casos de uso | 10 |
| 3.1 Descrição resumida | 10 |
| Caso de Uso: CRUD Proprietário | 10 |
| Caso de Uso: CRUD Propriedade | 10 |
| Caso de Uso: CRUD Projeto..... | 10 |
| 3.2Descrição completa e linguagem abstrata | 10 |
| Caso de Uso: Ver projeto | 10 |
| Caso de Uso: Montar formulário | 11 |
| Caso de Uso: Responder formulário | 12 |
| Caso de Uso: Enviar feedback do formulário | 13 |
| Caso de Uso: Ver feedback do formulário | 14 |
| 3.3 Descrição completa e linguagem concreta..... | 14 |
| Caso de Uso: Calcular Resultados; | 14 |
| 4.Modelo conceitual | 16 |
| 5.Diagrama de sequência do sistema | 17 |
| 5.1Ver projetos | 17 |
| 5.2Montar formulário..... | 18 |
| 5.3Responder formulário | 19 |
| 5.4Enviar feedback do formulário | 20 |
| 5.5Ver feedback do formulário..... | 21 |
| 5.6Calcular resultados | 22 |
| 6.Contratos de Operações | 23 |
| 6.1 getProjects(propertyId) | 23 |
| 6.2getProjects(propertyId, status) | 23 |
| 6.3getProperties()..... | 23 |
| 6.4getProperties(ownerId)..... | 24 |

| | |
|--|----|
| 6.5getProjectStages(projectId) | 24 |
| 6.6getForm(projectStageId) | 24 |
| 6.7getFormFields(projectStageId) | 25 |
| 6.8setFormField(projectStageId, stageFieldId, value) | 25 |
| 6.9sendForm(projectStageId)..... | 26 |
| 6.10 approveForm(projectStageId, comment) | 26 |
| 6.11 reapproveForm(projectStageId, comment) | 26 |
| 6.12 calculateProject(projectId) | 27 |
| 6.13 getFormFieldResult(formFieldId)..... | 27 |
| 6.14 getFormFieldResult(formFieldId)..... | 28 |
| 6.15 initForm(projectStageId)..... | 28 |
| 6.16 addField(projectStageId, formFieldId) | 28 |
| 7.Diagramas de comunicação | 29 |
| 7.1. Cadastrar propriedade | 29 |
| 7.2. Cadastrar proprietário | 30 |
| 7.3. Calcular resultados | 31 |
| 7.4. Criar etapa do projeto..... | 32 |
| 7.5. Criar projeto | 33 |
| 7.6. Montar formulário..... | 34 |
| 7.7. Responder formulário | 35 |
| 7.8. Enviar feedback do formulário | 36 |
| 7.9. Calcular resultados | 37 |
| 7.10. Ver projetos | 38 |
| 8.Diagrama de classe do projeto | 39 |

1. Descrição do Sistema

1.1. Introdução

Este documento especifica os casos de uso e requisitos não-funcionais (RNFs) do projeto intitulado *MapCarb3* referente ao projeto final da disciplina de Análise e Projeto de Software, contida no segundo semestre da matriz curricular do curso de Bacharelado em Ciência da Computação, ministrada pelo professor Dr. Lucio Geronimo Valentin. O objetivo do documento é apresentar as soluções elaboradas e discutidas durante o semestre para a construção da análise e do projeto de um software fundamentado nas ideias do paradigma orientado a objetos.

O projeto recebeu o nome de MapCarb3, e está sendo desenvolvido pelos alunos referidos neste documento. Seu propósito geral se dá como uma ferramenta para um administrador que poderá gerenciar a atividade dos proprietários e suas propriedades que estão em um relacionamento de negócio com o administrador. O sistema deve ser capaz de permitir o administrador verificar os proprietários cadastrados nele, as propriedades associadas a estes proprietários, associar projetos a estas propriedades, os quais conterão formulários referentes as etapas de produção que ocorrerão na propriedade, essas etapas terão campos específicos que deverão ser preenchidos pelo proprietário e entregues ao administrador, que avaliará, podendo aceitar o formulário, ou rejeitá-lo e enviar um feedback, mantendo o estado do formulário como pendente, os dados fornecidos serão utilizados para o cálculo de débito/crédito de carbono.

1.2 Etapas

As etapas serão definidas por nome e descrição, é interessante que exista a possibilidade de o administrador cadastrar mais etapas futuramente, as principais são:

- "Correção de solo", "Nesta etapa é feita a correção do solo para o plantio da nova cultura";
- "Preparação de solo", "Nesta etapa é feita a preparação do solo para o plantio da nova cultura";
- "Fertilização pré-plantio", "Nesta etapa é realizada a fertilização que antecede o plantio";

- "Irrigação", "Nesta etapa é realizada a irrigação do solo";
- "Plantio", "Nesta etapa é realizado o plantio da cultura";
- "Práticas de cultivo", "Esta etapa engloba todas as práticas de cultivo";
- "Colheita", "Esta etapa engloba todos os elementos utilizados no processo de colheita".

1.3 Campos

Assim como as etapas, é interessante que o administrador seja capaz de cadastrar mais futuramente, no mais, os campos possuem apenas nome para identificá-los, os principais são:

- Energia (Carvão);
- Energia (Gás natural);
- Energia (Hidráulica);
- Água (Rio);
- Água (Lago);
- Água (Chuva);
- Semente;
- Cal;
- Calcário;
- Nitrato de amônia;
- Fosfato de amônia;
- Sulfato de amônia;
- Ureia;
- Ácido nítrico;
- Esterco;
- Esterco de caldeira;
- Chorume de gado;
- Lixo de caldeira;
- Estrume líquido de porco;
- Cálcio;
- Enxofre;
- Etanol;
- Diesel;

- Gasolina.

1.4 Relação de campos e etapas

Esta relação se dá pelo identificador dos campos, das etapas e por um multiplicador, temos então:

- "Energia (Carvão)", "Fertilização pré-plantio", 1.4909621100788581;
- "Calcário", "Plantio", 1.5602179078014735;
- "Nitrato de amônia", "Práticas de cultivo", 1.6751812908567314;
- "Sulfato de amônia", "Práticas de cultivo", 0.4952527769797042;
- "Sulfato de amônia", "Colheita", 1.3870499073407145;
- "Ácido nítrico", "Plantio", 0.47530276475036776;
- "Esterco de caldeira", "Preparação de solo", 0.22566736287033873;
- "Chorume de gado", "Fertilização pré-plantio", 1.117537111246527;
- "Lixo de caldeira", "Práticas de cultivo", 0.2507200584287942;
- "Energia (Gás natural)", "Colheita", 1.0678420073053259;
- "Estrume líquido de porco", "Irrigação", 0.9021992627551771;
- "Enxofre", "Colheita", 0.5317235608880628;
- "Diesel", "Irrigação", 0.7839524060338292;
- "Gasolina", "Correção de solo", 0.2815026942468274;
- "Energia (Hidráulica)", "Correção de solo", 1.011610316682244;
- "Energia (Hidráulica)", "Irrigação", 1.3381100296397788;
- "Água (Rio)", "Preparação de solo", 0.31890479979934616;
- "Água (Lago)", "Plantio", 1.2054319873502775;
- "Semente", "Correção de solo", 1.172682567287872;
- "Semente", "Preparação de solo", 0.876476343233582;
- "Cal", "Fertilização pré-plantio", 1.2989078307514152;

1.5 Admin e Proprietário

Ambos são usuários e possuem os campos nome, e-mail e senha, o admin terá controle e acesso total ao sistema, ele quem cadastrará os proprietários e suas propriedades, além de poder delegar projetos às propriedades do proprietário e dar feedback aos formulários que os

proprietários responderem. Já o proprietário poderá ver suas propriedades e ver os projetos associados à suas propriedades, podem responder os formulários referentes a cada etapa do projeto, e enviá-los ao administrador, o qual devolverá um feedback aprovando ou recusando o que foi entregue.

1.6 Propriedade

Refere-se as propriedades dos proprietários, as propriedades possuirão as seguintes informações:

- Nome;
- Identificador de seu proprietário;
- Projetos associados;
- Logradouro;
- Bairro (opcional);
- Cidade;
- Estado.

1.7 Projetos

Aplicado sobre as propriedades pelo administrador, o projeto consistirá das etapas citadas anteriormente com seus respectivos formulários, cada etapa do projeto apresentará um resultado e um status que circula entre “PENDENTE”, “ENVIADO”, “RECUSADO” e “APROVADO”. Também conterá o nome do projeto, descrição, identificador da propriedade ao qual está associado, também possuindo status os quais são “PENDENTE” e “FINALIZADO”.

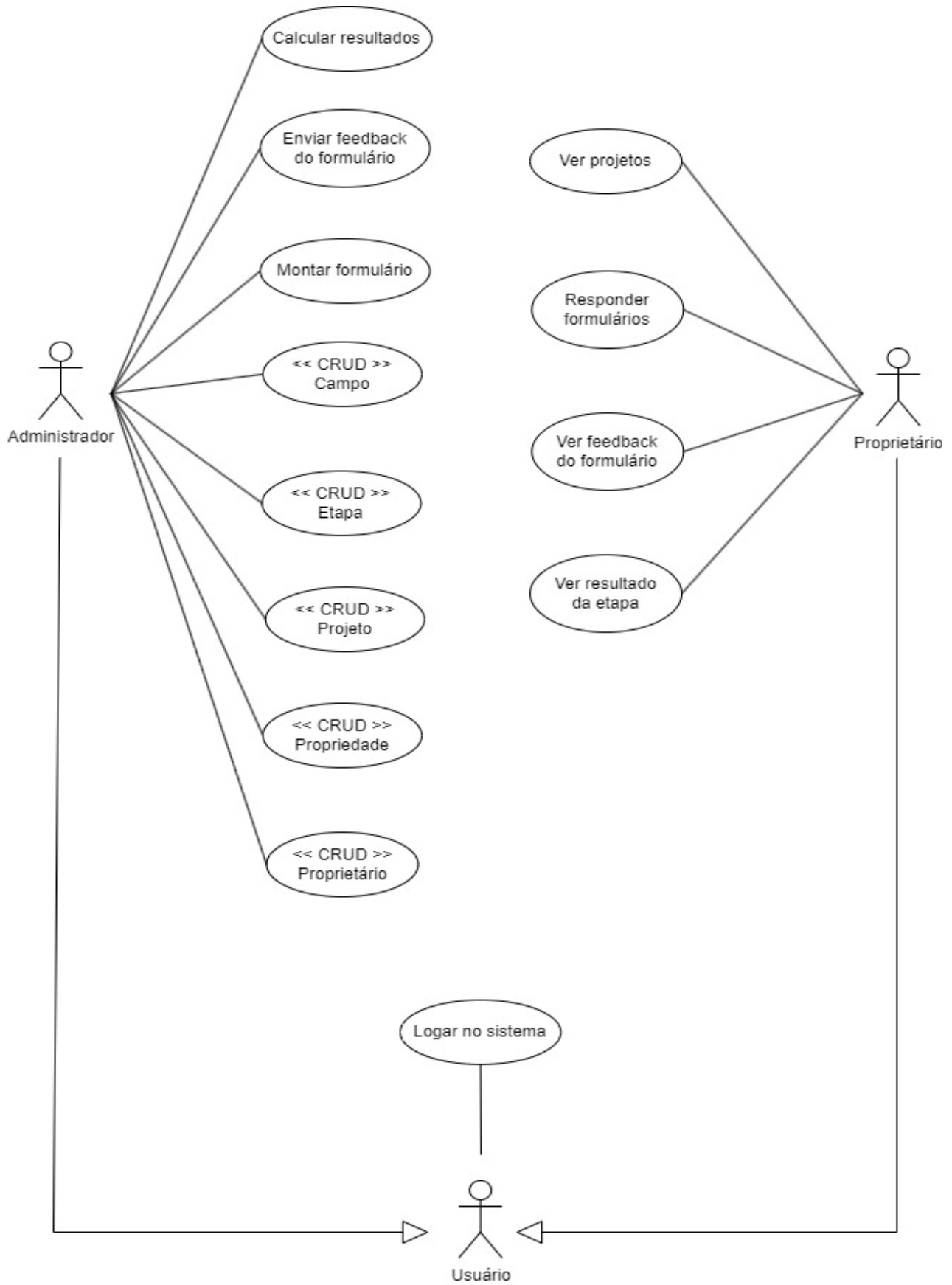
1.8 Interessados

A princípio, o administrador que gerenciará os proprietários que possuem ligações comerciais com ele, os proprietários que poderão ter ciência de seus créditos/débitos de carbono, e o professor da disciplina de Engenharia de Software, que serviu como ponte para o cliente e usará a conclusão do desenvolvimento deste projeto como avaliação da disciplina.

1.9 Tecnologias

Para o back-end será utilizado Javascript, Node.js e Express.js como frameworks, e para o front-end será utilizado Javascript, HTML/CSS/SASS, Vue.js e Tailwind como frameworks, sendo então definido a atuação do sistema em plataforma web.

2. Diagrama de casos de uso



3. Descrição textual de casos de uso

3.1 Descrição resumida

Caso de Uso: CRUD Proprietário

Visão Geral: O Administrador deve ser capaz de realizar todas as operações do CRUD quando relacionado com a existência do proprietário na plataforma, após isso o proprietário estará cadastrado. Dados que necessitam ser informados: nome, e-mail, senha.

Caso de Uso: CRUD Propriedade

Visão Geral: O Administrador deve ser capaz de realizar todas as operações de CRUD relacionado à propriedade, a qual deverá possuir um Proprietário antes de ser criada. Dados que necessitam ser informados: nome, rodovia, estado, cidade, bairro (opcional).

Caso de Uso: CRUD Projeto

Visão Geral: O Administrador deve ser capaz de criar o projeto a partir de dados fornecidos por ele, ler (acessar) o projeto que foi criado, se necessário atualizar o projeto e apagar o projeto caso haja a necessidade. Há de se notar a necessidade de haver um Proprietário e propriedade o qual receberão tal projeto. Dados que necessitam ser informados: nome, descrição.

3.2 Descrição completa e linguagem abstrata

Caso de Uso: Ver projeto

Ator Principal: Proprietário;

Interessados e Interesses:

Proprietário: Deseja acessar e verificar os projetos os quais estão atribuídos a si;

Pré-condições: O Usuário é um Proprietário que está cadastrado e logado no sistema, o projeto deverá ter sido criado previamente pelo Administrador;

Pós-condições: O Proprietário terá consciência de quais projetos ele foi incluso;

Cenário de Sucesso Principal:

1. O proprietário acessa o seu dashboard e adentra a página de projetos;
2. O sistema mostra uma lista de todos os projetos relacionados às suas propriedades;

Caso de Uso: Montar formulário

Ator Principal: Administrador;

Interessados e Interesses:

Administrador: Deseja criar o formulário que será um meio de recuperar dados da propriedade do Proprietário;

Proprietário: Deseja receber o formulário de sua propriedade, para saber as informações que serão necessárias serem escritas no formulário;

Pré-condições: Proprietário, propriedade, projeto e etapa do projeto devem ser previamente cadastrados no sistema;

Pós-condições: O formulário será criado e associado à etapa do projeto previamente selecionada, onde este proprietário poderá respondê-lo no futuro;

Cenário de Sucesso Principal:

1. O administrador procura por todas as propriedades cadastradas;
2. O sistema devolve todas uma lista com todas as propriedades existentes;
3. O administrador escolhe uma propriedade e procura por todos os projetos desta propriedade;
4. O sistema devolve todos os projetos desta propriedade;
5. O administrador clica neste projeto;
6. O sistema devolve uma lista contendo todas as etapas já criadas daquele projeto, nome do projeto, descrição e status;
7. O administrador clica em uma etapa;
8. O sistema devolve uma tela contendo todas as informações desta etapa (nome, status, resultado, e formulário caso exista);
9. O administrador clica em criar formulário;
10. O sistema devolve uma tela onde o administrador poderá adicionar vários campos relacionados àquela etapa;
11. Após o administrador escolher todos os campos, ele clica em um botão de “salvar formulário”;

12: O sistema salva todas as informações contidas neste formulário, e deixa o status da etapa em questão como PENDENTE, para que o proprietário possa responder;

Fluxos alternativos:

1.1. Caso não existam propriedades cadastradas, o administrador é informado disto e poderá criar com o caso de uso <<CRUD>> Propriedade. Retorna para o fluxo 1;

3.1. Caso não existam projetos cadastradas, o administrador é informado disto e poderá criar com o caso de uso <<CRUD>> Projeto. Retorna para o fluxo 3;

6.1. Caso não existam etapas cadastradas, o administrador é informado disto e poderá criar com o caso de uso <<CRUD>> Etapa. Retorna para o fluxo 6;

10.1. Caso não existam campos cadastrados para aquela etapa, o administrador é informado disto e poderá criar com o caso de uso <<CRUD>> Campo. Retorna para o fluxo 10;

Caso de Uso: Responder formulário

Ator Principal: Proprietário;

Interessados e Interesses:

Proprietário: Deseja responder o formulário que lhe foi atribuído pelo Administrador;

Administrador: Receberá o resultado do formulário que foi respondido pelo proprietário;

Pré-condições: O Usuário é um Proprietário que está cadastrado e logado no sistema, o formulário deverá ter sido criado previamente pelo Administrador;

Pós-condições: O formulário terá sido respondido e retornará seus resultados para o Administrador;

Cenário de Sucesso Principal:

1. O Proprietário acessa a plataforma através do login e encontra o local onde pode ver o seu dashboard;
2. No dashboard o Proprietário acessa o local responsável por projetos;
3. O sistema devolve todos os projetos em seu nome;
4. O proprietário seleciona um projeto;
5. O sistema devolve todas as etapas deste projeto;
6. O proprietário seleciona a etapa;

7. O sistema devolve as informações da etapa (nome, status e formulário);
8. O proprietário acessa o formulário da etapa;
9. O sistema devolve uma tela com todos os campos do formulário;
10. O proprietário responde todos os campos e clica em enviar formulário;
11. O sistema exibe uma mensagem de sucesso para o proprietário.

Fluxos alternativos:

- 11.1. Caso o proprietário deixou de preencher algum campo, informa mensagem de erro e retorna para o fluxo 10.

Caso de Uso: Enviar feedback do formulário

Ator Principal: Administrador;

Interessados e Interesses:

Administrador: Verificará o formulário e enviará o feedback para o Proprietário;

Proprietário: Deseja acessar e verificar os feedbacks dos formulários os quais respondeu;

Pré-condições: O Usuário é um Administrador que está cadastrado e logado no sistema, o formulário deverá ter sido respondido previamente pelo Proprietário;

Pós-condições: O Proprietário terá consciência de quais projetos ele foi incluso;

Cenário de Sucesso Principal:

1. Ao estar logado na plataforma e ter acessado seu dashboard o Administrador deverá acessar uma área para ver os projetos respondidos pelos proprietários;
2. O sistema devolverá todos os projetos cujo alguma etapa dentro dele está com o status ENVIADO;
3. O administrador acessa este projeto;
4. O sistema devolve todas as informações do projeto, incluindo as suas etapas. Diferenciando com alguma cor quais delas estão com o status ENVIADO;
5. O administrador seleciona uma etapa com o status ENVIADO;
6. O sistema devolve todo o formulário preenchido pelo proprietário;
7. O administrador avalia se os dados são coerentes, e consegue rejeitar ou aprovar este formulário desta etapa;

Fluxos alternativos:

7.1. Considerando que o formulário não foi respondido de acordo com a forma como foi montada, então o Administrador deverá enviar o feedback na caixa de comentário e reenviar o formulário para o Proprietário. Encerra o caso de uso;

Caso de Uso: Ver feedback do formulário

Ator Principal: Proprietário;

Interessados e Interesses:

Proprietário: Deseja acessar e verificar os feedbacks dos formulários os quais respondeu;

Administrador: Será quem enviou o formulário e o feedback para o Proprietário;

Pré-condições: O Usuário é um Proprietário que está cadastrado e logado no sistema, o formulário deverá ter sido respondido previamente pelo Proprietário;

Pós-condições: O Proprietário terá consciência de se seu formulário foi respondido de forma correta ou não;

Cenário de Sucesso Principal:

1. O Proprietário acessará a página de projetos que estão associados a si;
2. O sistema devolve todos os projetos associados a este proprietário;
3. O proprietário filtra por projetos que possuem alguma etapa com o status RECUSADO ou APROVADO.
4. O sistema devolve uma lista de projetos com estes filtros;
5. O proprietário seleciona um projeto;
6. O sistema devolve todas as etapas deste projeto, diferenciando com alguma cor, quais deles estão como RECUSADO ou como APROVADO;
7. O usuário seleciona uma etapa com o status APROVADO;
8. O sistema devolve uma visualização do formulário respondido;

Fluxos alternativos:

7.1 Caso o proprietário selecione uma etapa com o status REPROVADO, ele verá o feedback dito pelo administrador e retornará para o caso de uso Responder Formulário;

3.3 Descrição completa e linguagem concreta

Caso de Uso: Calcular Resultados;

Ator Principal: Administrador;

Interessados e Interesses:

Administrador: Deseja saber os resultados de todos os formulários de um certo projeto;

Proprietário: Também desejará saber o resultado futuro do cálculo dos projetos;

Pré-condições: O projeto deverá ter sido criado previamente pelo Administrador;

Pós-condições: O projeto terá um resultado que pode ser visto tanto pelo Administrador quanto pelo proprietário;

Cenário de Sucesso Principal:

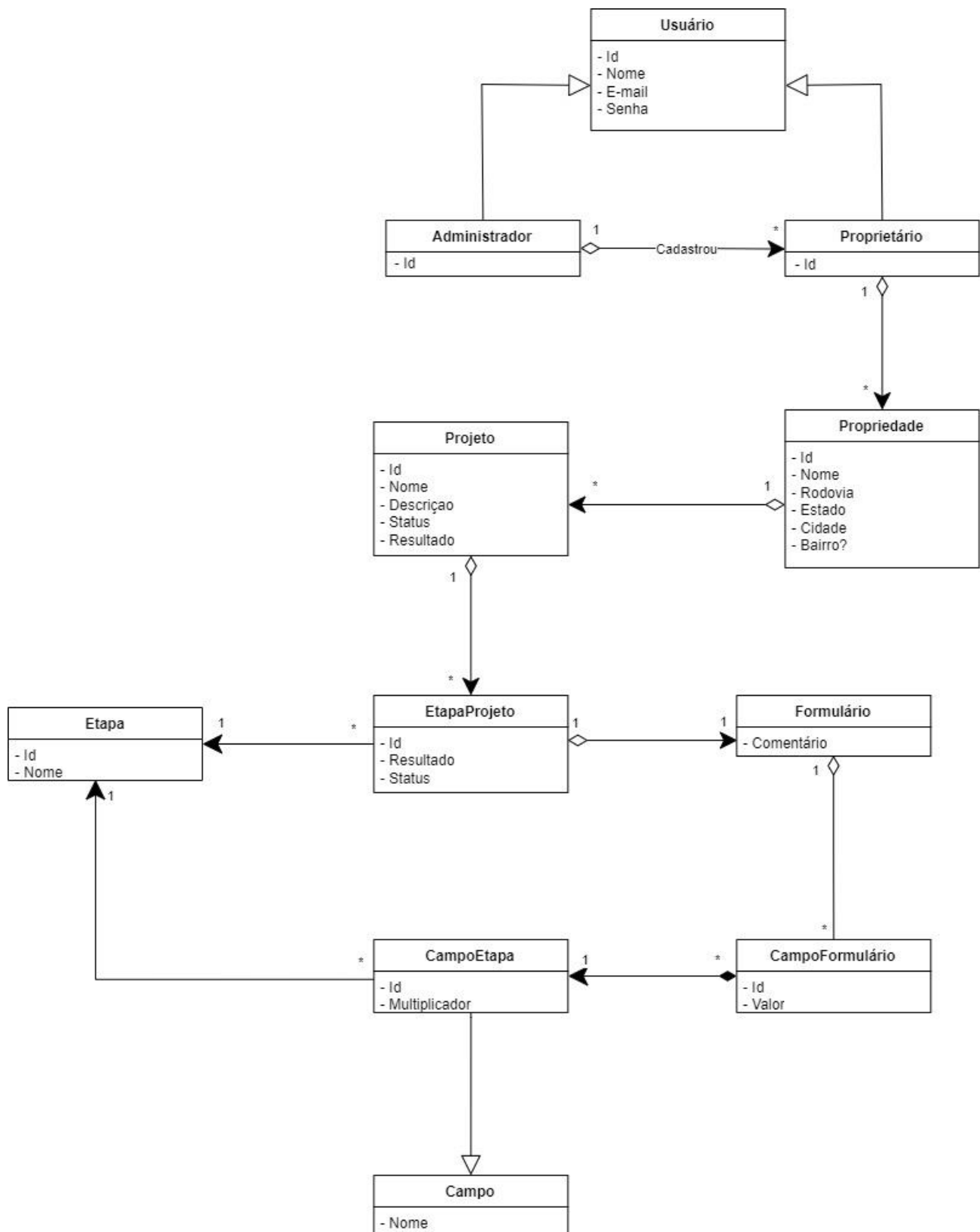
1. O Administrador dentro do seu dashboard clicará para ver os projetos;
2. O sistema exibirá o nome e a qual proprietário pertencem todos os projetos os quais foram cadastrados pelo Administrador;
3. O Administrador clicará no botão de filtrar por proprietário;
4. O Sistema lhe devolve uma caixa de seleção com todos os proprietários que ele cadastrou;
5. O Administrador seleciona qual Proprietário ele deseja ver os projetos;
6. O Sistema atualiza os projetos para filtrar apenas aqueles com o Proprietário selecionado;
7. O Administrador seleciona o projeto o qual ele deseja calcular os resultados;
8. O Sistema devolve uma tela com o nome do projeto, descrição e todas suas etapas, além de um botão de calcular resultados;
9. O Administrador clicará no botão de calcular resultados;
10. O sistema calcula o resultado de todas as etapas e grava o resultado no projeto que será informado imediatamente para o Administrador na tela.

Fluxos alternativos:

9.1 Caso o Administrador clique no botão de calcular resultados e o projeto não possui todas suas etapas necessárias (que são todas aquelas que foram definidas pelo Cliente);

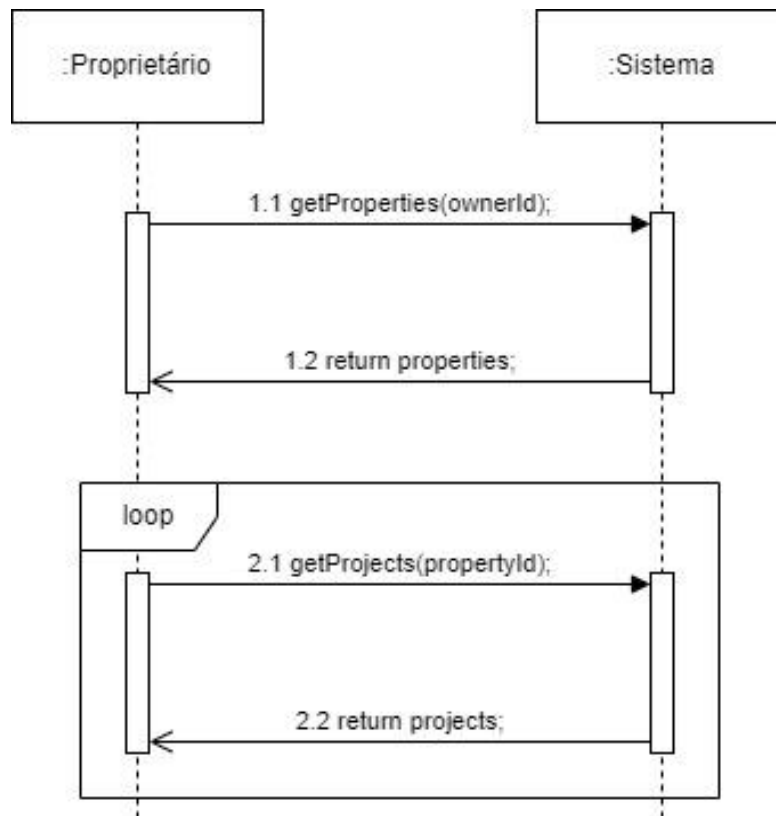
9.2 O Sistema informará um erro e encerrará o caso de uso.

4. Modelo conceitual

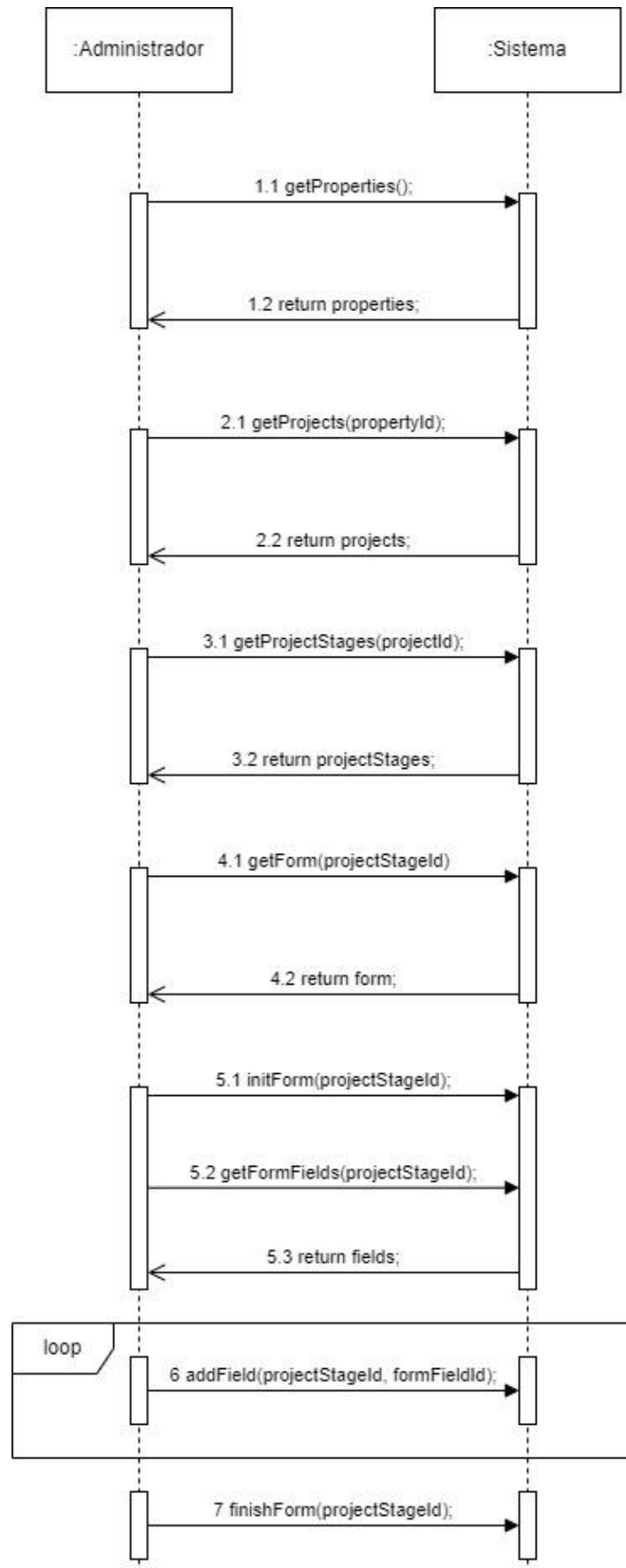


5. Diagrama de sequência do sistema

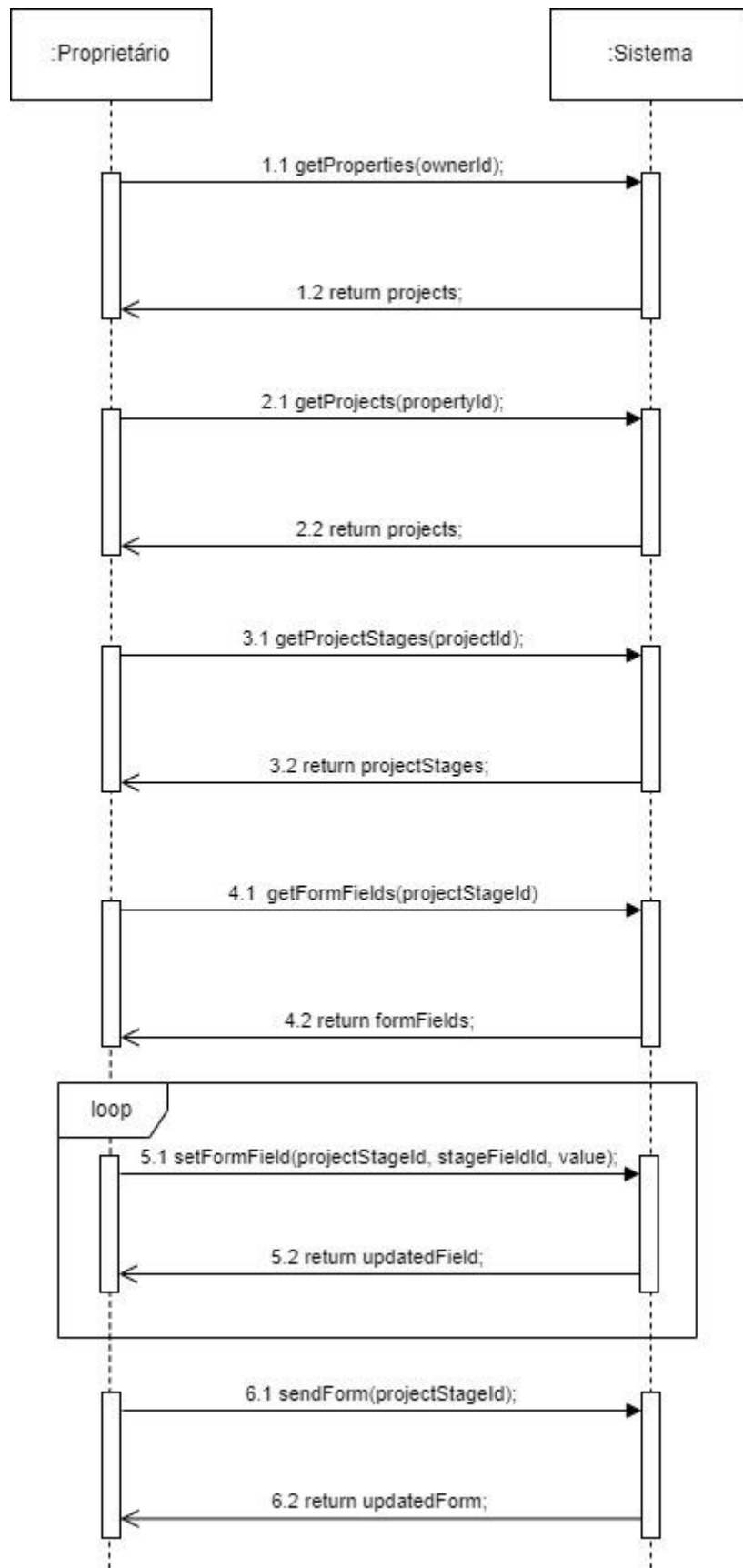
5.1 Ver projetos



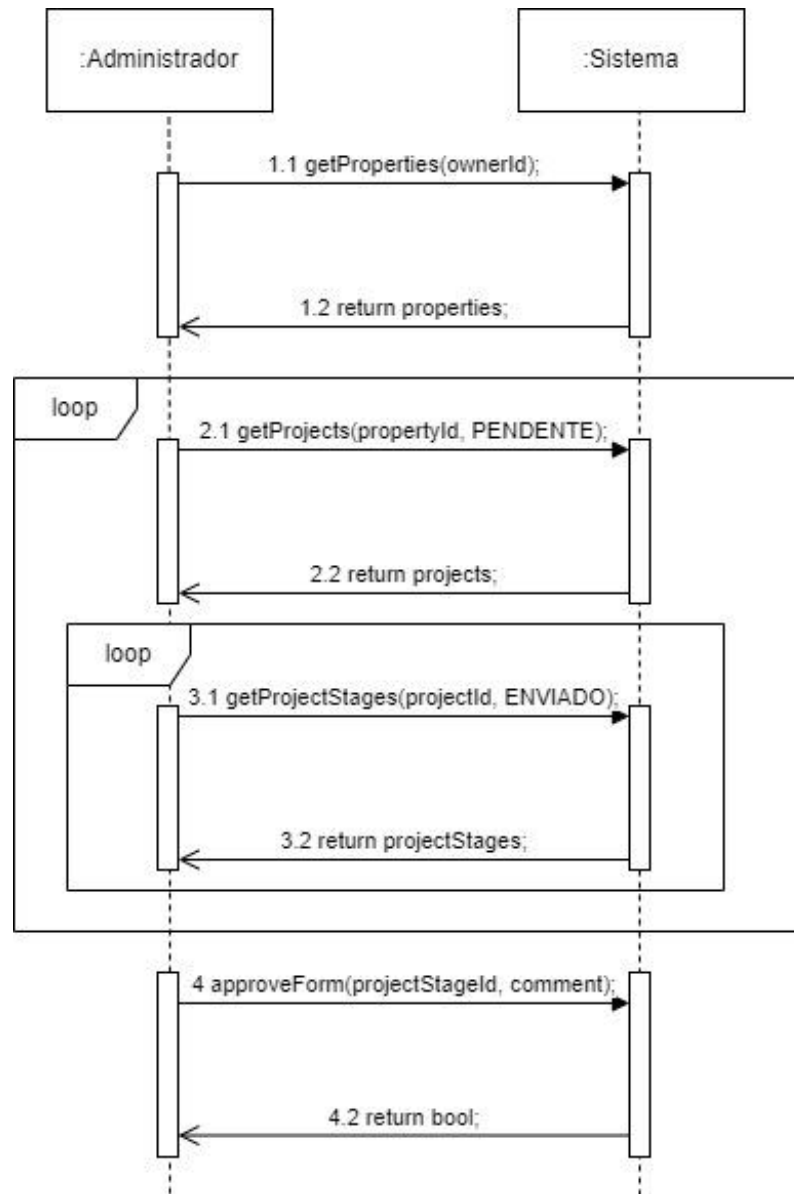
5.2 Montar formulário



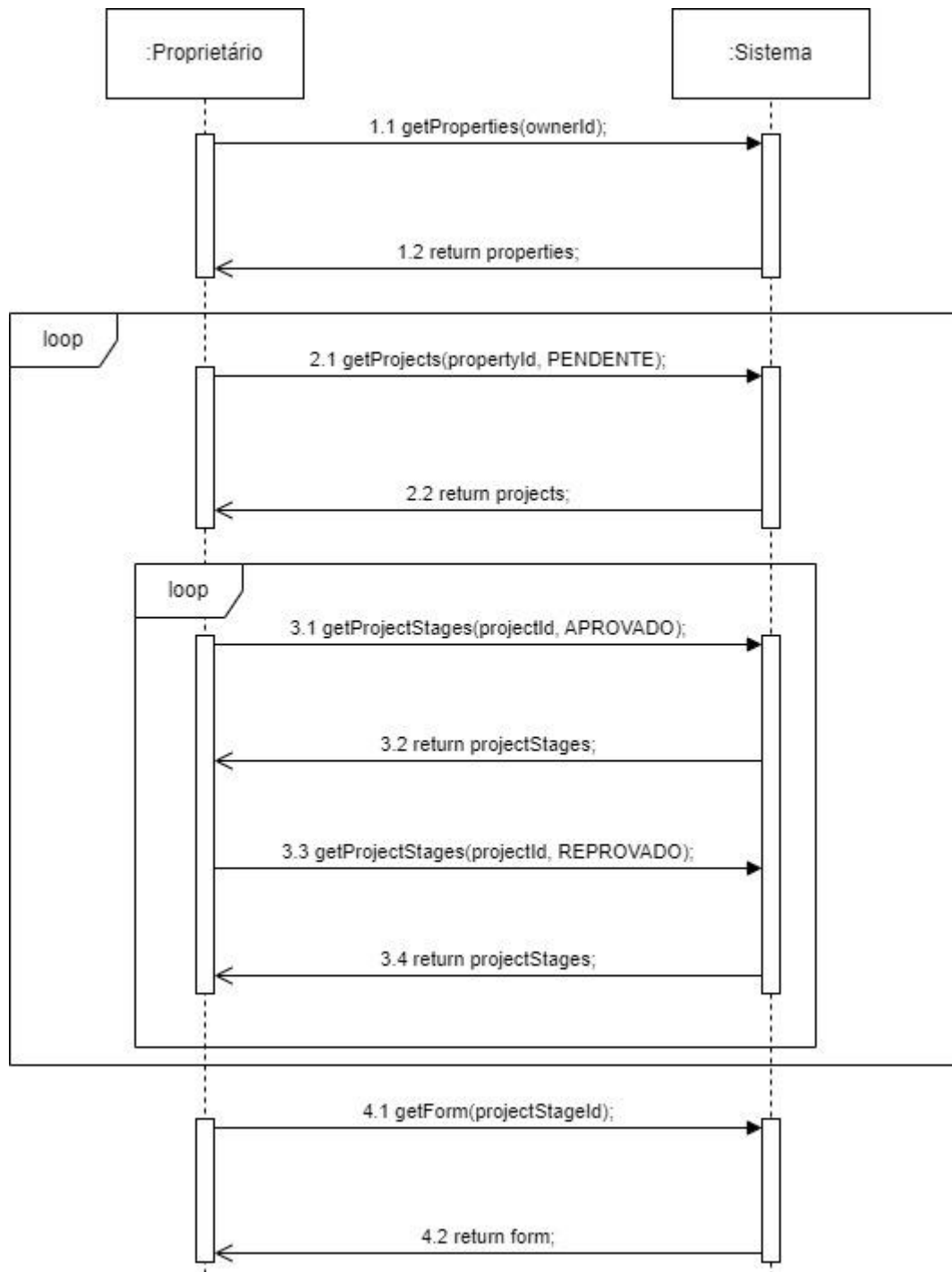
5.3 Responder formulário



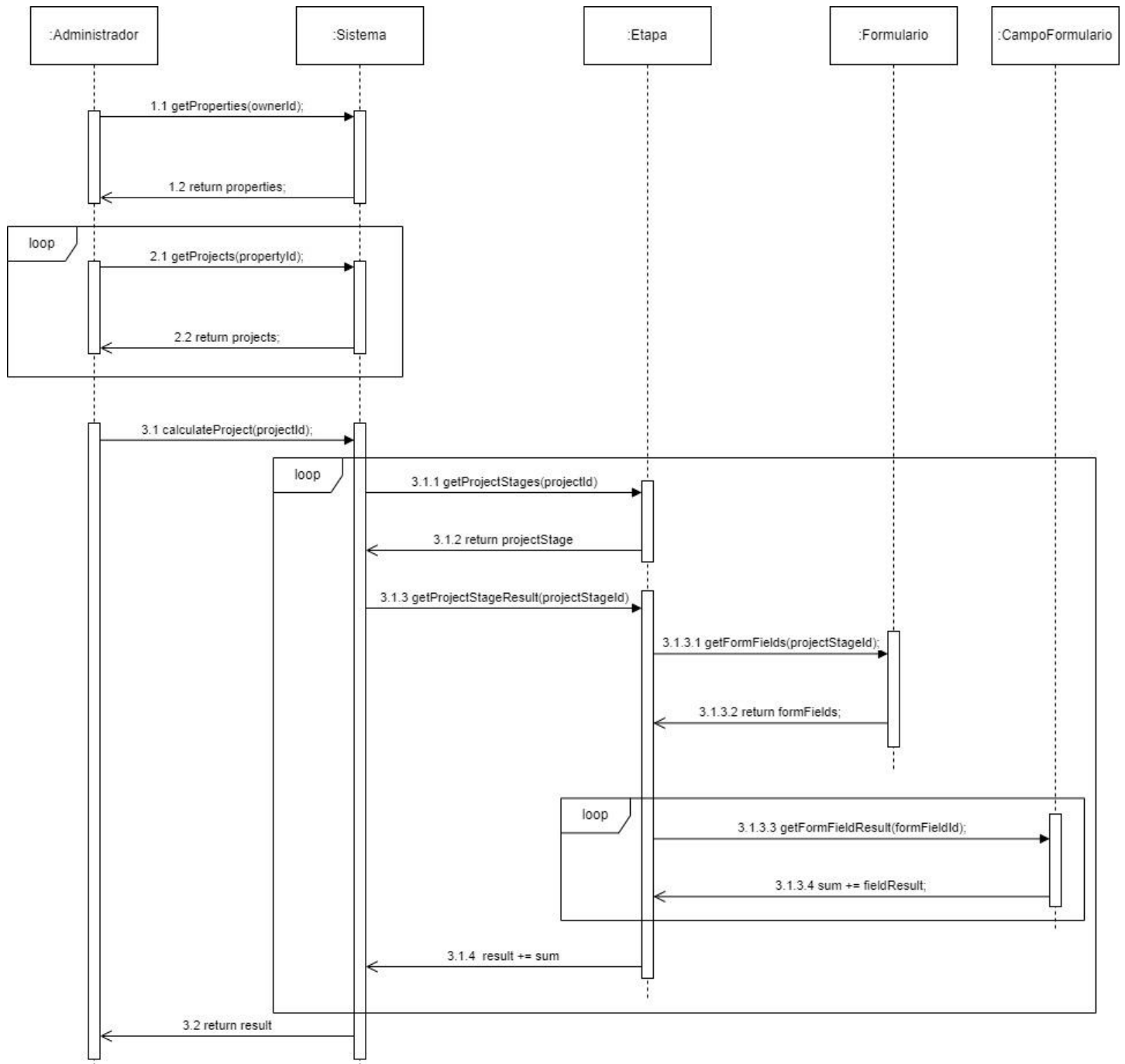
5.4 Enviar feedback do formulário



5.5 Ver feedback do formulário



5.6 Calcular resultados



6. Contratos de Operações

6.1 getProjects(propertyId)

Operação: getProjects(string propertyId);

Parâmetros:

- propertyId: Id da propriedade a qual deseja-se saber todos os projetos;

Referências Cruzadas:

- Casos de uso: Ver projeto, Montar Formulário, Responder formulário, Calcular resultados;

Pré-condições:

- Alguma propriedade deve ter sido cadastrada no sistema;

Pós-condições:

- O sistema apresentará todos os projetos relacionadas àquela propriedade;

6.2 getProjects(propertyId, status)

Operação: getProjects(string propertyId, enum status);

Parâmetros:

- propertyId: Id da propriedade a qual deseja-se saber todos os projetos;
- status: É um enum que pode procurar por projetos de acordo com o status escolhido. Os possíveis estados são: PENDENTE, FINALIZADO;

Referências Cruzadas:

- Casos de uso: Enviar feedback do formulário, Ver feedback do formulário;

Pré-condições:

- A propriedade deve ter sido criada previamente;

Pós-condições:

- O sistema retornará todos os projetos de acordo com a propriedade e status enviados como parâmetro;

6.3 getProperties()

Operação: getProperties();

Parâmetros: Nenhum;

Referências Cruzadas:

Casos de uso: Montar Formulário;

Pré-condições:

- O proprietário deve estar cadastrado e logado no sistema e possuir ao menos uma propriedade cadastrada no sistema;

Pós-condições:

- O sistema retornará todas as propriedades que já foram criadas;

6.4 getProperties(ownerId)

Operação: getProperties(string ownerId);

Parâmetros:

- ownerId: Id do proprietário que deseja-se buscar as propriedades;

Referências Cruzadas:

Casos de uso: Ver projeto, Responder formulário, Enviar feedback do formulário, Ver feedback do formulário, Calcular Resultados;

Pré-condições:

- O proprietário deve estar cadastrado no sistema;

Pós-condições:

- O sistema retornará todas as propriedades que já foram criadas para aquele proprietário;

6.5 getProjectStages(projectId)

Operação: getProjectStages(string projectId);

Parâmetros:

- projectId: Id do projeto o qual deseja-se recuperar as etapas;

Referências Cruzadas:

Casos de uso: Enviar feedback do formulário, Ver feedback do formulário;

Pré-condições:

- Ter um projeto criado o qual possuirá etapas;

Pós-condições:

- O sistema retorna todos as etapas do projeto que foi passado como parâmetro;

6.6 getForm(projectStageId)

Operação: getForm(string projectStageId);

Parâmetros:

- projectStageId: Id da etapa de um projeto da qual a função deve retornar os dados do formulário;

Referências Cruzadas:

Casos de uso: Montar formulário;

Pré-condições:

- Usuário estar cadastrado e logado no sistema;
- Ter um projeto criado com uma etapa;

Pós-condições:

- O sistema retorna os dados do formulário da etapa que foi pedido;

6.7 getFormFields(projectStageId)

Operação: getFormFields(string projectStageId);

Parâmetros:

- projectStageId: Id da etapa de um projeto da qual deseja-se saber os campos do formulário;

Referências Cruzadas:

- Casos de uso: Montar formulário, Responder Formulário, Ver feedback do formulário, Calcular resultados;

Pré-condições:

- Ter um projeto criado com uma etapa;

Pós-condições:

- O sistema retorna os campos do formulário;

6.8 setFormField(projectStageId, stageFieldId, value)

Operação: setFormField(string projectStageId, string stageFieldId, float value)

Parâmetros:

- projectStageId: Id da etapa do projeto que contém o formulário;
- stageFieldId: Id do campo do formulário que se deseja atualizar;
- value: novo valor que será atribuído ao campo que foi passado como parâmetros;

Referência Cruzada:

Casos de uso: Responder formulário;

Pré-condições:

- Ter um projeto, etapa, e campo de um formulário definido no qual será aplicado a função;

Pós-condições:

- O sistema atualiza o campo do formulário que foi solicitado com o novo valor;

6.9 sendForm(projectStageId)

Operação: sendForm(string projectStageId);

Parâmetros:

- projectStageId: Id da etapa do projeto que contém o formulário que será enviado avaliação;

Referência Cruzada:

Casos de uso: Responder formulário;

Pré-condições:

- Ter um projeto com uma etapa cadastrada;
- Esta etapa deve estar com o status “PENDENTE” ou “RECUSADO”;

Pós-condições:

- O sistema define o status daquele formulário como “ENVIADO”;

6.10 approveForm(projectStageId, comment)

Operação: approveForm(string projectStageId)

Parâmetros:

- projectStageId: Id da etapa de um projeto que deseja-se aprovar o formulário;
- comment: Comentário sobre a aprovação;

Referência Cruzada:

Casos de uso: Enviar feedback do formulário.

Pré-condições:

- Um projeto deve ter sido criado, com uma etapa também atribuída;
- Esta etapa deve estar com o status “ENVIADO”;

Pós-condições:

- O sistema define o status daquele formulário como “APROVADO”;

6.11 reproveForm(projectStageId, comment)

Operação: `reproveForm(string projectId)`

Parâmetros:

- `projectId`: Id da etapa de um projeto que deseja-se aprovar o formulário;
- `comment`: Comentário do motivo da reprovação;

Referência Cruzada:

Casos de uso: Enviar feedback do formulário.

Pré-condições:

- Um projeto deve ter sido criado, com uma etapa também atribuída;
- Esta etapa deve estar com o status “ENVIADO”;

Pós-condições:

- O sistema define o status daquele formulário como “REPROVADO”;

6.12 `calculateProject(projectId)`

Operação: `calculateProject(string projectId)`;

Parâmetro:

- `projectId`: Id do projeto o qual você quer calcular o resultado;

Referência Cruzada:

Casos de uso: Calcular resultados;

Pré-condições:

- O projeto deve ser criado previamente;
- O projeto deve conter **todas** as etapas possíveis, cadastradas;
- Todas estas etapas devem estar com o status “APROVADO”;

Pós-condições:

- O sistema calculará o resultado de todos os campos daquele projeto e atualizará o valor final do projeto, também definirá o status do projeto como “FINALIZADO”;

6.13 `getFormFieldResult(formFieldId)`

Operação: `getFormFieldResult(string formFieldId)`

Parâmetros:

- `formFieldId`: id do campo do formulário que se deseja recuperar o resultado;

Referência Cruzada:

Casos de uso: Calcular resultado;

Pré-condições:

- Uma etapa de um projeto deve ter sido criada e respondida;

Pós-condições:

- O Sistema retornará o valor que o proprietário inseriu do campo o qual foi pedido;

6.14 getFormFieldResult(formFieldId)

Operação: getFormFieldResult(string formFieldId)

Parâmetros:

- formFieldId: id do campo do formulário que se deseja recuperar o resultado;

Referência Cruzada:

Casos de uso: Calcular resultado;

Pré-condições:

- Uma etapa de um projeto deve ter sido criada e respondida;

Pós-condições:

- O Sistema retornará o valor que o proprietário inseriu do campo o qual foi pedido;

6.15 initForm(projectStageId)

Operação: initForm(string projectStageId)

Parâmetros:

- projectStageId: Id da etapa de um projeto;

Referência Cruzada:

Casos de uso: Montar formulário;

Pré-condições:

- Um projeto deve ter sido criado, juntamente com uma etapa;

Pós-condições:

- O sistema permitirá a inclusão de campos para este formulário;

6.16 addField(projectStageId, formFieldId)

Operação: addField (string projectStageId, string formFieldId)

Parâmetros:

- projectStageId: Id da etapa de um projeto;
- formFieldId: Id do campo do formulário a ser adicionado;

Referência Cruzada:

Casos de uso: Montar formulário;

Pré-condições:

- Um projeto deve ter sido criado, juntamente com uma etapa;

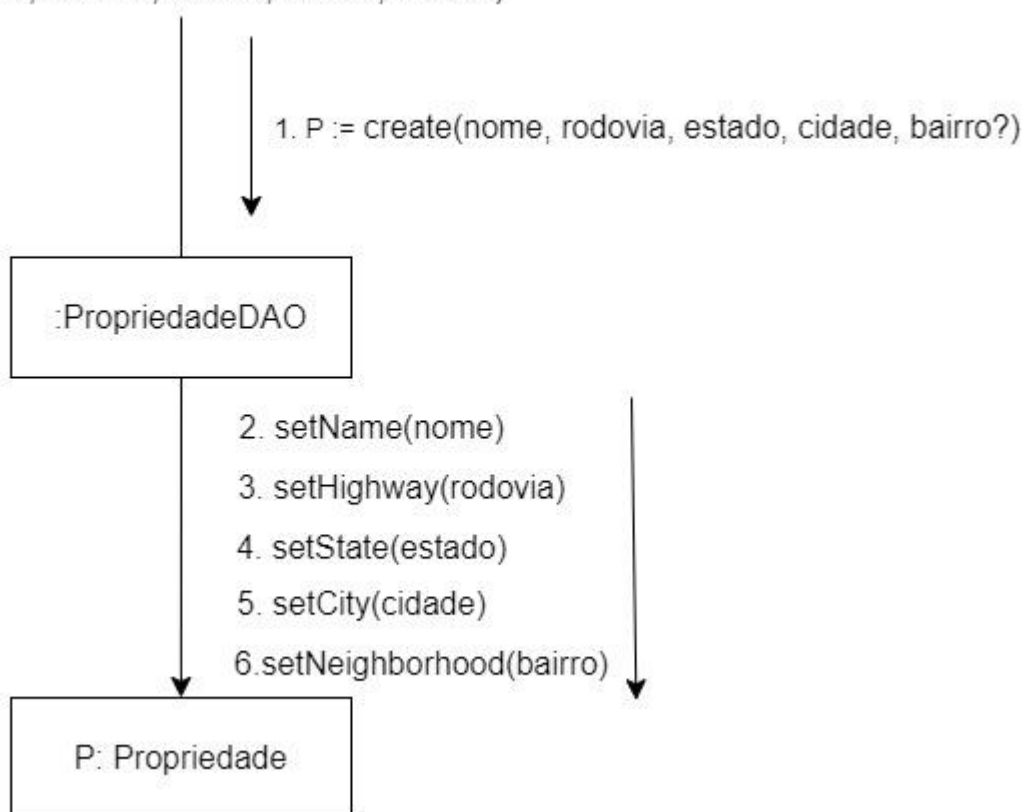
Pós-condições:

- O sistema incluirá este campo ao formulário da etapa;

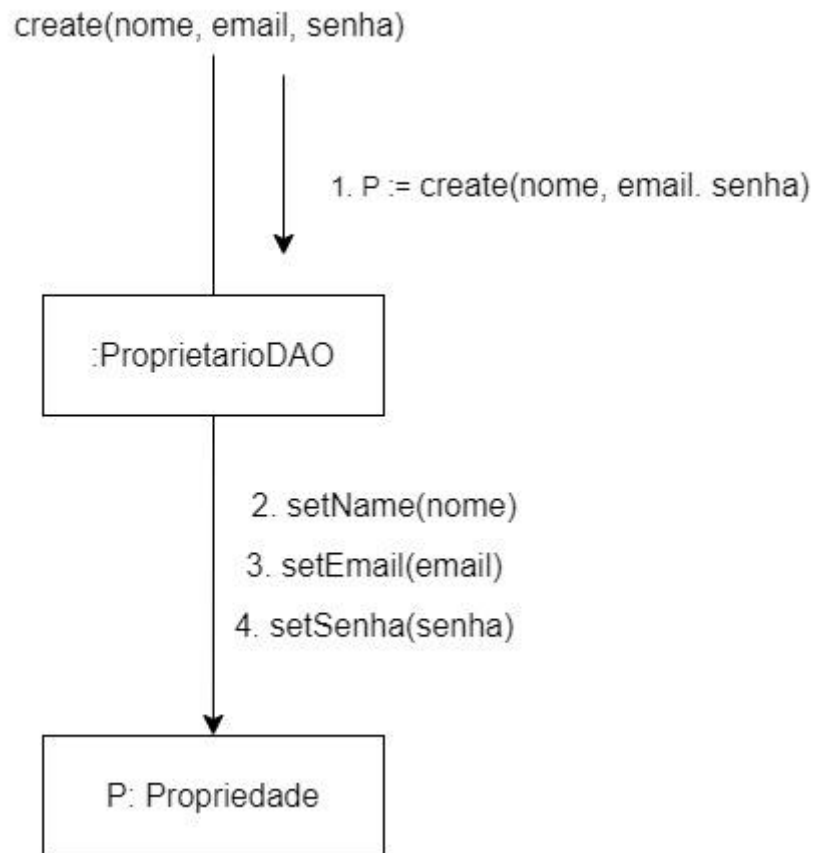
7. Diagramas de comunicação

7.1 Cadastrar propriedade

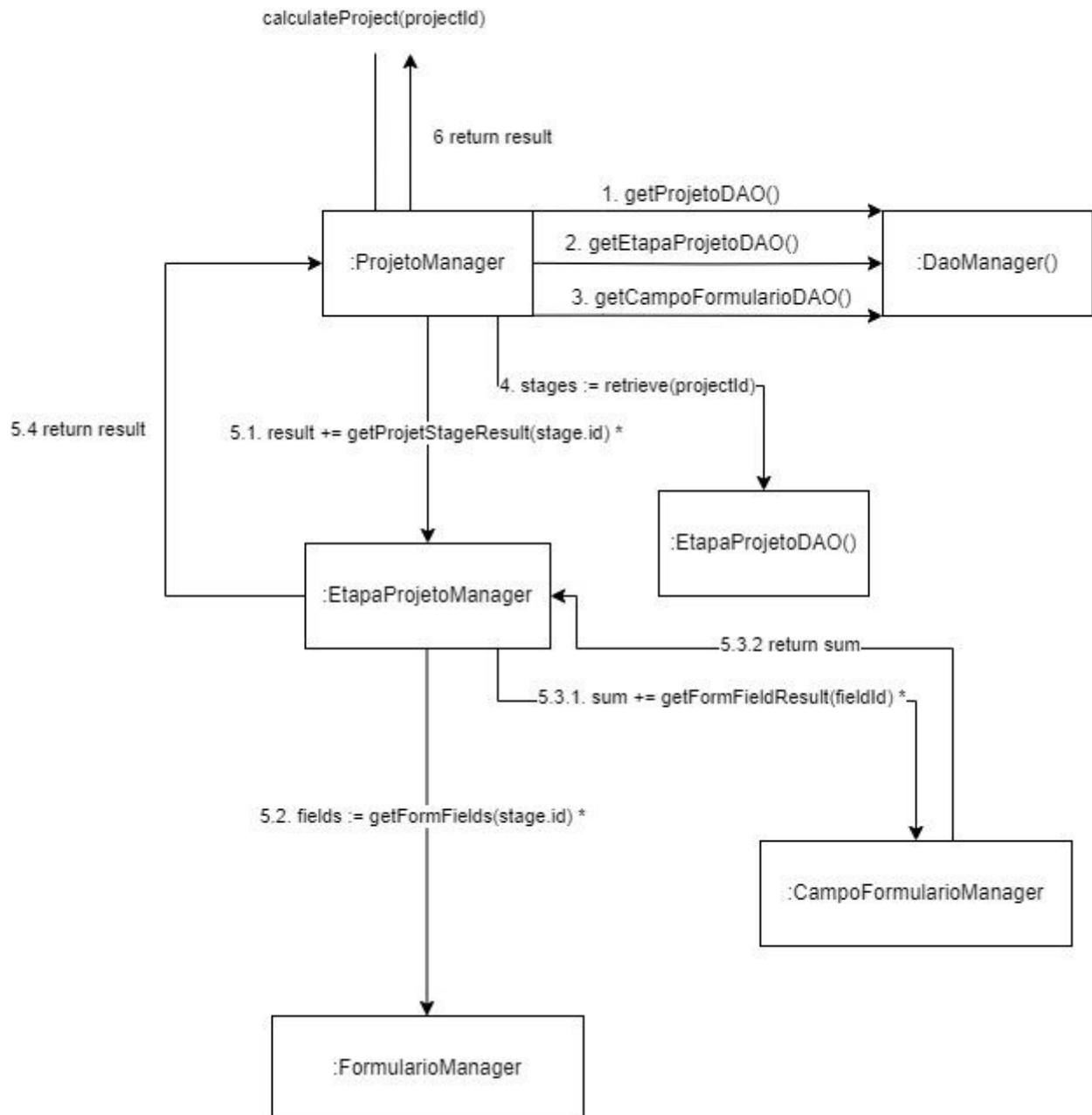
create(nome, rodovia, estado, cidade, bairro?)



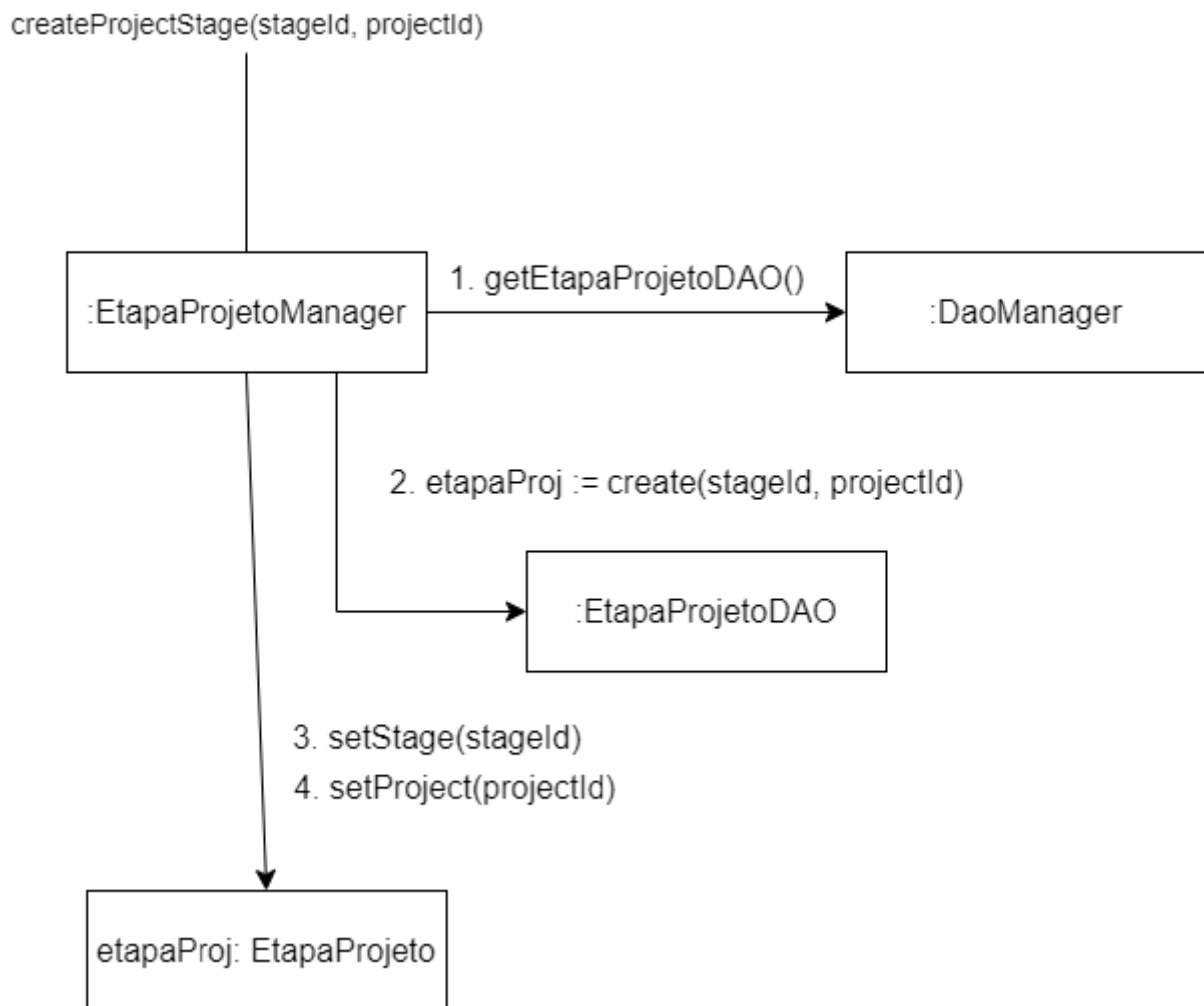
7.2 Cadastrar proprietário



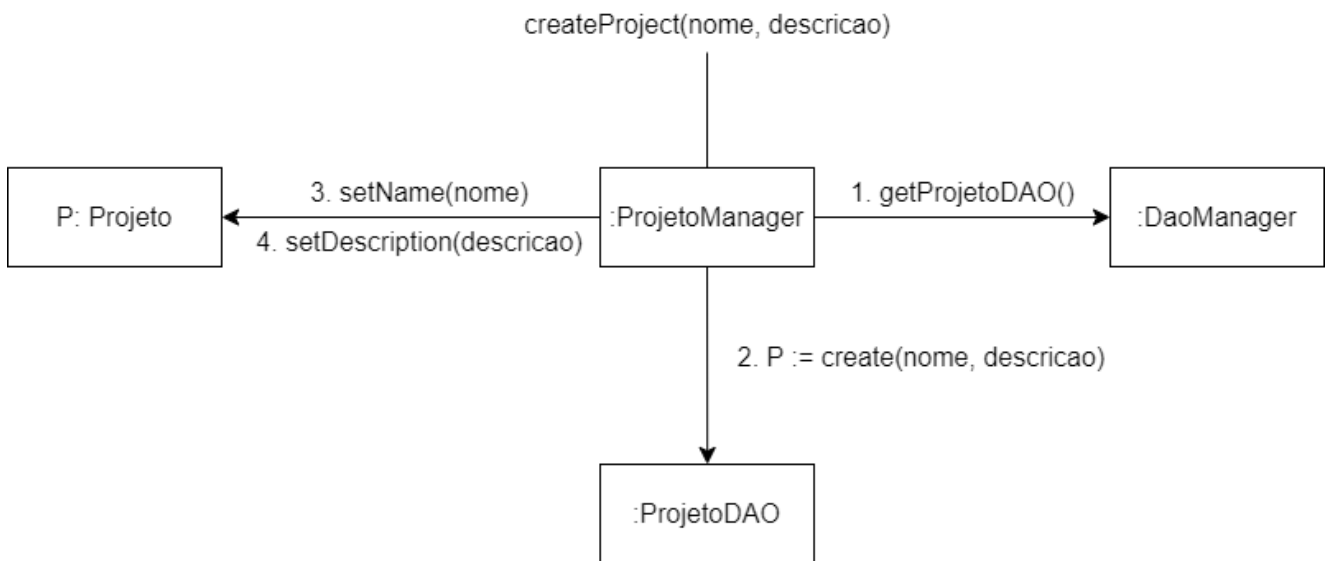
7.3 Calcular resultados



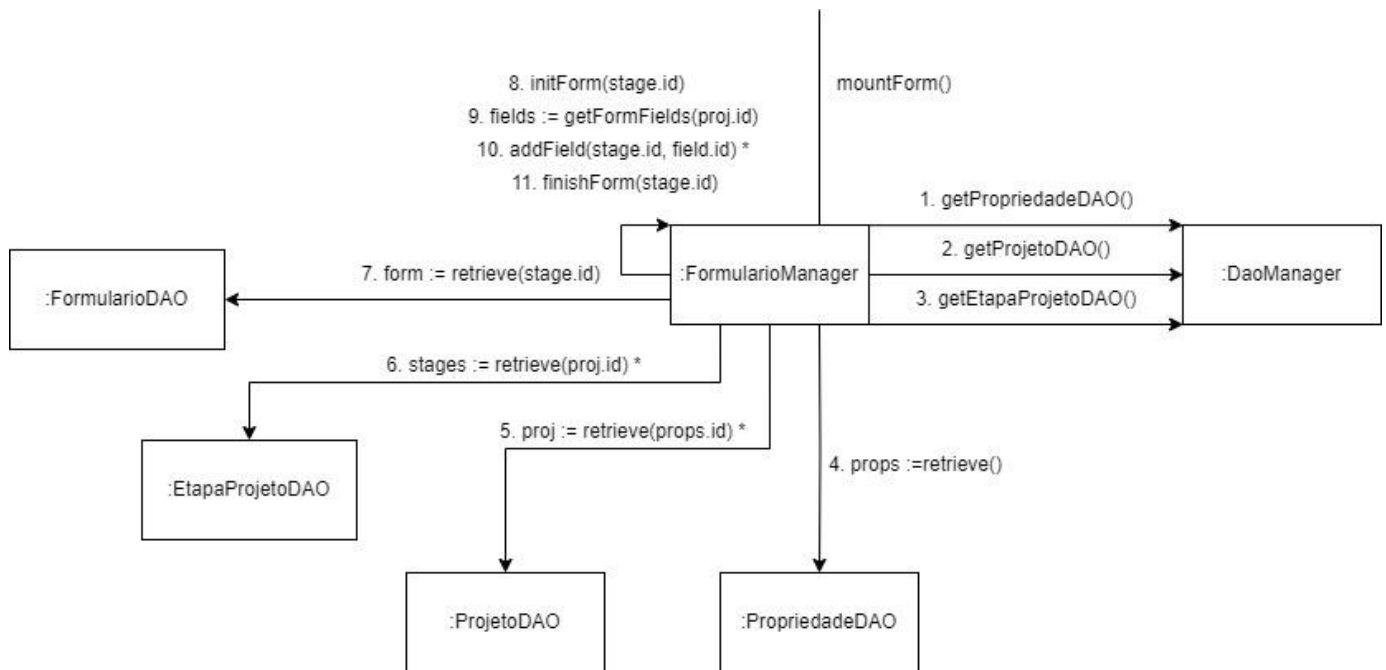
7.4 Criar etapa do projeto



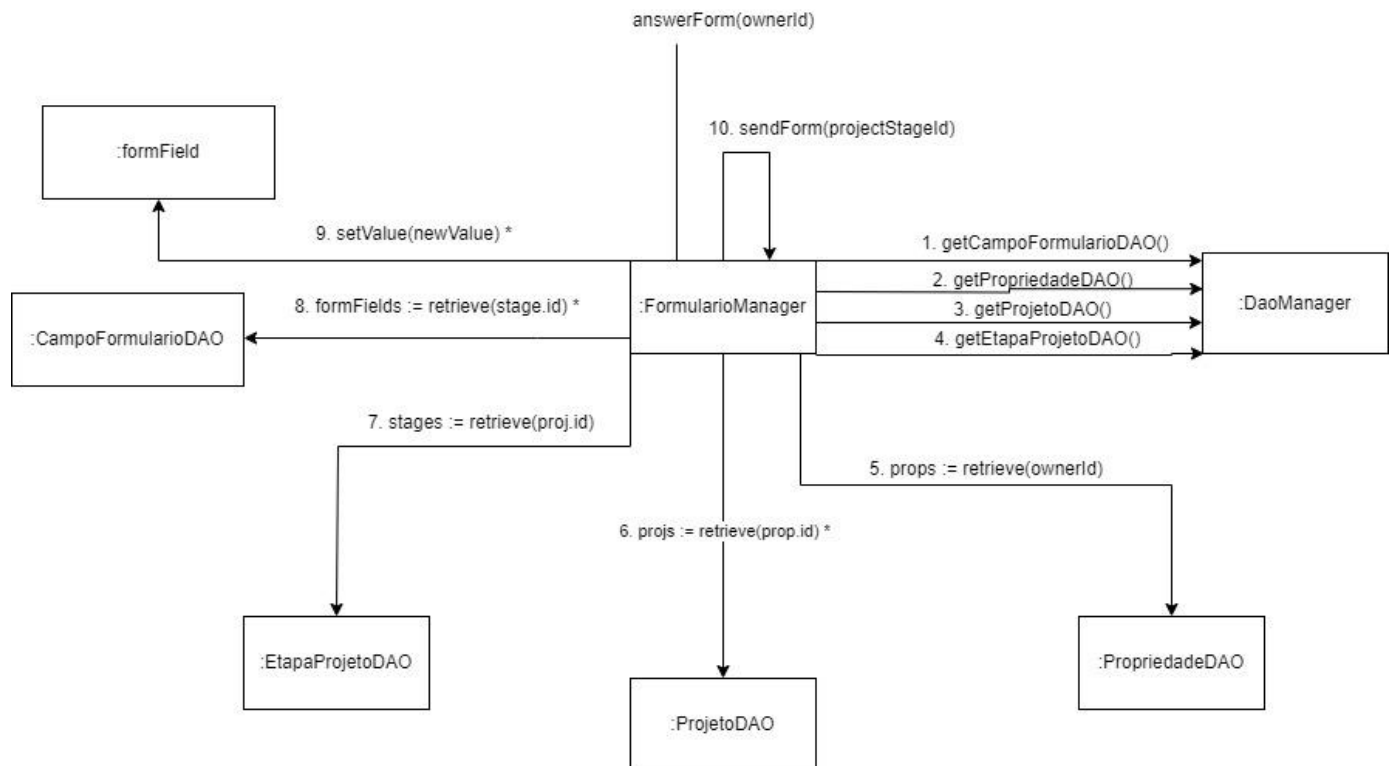
7.5 Criar projeto



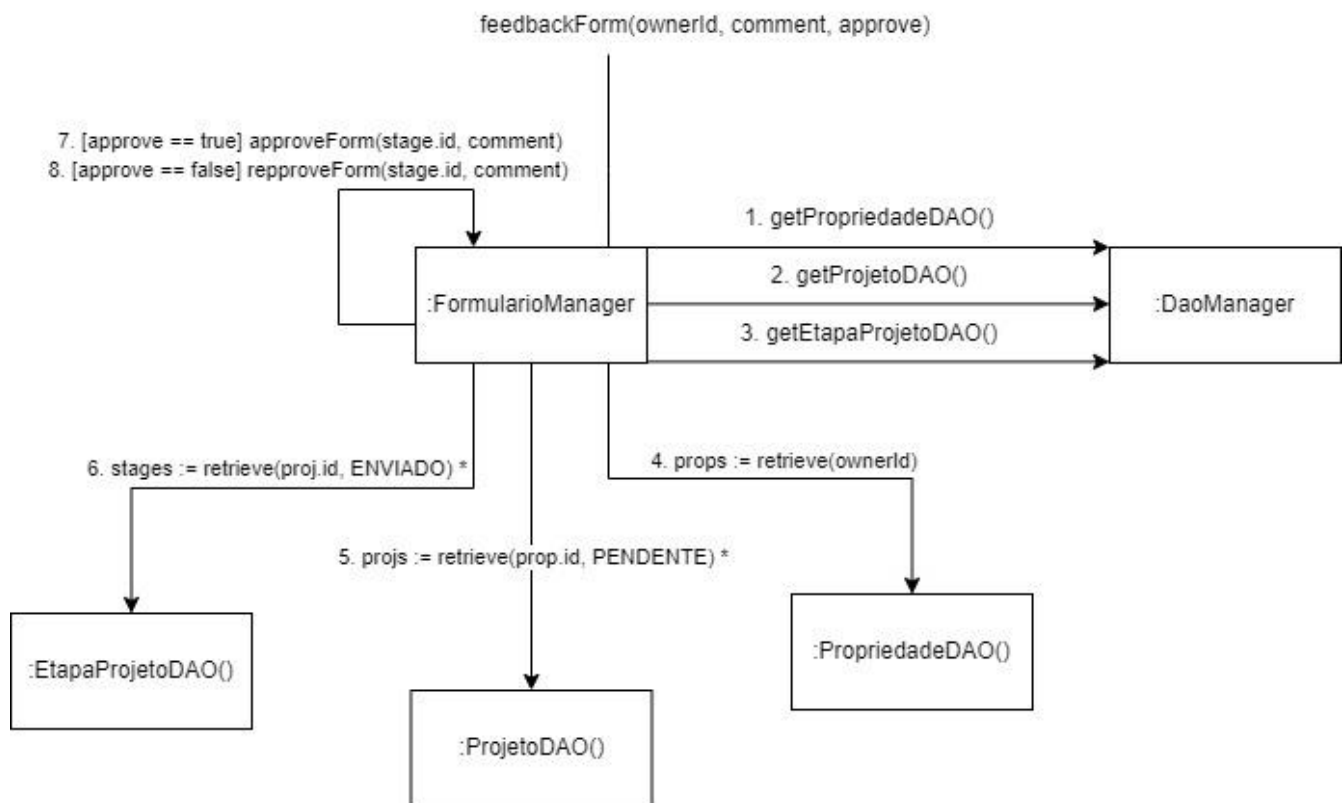
7.6 Montar formulário



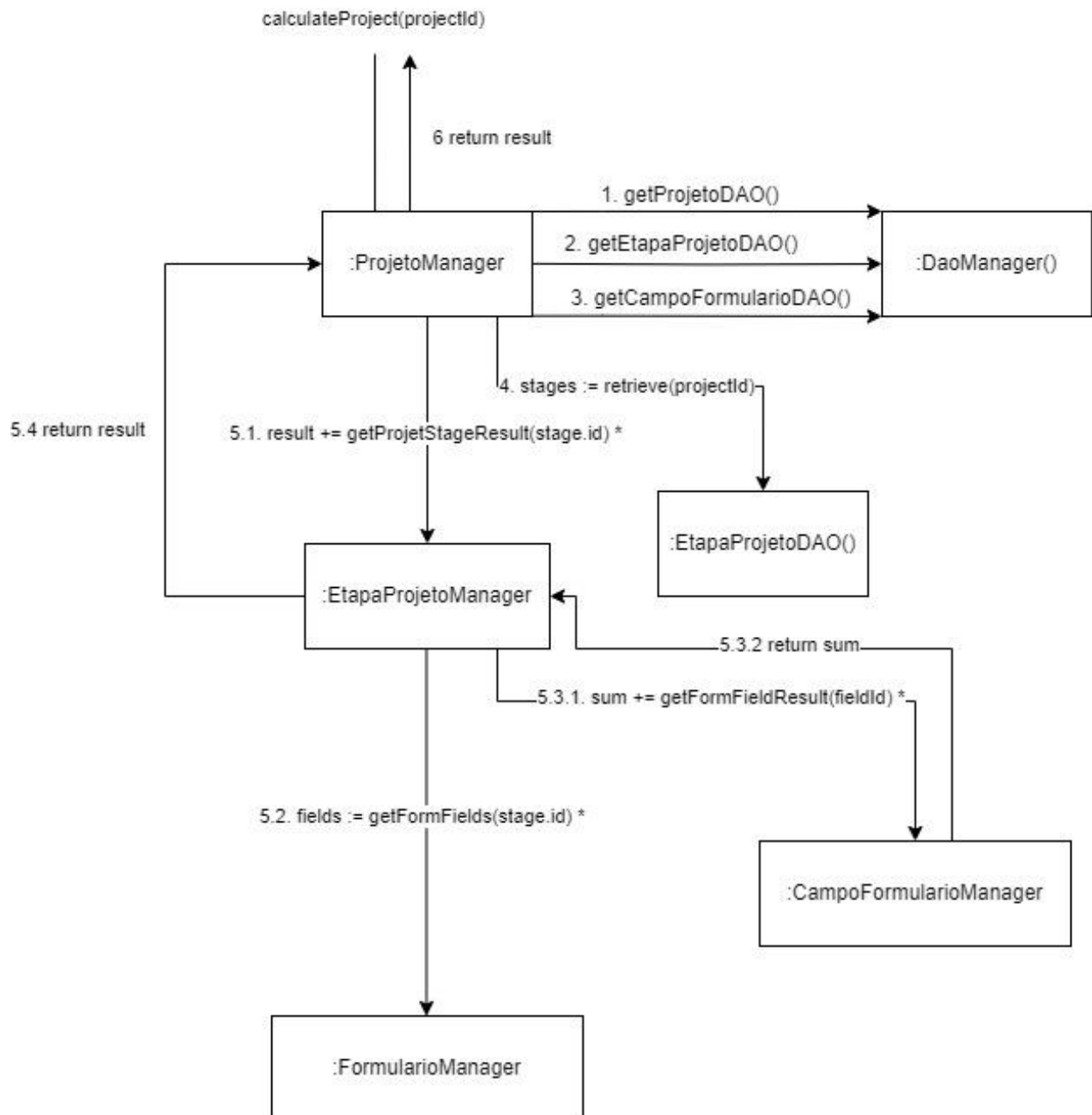
7.7 Responder formulário



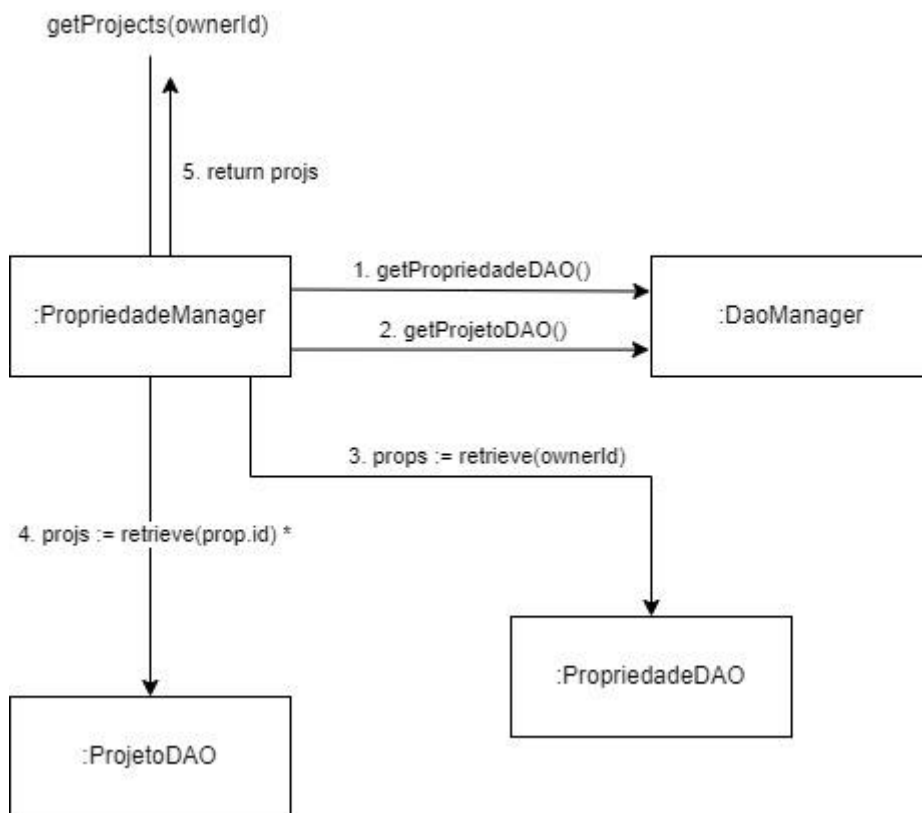
7.8 Enviar feedback do formulário



7.9 Calcular resultados



7.10 Ver projetos



8. Diagrama de classe do projeto

