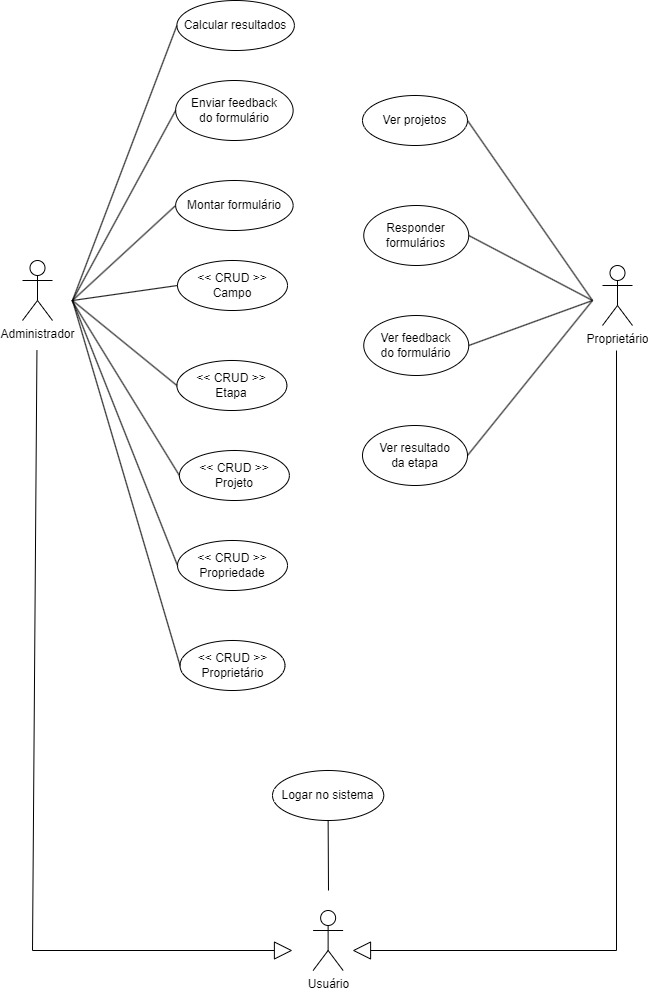
# Descrição do Sistema

## 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 idéias do paradigma orientado à 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 avaliára, 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.

# Diagrama de casos de uso



# 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.

## 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 responde-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. 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:**

* 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;

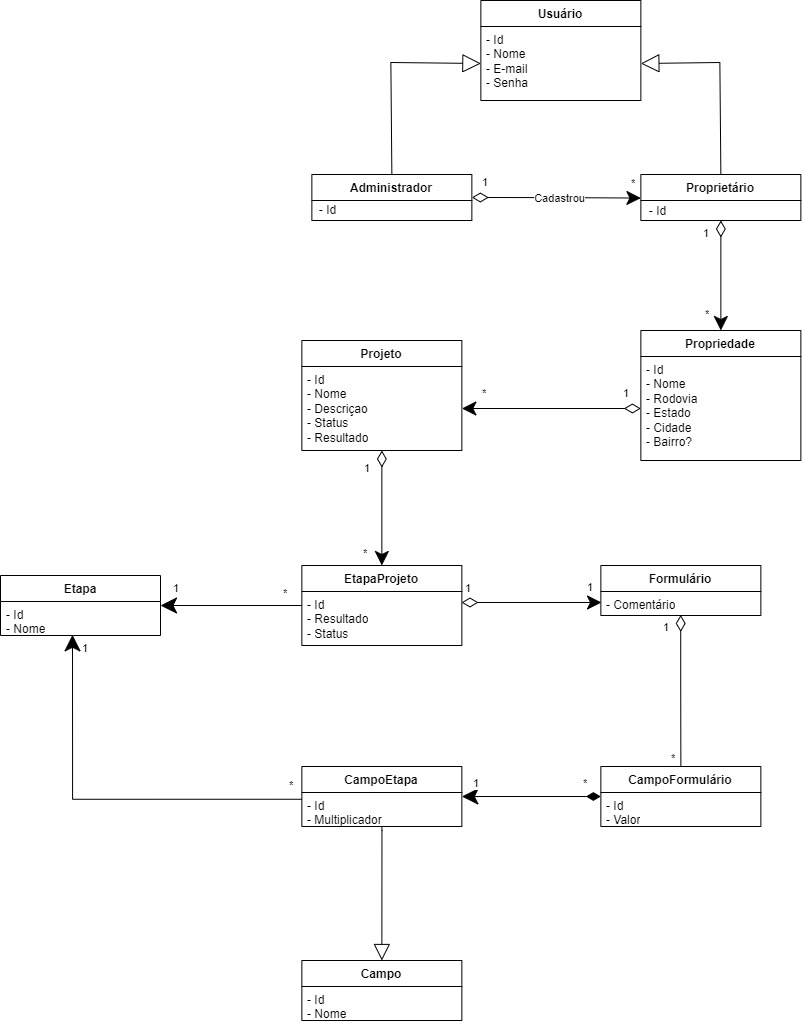
10. O sistema calcula o resultado de todas as etapas e grava o resultado no projeto o qual será informado imediatamente para o Administrador na tela.

**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);

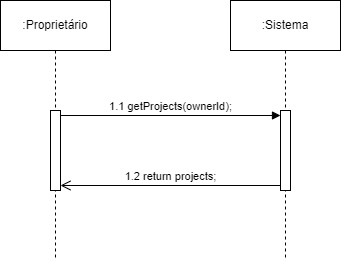
9.2 O Sistema informará um erro e encerrará o caso de uso.

# Modelo conceitual

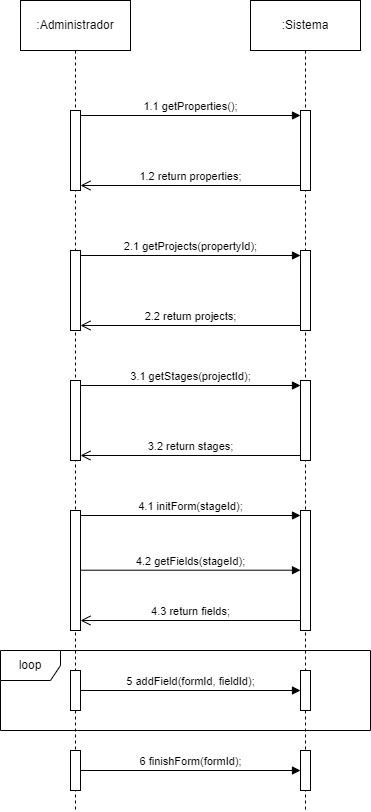


# Diagrama de sequência do sistema

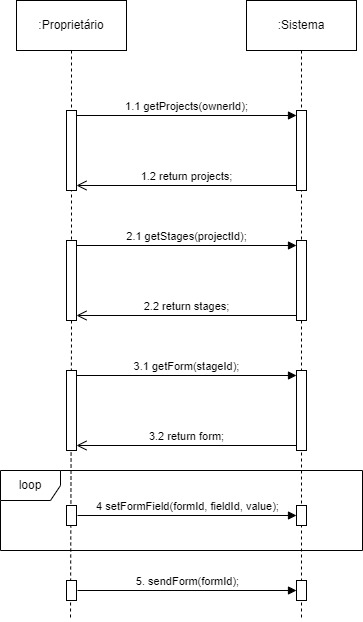
## Ver projetos



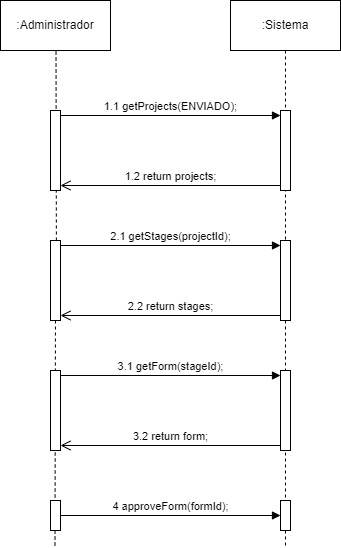
## Montar formulário



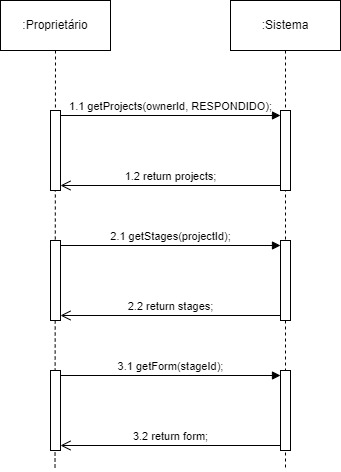
## Responder formulário



## Enviar feedback do formulário



## Ver feedback do formulário



## Calcular resultados

