



**Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"**

**MATHEUS FRANCISCO ASSMANN DE FREITAS**

**SISTEMA PARA GERENCIAMENTO DA PROVA DE TRÊS TAMBORES**

**Assis/SP  
2020**



**Fundação Educacional do Município de Assis  
Instituto Municipal de Ensino Superior de Assis  
Campus "José Santilli Sobrinho"**

**MATHEUS FRANCISCO ASSMANN DE FREITAS**

## **SISTEMA PARA GERENCIAMENTO DA PROVA DE TRÊS TAMBORES**

Trabalho de Conclusão de Curso de Análise e Desenvolvimento de Sistemas, do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

**Orientando: Matheus Francisco Assmann de Freitas  
Orientador: Dr. Almir Rogério Camolesi**

**Assis/SP  
2020**

#### FICHA CATALOGRÁFICA

FREITAS, Matheus Francisco Assmann.

**Sistema para Gerenciamento da Prova de Três Tambores** / Matheus Francisco Assmann de Freitas. Fundação Educacional do Município de Assis – FEMA – Assis, 2020.

Número de páginas.

Orientador: Dr. Almir Rogério Camolesi

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis - IMESA

1. Visual Studio. 2.MySQL. 3.Três Tambores

CDD:  
Biblioteca da FEMA

# SISTEMA PARA GERENCIAMENTO DA PROVA DE TRÊS TAMBORES

MATHEUS FRANCISCO ASSMANN DE FREITAS

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso Superior de Análise e Desenvolvimento de Sistemas, avaliado pela seguinte comissão examinadora:

**Orientador:** \_\_\_\_\_  
Dr. Almir Rogério Camolesi

**Examinador:** \_\_\_\_\_  
Dr. Luiz Ricardo Begosso

Assis/SP  
2020

## DEDICATÓRIA

Dedico este trabalho a minha mãe, Ingrid, por sempre ter me apoiado nas horas difíceis e por ter sempre me incentivado. Ao meu irmão Thiago, pelo exemplo de pessoa e profissional, e a minha irmã, Kalinka, por sempre me fazer enxergar as coisas boas em momentos de dificuldades. Ao meu padrasto Stefan, por sempre me aturar e pelas brincadeiras. A minha noiva, Camilla, por sempre estar ao meu lado, e por me tornar um homem melhor, a cada dia que passa. A todos os meus queridos amigos, por sempre acreditarem em meu potencial e, finalmente, ao Dr. Almir, professor e amigo, por sempre me fazer acreditar na minha capacidade e sempre me incentivar a dar sempre o meu melhor. Muito obrigado a todos vocês.

## RESUMO

O trabalho apresentado descreve o desenvolvimento de um software para o gerenciamento da prova de três tambores. Tal sistema visa controlar as informações de cadastros de animais, competidores, campeonatos, etapas, categorias e todas as informações relevantes para o processo. Além disso, também terá como funcionalidade a gestão completa da prova, que consiste desde os cadastros básicos necessários a realização da mesma, até a geração de classificações e rankings. Este sistema poderá emitir relatórios, para facilitar e agilizar processos classificatórios durante ou após o término das provas. Por fim deve permitir a geração de imagens para equipamentos audiovisuais em eventos que disponibilizam de tais equipamentos durante a sua realização e/ou para provas que são transmitidas pela internet. As ferramentas utilizadas na elaboração deste trabalho foram: Visual Studio 2019, MySql, Astah UML, DrawIO.

**Palavras-chave:** Visual Studio 2019; MySQL; Três Tambores;

## **ABSTRACT**

The presented paper describes the development of software for the management of barrel racing competition. Such a system aims to control the information of animal registrations, competitors, championships, stages, categories and all relevant information for the process. In addition, it will also have the functionality to manage the entire event, which consists of the basic records necessary to carry out the race, generation of classifications and rankings. This system will be able to issue reports, to facilitate and speed up classification processes during or after the end of the events. Finally, it must allow the generation of images for audiovisual equipment in events that possess such equipment available during its realization and / or for events that are transmitted over the internet. The tools used in the elaboration of this project were: Visual Studio 2019, MySql, Astah UML, DrawIO.

**Keywords:** Visual Studio 2019; MySQL; Barrel Racing;

## LISTA DE ILUSTRAÇÕES

Figura 1 - Estrutura Analítica do Projeto.....	22
Figura 2 - Três Tambores .....	23
Figura 3 - Percurso Três Tambores.....	23
Figura 4 - Diagrama de Caso de Uso Geral.....	27
Figura 5 - UC 1: Manter Animal .....	28
Figura 6 - Diagrama de Sequência: Manter Animal .....	30
Figura 7 - UC 2: Manter Banco .....	31
Figura 8 - Diagrama de Sequência: Manter Banco.....	33
Figura 9 - UC 3: Manter Campeonato.....	34
Figura 10 - Diagrama de Sequência: Manter Campeonato.....	36
Figura 11 - UC 4: Manter Categoria.....	37
Figura 12 - Diagrama de Sequência: Manter Categoria .....	39
Figura 13 - UC 5: Manter Cidade.....	40
Figura 14 - Diagrama de Sequência: Manter Cidade.....	42
Figura 15 - UC 6: Manter Competidor.....	43
Figura 16 - Diagrama de Sequência: Manter Competidor.....	45
Figura 17 - UC 7: Manter Etapa .....	46
Figura 18 - Diagrama de Sequência: Manter Etapa.....	48
Figura 19 - UC 8: Manter Tipo Penalidade .....	49
Figura 20 - Diagrama de Sequência: Manter Tipo Penalidade .....	51
Figura 21 - UC 9: Gerar Relatórios .....	52
Figura 22 - Diagrama de Sequência: Gerar Relatórios.....	54
Figura 23 – UC 10 - Gerar Relatórios Gerais.....	55
Figura 24 - Diagrama de Sequência: Gerar Relatórios Gerais .....	56

Figura 25 - UC 11 - Gerar Relatório Etapa .....	57
Figura 26 - Diagrama de Sequência: Gerar Relatório Etapa .....	58
Figura 27 - UC 12: Gerar Backup .....	59
Figura 28 - Diagrama de Sequência: Gerar Backup .....	61
Figura 29 - Diagrama de Classes .....	62
Figura 30 - Diagrama de Atividades: Gerar Passada .....	63
Figura 31 - Diagrama Entidade Relacionamento .....	64
Figura 32 - Estrutura do Projeto.....	65
Figura 33 - Tela Inicial SGTG .....	66
Figura 34 - Cadastro Animal.....	67
Figura 35 - Cadastro de Banco.....	68
Figura 36 - Cadastro Competidor 1 .....	69
Figura 37 - Cadastro Competidor 2 .....	69
Figura 38 - Cadastro Competidor 3 .....	70
Figura 39 - Cadastro Competidor 4 .....	70
Figura 40 - Cadastro Competidor 5 .....	71
Figura 41 - Cadastro de Campeonato .....	72
Figura 42 - Cadastro de Categoria .....	73
Figura 43 - Cadastro de Etapa.....	74
Figura 44 - Cadastro de Etapa 1.....	75
Figura 45 - Cadastro de Etapa 2.....	76
Figura 46 - Cadastro de Etapa 3.....	77
Figura 47 - Código Antes Passada .....	78
Figura 48 - gerarInicioHtmlBanner() .....	79
Figura 49 - gerarFimHtmlBanner().....	79
Figura 50 - Trecho antesMontariaHtml().....	80

Figura 51 - Trecho depoisMontariaHtml() .....	80
Figura 52 - Antes Passada 1 .....	81
Figura 53 - Depois Passada 1 .....	82
Figura 54 - Segunda Passada .....	82
Figura 55 - Antes Passada 2 .....	83
Figura 56 - Depois Passada 2 .....	84
Figura 57 - Terceira Passada .....	85
Figura 58 - Antes Passada 3 .....	86
Figura 59 - Depois Passada 3 .....	86
Figura 60 - Tela Classificação .....	87

## LISTA DE TABELAS

Tabela 1 - Lista de Eventos .....	26
Tabela 2 - Narrativa UC 1: Manter Animal .....	29
Tabela 3 - Narrativa UC 2: Manter Banco .....	32
Tabela 4 - Narrativa UC 3: Manter Campeonato .....	35
Tabela 5 - Narrativa UC 4: Manter Categoria .....	38
Tabela 6 - Narrativa UC 5: Manter Cidade .....	41
Tabela 7 - Narrativa UC 6: Manter Competidor .....	44
Tabela 8 - Narrativa UC 7: Manter Etapa .....	47
Tabela 9 - Narrativa UC 8: Manter Tipo Penalidade .....	50
Tabela 10 - Narrativa UC 9: Gerar Relatórios .....	53
Tabela 11 - Narrativa UC 10: Gerar Relatórios Gerais .....	56
Tabela 12 - Narrativa UC 11: Gerar Relatório Etapa .....	58
Tabela 13 - Narrativa UC 12: Gerar Backup .....	60

## **LISTA DE ABREVIATURAS E SIGLAS**

SGTT – Sistema de Gerenciamento

CRP – Circuito Rancho Primavera

SI – Sistema de Informação

UML – Unified Modeling Language (Linguagem de Modelagem Unificada)

IDE – Integrated Development Environment (Ambiente de Desenvolvimento Integrado)

.NET – Dot NET

C++ - C PLUS PLUS ( C MAIS MAIS)

C# - C SHARP

SQL – System Query Language (Linguagem de Consulta Estruturada)

SGBD – Sistema de Gerenciamento de Banco de Dados

DB – Database (Banco de Dados)

DER – Diagrama Entidade Relacionamento

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>15</b>
1.1. OBJETIVOS .....	16
1.2. JUSTIFICATIVAS .....	16
1.3. MOTIVAÇÕES .....	17
1.4. ESTRUTURA DO TRABALHO.....	17
<b>2. REVISÃO BIBLIOGRÁFICA .....</b>	<b>18</b>
2.1. MÉTODO DE ANÁLISE.....	18
2.2. SISTEMAS DESKTOP.....	18
2.3. FERRAMENTAS UTILIZADAS PARA ANÁLISE .....	19
2.3.1. Astah UML .....	20
2.3.2. Draw.IO.....	20
2.4. FERRAMENTAS UTILIZADAS PARA O DESENVOLVIMENTO .....	20
2.4.1. Microsoft Visual Studio 2019 .....	20
2.4.2. C# .....	21
2.4.3. MySQL.....	21
2.4.4. Entity Framework.....	21
<b>3. ANÁLISE E PROJETO DO SISTEMA .....</b>	<b>22</b>
3.1. SISTEMA DE GESTÃO DE TRÊS TAMBORES .....	23
3.2. DETALHAMENTOS DO PROBLEMA A SER RESOLVIDO .....	24
3.3. RESULTADOS ESPERADOS NA IMPLEMENTAÇÃO.....	25
3.4. FORMA ADOTADA PARA LEVANTAMENTO DOS REQUISITOS .....	25
3.5. RESTRIÇÕES DE DESENVOLVIMENTO DO SOFTWARE.....	25
3.6. PROBLEMAS POTENCIAIS .....	25
3.7. PRIORIZAÇÃO DA IMPLANTAÇÃO DOS REQUISITOS .....	25
3.8. LISTA DE EVENTOS.....	26
3.9. DIAGRAMA.....	26
3.9.1. Diagrama de Caso de Uso .....	27
3.10. ESPECIFICAÇÃO DOS CASOS DE USO .....	27
3.10.1. Manter Animal .....	28
3.10.2. Manter Banco .....	31
3.10.3. Manter Campeonato .....	34

3.10.4.	Manter Categoria.....	37
3.10.5.	Manter Cidade .....	40
3.10.6.	Manter Competidor .....	43
3.10.7.	Manter Etapa .....	46
3.10.8.	Manter Tipo Penalidade.....	49
3.10.9.	Gerar Relatórios.....	52
3.10.10.	Gerar Relatórios Gerais .....	55
3.10.11.	Gerar Relatório Etapa.....	57
3.10.12.	Gerar Backup .....	59
3.11.	DIAGRAMA DE CLASSE .....	62
3.12.	DIAGRAMA DE ATIVIDADES .....	63
3.12.1.	Gerar Passada.....	63
3.13.	DIAGRAMA ENTIDADE-RELACIONAMENTO (DER) .....	64
<b>4.</b>	<b>IMPLEMENTAÇÃO .....</b>	<b>65</b>
4.1.	ORGANIZAÇÃO DO PROJETO .....	65
4.2.	TELA INICIAL .....	66
4.3.	CADASTRO ANIMAL .....	67
4.4.	CADASTRO DE BANCO.....	68
4.5.	CADASTRO DE COMPETIDOR.....	69
4.5.1.	CONTINUAÇÃO CADASTRO DE COMPETIDOR 1 .....	69
4.5.2.	CONTINUAÇÃO CADASTRO DE COMPETIDOR 2.....	70
4.5.3.	CONTINUAÇÃO CADASTRO DE COMPETIDOR 3.....	70
4.5.4.	CONTINUAÇÃO CADASTRO DE COMPETIDOR 4.....	71
4.6.	CADASTRO DE CAMPEONATO.....	72
4.7.	CADASTRO DE CATEGORIA .....	73
4.8.	CADASTRO DE ETAPA.....	74
4.8.1.	Continuação cadastro de Etapa 1.....	75
4.8.2.	Continuação cadastro de Etapa 2.....	76
4.8.3.	Continuação cadastro de Etapa 3.....	77
4.8.4.	Tela Antes Passada .....	81
4.8.5.	Tela Depois Passada .....	81
4.8.6.	Segunda Passada .....	82
4.8.7.	Tela Antes Passada 2 .....	83
4.8.8.	Tela Depois Passada 2 .....	84

4.8.9. Terceira Passada (Final).....	85
4.8.10. Antes Passada 3 .....	86
4.8.11. Depois Passada 3 .....	86
4.8.12. Tela Classificação.....	87
<b>5. CONCLUSÃO .....</b>	<b>88</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>90</b>
<b>6. ANEXOS .....</b>	<b>91</b>
6.1. ANEXO A – CRONOGRAMA DE TAREFAS REALIZADAS.....	91
6.2. CÓDIGO DE NOVAPASSADA.....	91
6.2.1. Primeira Parte.....	91
6.2.2. Segunda Parte .....	92
6.2.3. Terceira Parte .....	93
6.2.4. Quarta Parte.....	94
6.2.5. Quinta Parte.....	95

## 1. INTRODUÇÃO

Segundo LOH (2014) um Sistema de Informação (SI) é um sistema cujo elemento principal é a informação. Já um Sistema, segundo a Teoria Geral de Sistemas, é um conjunto de elementos interrelacionados com um objetivo em comum. O objetivo principal de um SI é gerenciar informações, de tal modo a apoiar decisões e processos de uma organização.

Sistema de informação é usado para descrever um sistema que possui algum tipo de automação, seja manual, abrangendo pessoas, ou automático, abrangendo máquinas e tem como objetivo coletar, processar e transmitir dados que tenham algum valor para o usuário final. Dado nada mais é que o resultado da coleta e do processamento das informações que são oferecidas ao usuário da forma que ele julgar útil e significativa (LOH, 2014).

Com o passar dos anos, e com o avanço contínuo e desenfreado de várias tecnologias, pode se afirmar que empresas que não fazem uso dessas, podem comprometer sua produtividade, eficiência e agilidade, além de dificultar rotinas simples para seus colaboradores e outras partes envolvidas diretamente com a empresa.

A gestão com auxílio de sistemas computacionais é cada vez mais necessária, já que visa melhorar a eficiência das rotinas administrativas, ajudando a reduzir a quantidade de erros e de fornecer informações mais precisas.

O Circuito Rancho Primavera (CRP)<sup>1</sup> vem se destacando cada vez mais com seu profissionalismo e competência na organização e realização de rodeios por todo o país. Com a popularização do torneio de Três Tambores, viram a oportunidade de desenvolvimento de um sistema capaz de gerenciar todas as etapas da prova, que engloba desde simples cadastros de competidores e animais, até a emissão de relatórios detalhados.

---

<sup>1</sup> CRP – Circuito Rancho Primavera. Disponível em: <http://circuitoranchoprivavera.com.br/index.asp>

## 1.1. OBJETIVOS

Conforme apresentado, a informatização do Sistema de Gerenciamento de Provas de Três Tambores (SGTT) será organizada, basicamente em três módulos. Que serão divididos da seguinte forma: Cadastros, Gerenciamento de Prova e Emissão de Relatórios.

O primeiro módulo, Cadastros englobará quaisquer tipos de informações que sejam relevantes para realização da prova. Podemos citar os cadastros referentes a os competidores, animais, patrocinadores e etc.

O segundo módulo, Gerenciamento de Prova, será o mais complexo de todos. Consistirá em gerenciar todos os tempos, ordem, penalizações e classificações que ocorrem durante a realização da prova. Tais informações devem ser apresentadas em tempo real durante a prova em equipamentos audiovisuais.

Por fim, o último módulo, de Emissão de Relatórios, consistirá desde relatórios simples como relação de competidores e animais cadastrados em determinada prova, a até um relatório final relacionado a determinada etapa, onde constará todos os tempos, classificação, ranking e etc. Informações estas muitíssimas relevantes, para a platéia, para os organizadores do evento, como para os competidores, devido a premiação e etc.

O software será desenvolvido com base em plataforma Desktop<sup>2</sup>, em um servidor local, devido ao local das provas dificilmente ter um bom acesso à internet. Para a codificação do projeto será utilizada a linguagem de programação C# e a base de dados utilizada para o armazenamento local será o Banco de Dados MySQL.

## 1.2. JUSTIFICATIVAS

O CRP, graças a seu crescimento no segmento de eventos/rodeios, também ministra agora torneios relacionados a prova de Três Tambores, e graças a isso, decidiu-se iniciar um investimento para o desenvolvimento de um sistema próprio para o

---

<sup>2</sup> Do inglês, significa “em cima da mesa”. Era frequentemente utilizado para se referir a um computador de mesa. Desenvolvimento foca-se em aplicações que serão executadas em máquinas locais com equipamentos locais (memória, disco rígido, processador, etc.), sem a necessidade de Internet ou armazenamento em nuvem.

gerenciamento deste tipo de prova. Por já possuírem um sistema para o gerenciamento de rodeios, as plataformas escolhidas para o desenvolvimento desse novo sistema, tem como objetivo a integração entre eles, para tornar-se um sistema mais completo.

### **1.3. MOTIVAÇÕES**

Atualmente, toda a estrutura para gerenciamento de uma prova de Três Tambores, envolve planilhas e diversas anotações em próprio punho para documentar o tempo de cada competidor. No caso de um evento que participem diversos competidores diferentes, mantêr o ritmo de informações sobre tempo é demasiadamente exaustivo, e cobra muito das partes envolvidas, causando desgaste físico e mental. Vale salientar também, que a prova é muito dinâmica e possui poucas paradas para descanso, geralmente quando é necessária a manutenção da arena onde é realizada a prova.

Outra dificuldade também é de mantêr sempre os tempos atualizados e informar ao público quem lidera a prova em determinado momento, ou informar o tempo de corte, quando no caso existe uma quantidade determinada de competidores que irá passar para uma próxima fase na etapa.

Com os principais problemas descritos acima, viu-se a oportunidade para desenvolvimento de um software capaz de gerenciar todas as etapas da prova, utilizando essa oportunidade também para aprendizado e estudo.

### **1.4. ESTRUTURA DO TRABALHO**

O trabalho foi dividido em 6 capítulos. No primeiro, foi realizada a introdução sobre o projeto, esclarecendo quais os objetivos pretendidos, a justificativa e a motivação para a realização do mesmo. No segundo, são abordadas as tecnologias que serão utilizadas no desenvolvimento. O terceiro, mostra a estrutura analítica do projeto e as etapas de elaboração. O quarto, o levantamento e análise de requisitos identificados. No quinto, os diagramas e por fim, no sexto, a conclusão da atual etapa deste trabalho.

## 2. REVISÃO BIBLIOGRÁFICA

Neste capítulo serão apresentados os métodos, ferramentas e o ambiente utilizados para o desenvolvimento deste trabalho.

### 2.1. MÉTODO DE ANÁLISE

Ao se falar atualmente sobre metodologias utilizadas para documentação de software, o SCRUM<sup>3</sup> destaca-se, por se tratar de uma metodologia ágil e que caminha juntamente com o desenvolvimento do código e por possuir uma documentação mais enxuta.

O UML (Unified Modeling Language), Linguagem de Modelagem Unificada, é uma linguagem padrão para elaborar a estrutura de um projeto de sistemas. Segundo PEREIRA E SILVA (2009), o UML nos ajuda a definir as partes de um sistema orientado a objetos, nos ajudando a modelá-lo e documentá-lo. Busca padronizar a parte inicial do desenvolvimento, por ser muito simples e intuitiva.

O UML e o SCRUM caminham em sentidos opostos, já que o UML tem como objetivo o foco em uma boa documentação, enquanto o SCRUM foca-se no desenvolvimento do código.

No início do desenvolvimento deste trabalho, a codificação foi priorizada, e foi adotado assim o método SCRUM, já que o software precisava ser posto a teste para averiguar seus pontos fortes e fracos. Depois de feitos esses testes, a atenção voltou-se para documentação completa (UML) para ajudar na melhor compreensão do código caso novos programadores necessitem melhorá-lo ou apenas para manutenção do código.

### 2.2. SISTEMAS DESKTOP

Sistema Desktop pode ser descrito como o sistema feito de um conjunto de programas executados em torno de um sistema operacional e que compartilha uma interface gráfica para o usuário. Segundo DOMINGUES (2014), o objetivo principal de

---

<sup>3</sup> Scrum - <https://www.desenvolvimentoagil.com.br/scrum/>

sistemas gerenciais, é de otimizar processos que geralmente são lentos e repetitivos, a fim de reduzir erros, tornando-os mais eficazes.

No caso do tema proposto com o SGTT, seria de cadastros de competidores e animais, emissão de relatórios de provas e etc.

Alguns outros benefícios advindos de um sistema Desktop, valem ser salientados:

- **Otimização:**

Tarefas que necessitariam uma determinada atenção para serem executadas, podem ser facilmente padronizadas com o sistema Desktop, tornando-as menos suscetíveis a erros e não tendo necessidade de revisão.

- **Economia:**

Por ser utilizado juntamente com o MySQL, e utilizando um servidor local, não existe quaisquer custo com hospedagem de servidores.

- **Segurança:**

Graças ao advento de um servidor local, o banco de dados fica mais seguro que aplicações normais. Ele fica isolado, somente tendo acesso as partes que gerenciam o sistema.

- **Personalização:**

Um sistema Desktop pode ser ajustado especificamente para as necessidades da empresa, tornando-o único e exclusivo.

Para todas as etapas de análise e desenvolvimento deste software, foram utilizadas as ferramentas que serão descritas nas próximas seções.

### **2.3. FERRAMENTAS UTILIZADAS PARA ANÁLISE**

Neste tópico serão descritas as ferramentas utilizadas para o método inicial de análise do sistema, a fim de facilitar o início do desenvolvimento.

### 2.3.1. Astah UML

O Astah UML<sup>4</sup> é uma plataforma que permite ao usuário criar todos os diagramas relevantes para documentação e estruturação de um projeto de software. Tendo como benefício o fato de disponibilizar uma licença gratuita para estudantes.

### 2.3.2. Draw.IO<sup>5</sup>

É um software livre online para confecção de fluxogramas, diagramas de processo, organogramas, UML, ER e diversos outros. Destaca-se pela sua facilidade de operacionalidade e capacidade de gravar arquivos em computador local ou em nuvem.

## 2.4. FERRAMENTAS UTILIZADAS PARA O DESENVOLVIMENTO

Aqui serão descritas as principais ferramentas que foram utilizadas para o desenvolvimento do projeto. Ferramentas essas, que foram escolhidas devido a sua facilidade de aprendizado, alta produtividade e baixos custos.

### 2.4.1. Microsoft Visual Studio 2019<sup>6</sup>

É o ambiente de desenvolvimento integrado (IDE) pertencente a Microsoft dedicado especialmente ao desenvolvimento de software utilizando o framework .NET e às linguagens C, C++ e C# (C Sharp). Dentre todas as facilidades que uma IDE proporciona para seu usuário, podemos destacar algumas delas:

- O grande aumento de produtividade, pois possui diversas ferramentas que auxiliam durante o desenvolvimento;
- A geração automática de códigos, recurso presente em todas as IDE's atuais, que sugere trechos de código ao usuário;
- A facilidade em realizar verificações de código e a correção de erros em tempo de desenvolvimento, permitindo a visualização destes antes da compilação;

---

<sup>4</sup> Astah - <http://astah.net>

<sup>5</sup> Draw.IO - <https://www.draw.io>

<sup>6</sup> Visual Studio - <https://visualstudio.microsoft.com/pt-br/vs/>

- Ferramenta completa, o desenvolvedor tem tudo o que precisa para auxiliá-lo durante todo o processo de desenvolvimento (escrever código, testar código e corrigir código);

### **2.4.2. C#**

Segundo STELLMAN & GREENE (2011), C# é uma linguagem que faz parte do conjunto de ferramentas oferecidas na plataforma .NET e tem como características sua simplicidade, robustez, orientação a objetos, fortemente tipada e altamente escalável a fim de permitir que uma mesma aplicação seja executada em diversos dispositivos, independente do seu tipo e de seu hardware.

### **2.4.3. MySQL<sup>7</sup>**

O MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD), que utiliza a linguagem SQL (System Query Language). Segundo MANZANO (2011), cada vez mais empresas aderem ao MySQL, pois possui servidor confiável, rápido e é de fácil utilização. Utilizando-o não somente em pequenas aplicações, como também em aplicações críticas.

### **2.4.4. Entity Framework<sup>8</sup>**

O Entity Framework faz parte das ferramentas do Microsoft Visual Studio, é uma poderosa ferramenta de persistência de dados. Com ela é possível trabalhar com dados em forma de objetos, sem ter que se preocupar com tabelas e colunas de um banco de dados. Utilizando o LINQ, é possível realizar consultas complexas de maneira fácil e simplificada, já que sua linguagem é muito fácil de se compreender, uma vez tendo um conhecimento básico de LAMBDA. Com isso temos um pacote capaz de lidar com grandes volumes de dados de maneira simples, e tendo uma redução significativa nas linhas de código.

---

<sup>7</sup> MySQL - <https://www.mysql.com>

<sup>8</sup> Entity Framework - <https://docs.microsoft.com/pt-br/ef/>

### 3. ANÁLISE E PROJETO DO SISTEMA

Segundo MATOS; BERMEJO; SALM JUNIOR (2010), uma declaração de escopo bem detalhada do projeto, deve ser bem preparada, para que as reais necessidades das partes envolvidas sejam analisadas e convertidas em requisitos. Entrevistas com pessoas que tem conhecimento sobre o gerenciamento de uma prova de Três Tambores é essencial, para que o sistema seja dinâmico o suficiente para acompanhar o ritmo da prova, sem esquecer também de algo que possua uma usabilidade simples, sem a necessidade de um treinamento extenso. Tendo isso em mente, as etapas de planejamento para o sistema proposto, pode ser observada na Figura 1 abaixo:

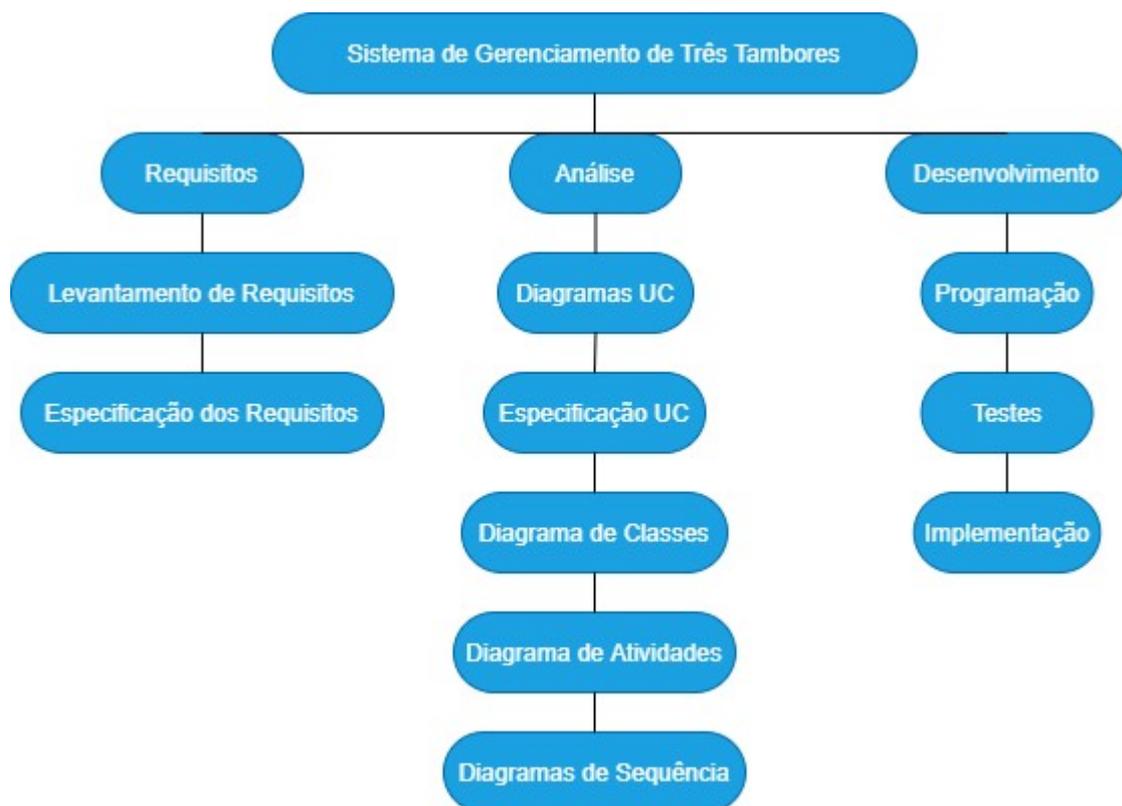


Figura 1 - Estrutura Analítica do Projeto

### 3.1. SISTEMA DE GESTÃO DE TRÊS TAMBORES

A prova de Três Tambores é o evento onde o cavalo e cavaleiro percorrem um circuito em forma de trevo, em torno de barris predefinidos no menor tempo possível. Esse tipo de prova vem se popularizando cada vez mais, por ter competidores das mais diversas idades, sendo tanto do sexo feminino como masculino. Na Figura 2, pode observar-se o momento em que o competidor passa pelo tambor.



Figura 2 - Três Tambores

Apesar de possuir diversas categorias, vale salientar que, profissionalmente, a prova é dominada por mulheres. Para maiores informações sobre as regras desta prova, é possível encontrar no site da ABQM<sup>9</sup>, todos os regulamentos existentes.

Pode-se visualizar o percurso da prova na Figura 3, abaixo:

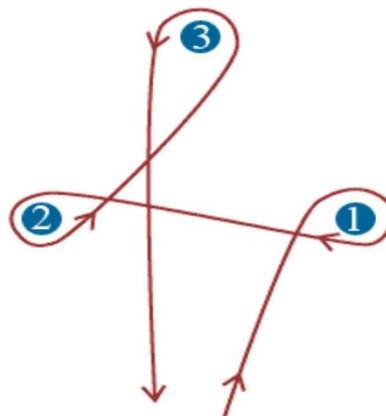


Figura 3 - Percurso Três Tambores

<sup>9</sup> Associação Brasileira de criadores de cavalo Quarto de Milha (<https://abqm.com.br>)  
[Regulamentos](#)

Geralmente, para se ter uma boa precisão acerca do tempo de início e término da passada, posiciona-se no início do trajeto um dispositivo de fotocélula, que ao ter seu feixe interrompido pela primeira vez, dispara um cronômetro, e ao término do percurso, interrompendo novamente o feixe, para o cronômetro. Neste tipo de prova é o equipamento de maior precisão, sendo utilizado sempre três casas decimais para controle dos tempos.

A prova é dividida, geralmente, em três passadas (três percursos), sendo que:

- Após a primeira e segunda passada, é contabilizado o MENOR tempo entre elas e assim gerado o ranking de classificação.
- Com esse ranking em mãos, será determinada a quantidade de participantes para a terceira passada (final).
- Finalizada a terceira passada, a classificação se dará pelo seguinte critério, o MENOR tempo das duas primeiras passadas SOMADO ao tempo da final.

Em casos de provas com finais a noite, esse é o critério mais utilizado, no caso a prova feminina (profissional). Em casos de provas com diversos participantes, como, por exemplo, free-horse<sup>10</sup>, juvenil, infantil e etc, são adotadas somente duas passadas e utilizado como critério de classificação o MENOR tempo dentre elas.

No caso, independente do tipo de categoria e prova determinada, todas seguem a seguinte regra:

- Um competidor poderá correr com vários cavalos diferentes, porém, um cavalo só poderá participar da prova uma única vez, com um único competidor.

### **3.2. DETALHAMENTOS DO PROBLEMA A SER RESOLVIDO**

Visando facilitar o controle e organização de uma prova de Três Tambores, surgiu-se a idéia da viabilização de um sistema capaz de gerenciar todas as etapas pertinentes desse evento.

---

<sup>10</sup> Categoria com o maior número de participantes, onde é permitida diversas montarias, para treino ou para averiguar o ritmo do conjunto (cavalo e competidor).

### **3.3. RESULTADOS ESPERADOS NA IMPLEMENTAÇÃO**

O sistema faz o gerenciamento completo de uma prova de Três Tambores, executando desde os cadastros das partes envolvidas, como também a geração de imagens para equipamentos audiovisuais, a fim de informar de maneira rápida e fácil determinado momento da prova para o público espectador, estando ele no local ou pela internet.

### **3.4. FORMA ADOTADA PARA LEVANTAMENTO DOS REQUISITOS**

Todo o levantamento de requisitos foi feito através de entrevistas com pessoas que participam do gerenciamento da prova. Consistindo, basicamente, em narradores e operadores.

### **3.5. RESTRIÇÕES DE DESENVOLVIMENTO DO SOFTWARE**

Para execução do sistema, serão necessários um computador e uma impressora para impressão dos relatórios.

### **3.6. PROBLEMAS POTENCIAIS**

Controlar Animais

Controlar Bancos

Controlar Campeonato

Controlar Categoria

Controlar Cidade

Controlar Competidor

Controlar Etapa

Controlar Tipo Penalidade

Emitir/Restaurar Backup

Emitir Relatórios

Emitir Imagens em Equipamentos Audiovisuais

### **3.7. PRIORIZAÇÃO DA IMPLANTAÇÃO DOS REQUISITOS**

Cadastro de animais, bancos, campeonato, categoria, cidade, competidor, etapa, tipo de penalidade.

Capacidade de realizar cópia e de restaurar Backup.

Emitir relatórios diversos referentes as etapas da prova.

Emitir imagens em equipamentos audiovisuais com as informações atuais da prova para o público.

### 3.8. LISTA DE EVENTOS

Nº	Descrição	Use Case
1	Usuário solicita cadastro de animais	Manter Animal
2	Usuário solicita cadastro de bancos	Manter Banco
3	Usuário solicita cadastro de campeonato	Manter Campeonato
4	Usuário solicita cadastro de categoria	Manter Categoria
5	Usuário solicita cadastro de cidade	Manter Cidade
6	Usuário solicita cadastro de competidor	Manter Competidor
7	Usuário solicita cadastro de etapa	Manter Etapa
8	Usuário solicita cadastro de tipo penalidade	Manter Tipo Penalidade
9	Usuário solicita relatório	Gerar Relatórios
10	Usuário solicita relatório geral	Gerar Relatórios Gerais
11	Usuário solicita relatório etapa	Gerar Relatório Etapa
12	Usuário solicita emitir/recuperar backup	Gerar Backup

Tabela 1 - Lista de Eventos

### 3.9. DIAGRAMA

Para modelagem do sistema, foram utilizados diagramas de caso de uso, de sequência, diagrama de Classe, diagrama de Atividades e diagrama de entidade-relacionamento (DER), que serão apresentados nas próximas seções deste capítulo.

### 3.9.1. Diagrama de Caso de Uso

Podemos observar, na Figura 4, a estrutura geral dos diagramas de caso de uso do sistema.

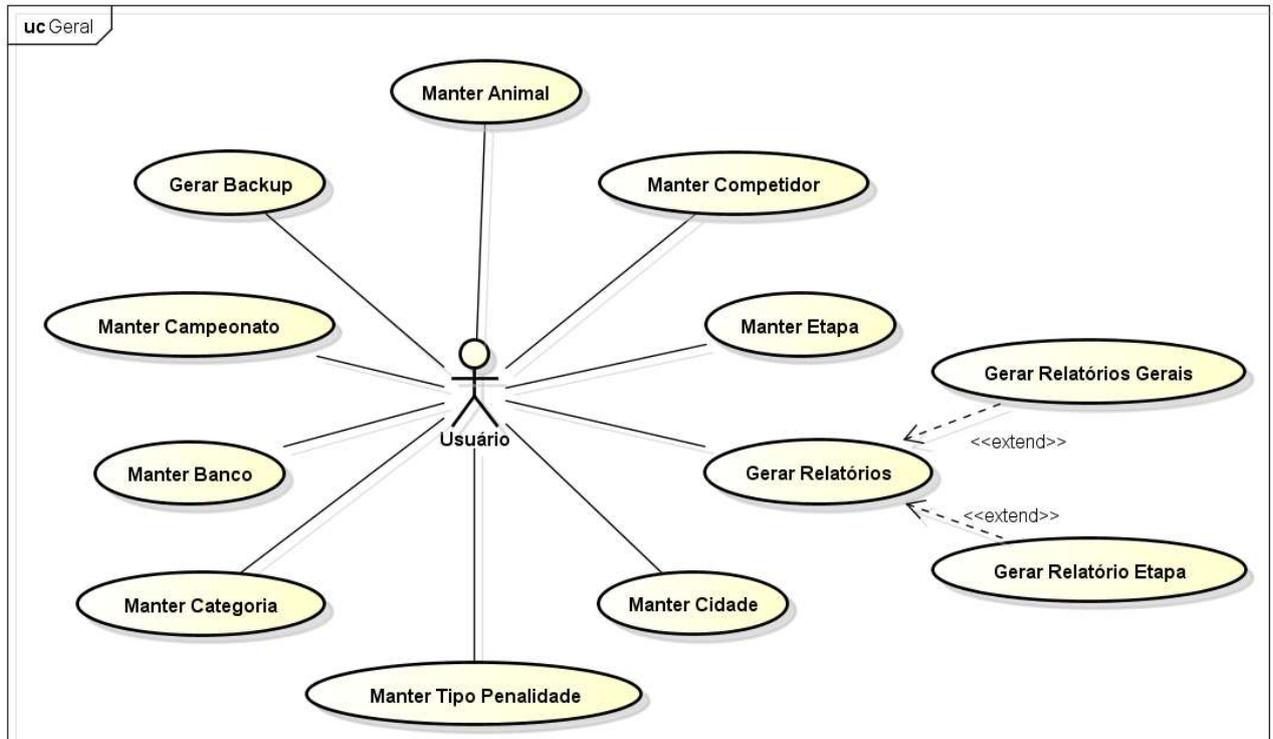


Figura 4 - Diagrama de Caso de Uso Geral

### 3.10. ESPECIFICAÇÃO DOS CASOS DE USO

Cada caso de uso possui uma funcionalidade específica, que serão apresentadas detalhadamente, a seguir.

### 3.10.1. Manter Animal

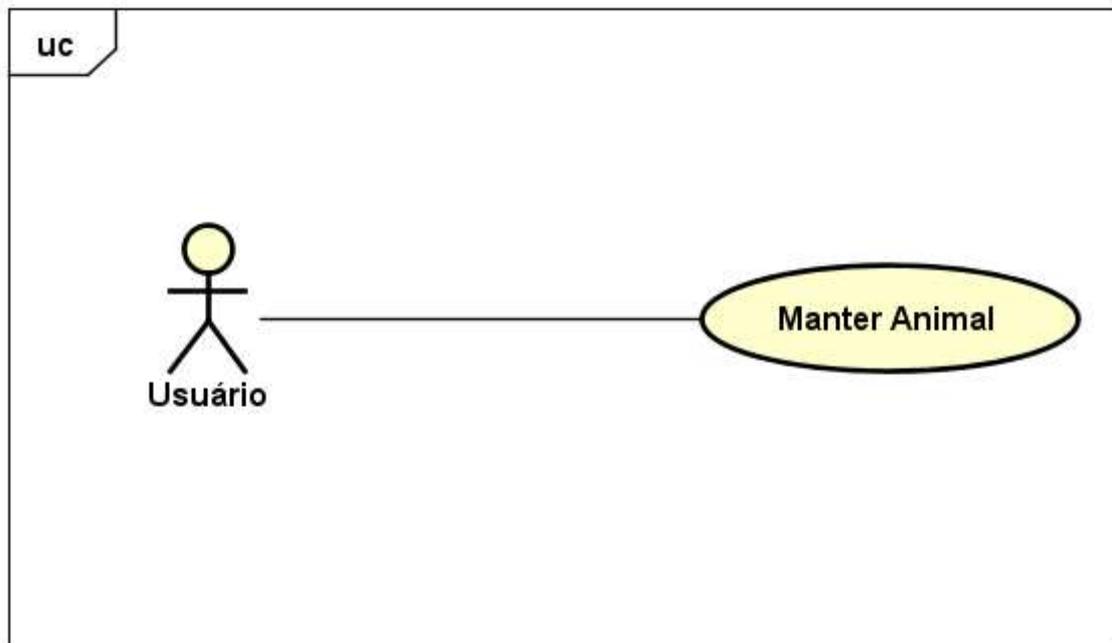
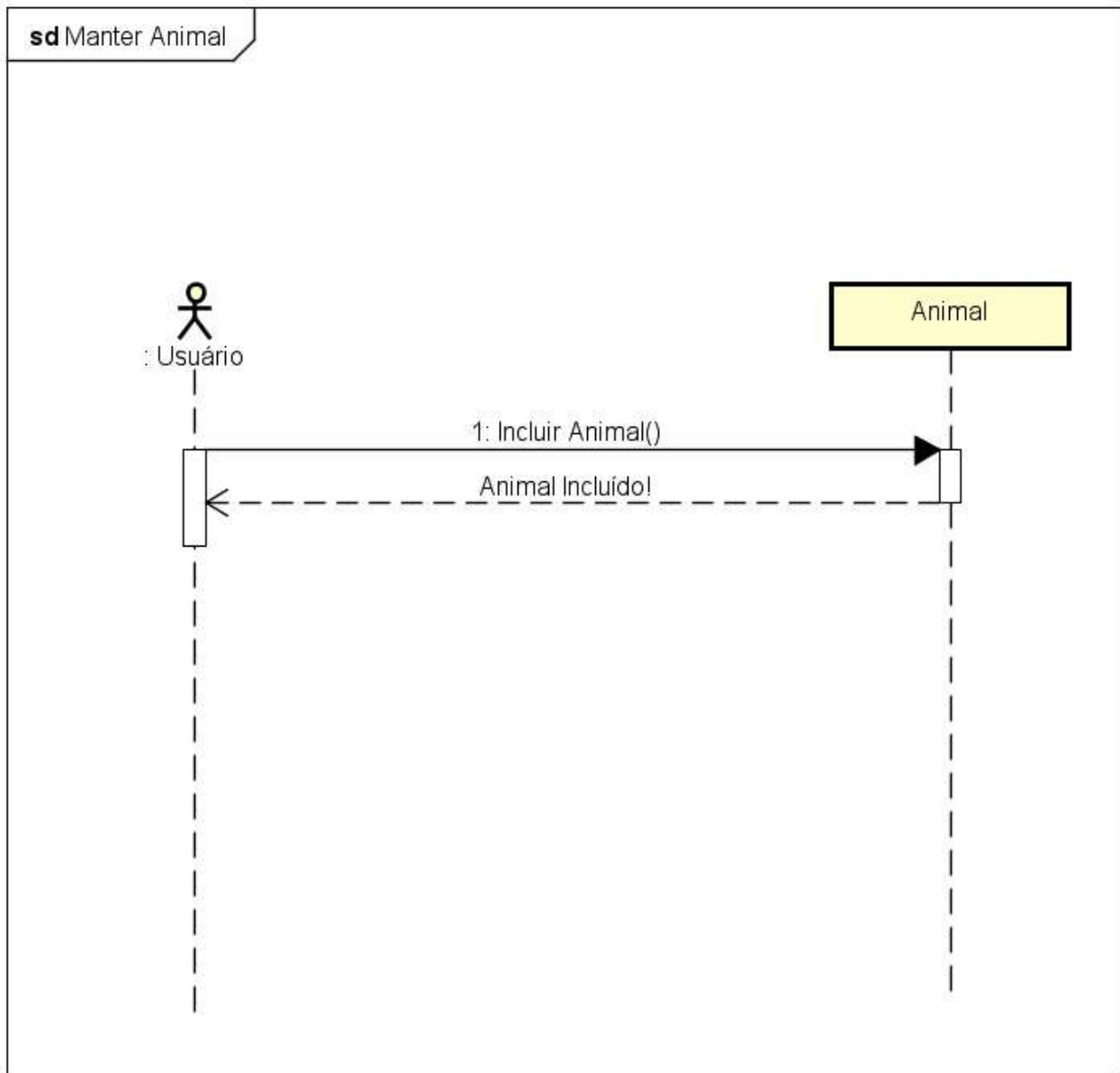


Figura 5 - UC 1: Manter Animal

Nome do UC 01	Manter Animal
Objetivo	Permitir ao usuário incluir, consultar ou alterar dados de animais.
Ator	Usuário
Pré-Condições	Não existentes.
Fluxo Principal	<p>1 - O usuário seleciona no menu Cadastros e em seguida seleciona Animal.</p> <p>2 - Uma nova janela abrirá, com as informações.</p> <p>3 - Caso queira incluir um novo Animal, o usuário deverá pressionar o botão "Novo" (A1).</p> <p>4 - Caso queira consultar por um Animal, o usuário deverá pressionar o botão "Pesquisar" (A2).</p>

	<p>5 - Caso queira alterar os dados de um Animal existente, o usuário deverá selecionar o Animal desejado (duplo clique) e em seguida pressionar o botão "Editar". (A3).</p> <p>6 - Caso o usuário queira finalizar essa janela, deverá pressionar o botão "Sair".</p>
Fluxo Alternativo	<p>A1 - O usuário digitará nos campos as informações requisitadas e em seguida pressionará o botão "Gravar".</p> <p>A1.1 - Se tudo estiver correto, receberá uma mensagem que o Animal foi incluído com sucesso.</p> <p>A2 - Um campo de pesquisa ficará habilitado, que permite ao usuário escolher o parâmetro que deseja procurar (nome, id, etc).</p> <p>A2.1 - Após digitado, o usuário deverá pressionar o botão "Pesquisar", na aba que foi aberta, e então será feita a filtragem dos dados.</p> <p>A3 - Os campos ficarão habilitados para que seja feita a alteração necessária.</p> <p>A3.1 - Concluída a edição, o usuário deverá pressionar o botão "Gravar" para que as alterações sejam salvas.</p>

**Tabela 2 - Narrativa UC 1: Manter Animal**



**Figura 6 - Diagrama de Sequência: Manter Animal**

### 3.10.2. Manter Banco

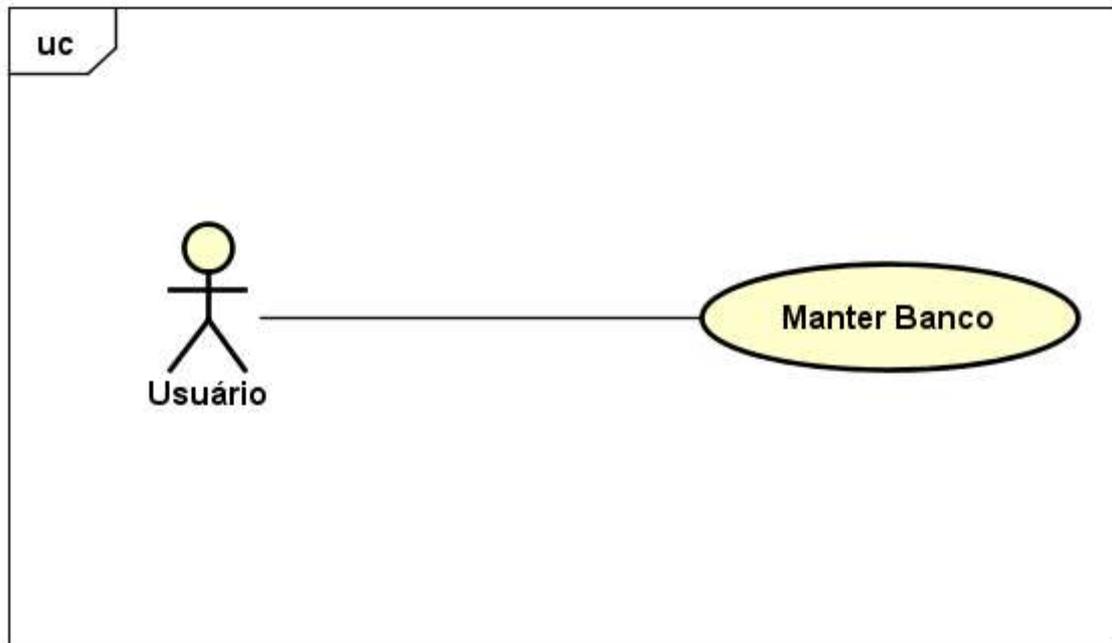


Figura 7 - UC 2: Manter Banco

Nome do UC 02	Manter Banco
Objetivo	Permitir ao usuário incluir, consultar ou alterar dados de um banco.
Ator	Usuário
Pré-Condições	Não existentes.
Fluxo Principal	<p>1 - O usuário seleciona no menu Cadastros e em seguida seleciona Banco.</p> <p>2 - Uma nova janela abrirá, com as informações.</p> <p>3 - Caso queira incluir um novo Banco, o usuário deverá pressionar o botão "Novo" (A1).</p> <p>4 - Caso queira consultar por um Banco, o usuário deverá</p>

	<p>pressionar o botão "Pesquisar" (A2).</p> <p>5 - Caso queira alterar os dados de um Banco existente, o usuário deverá selecionar o Banco desejado (duplo clique) e em seguida pressionar o botão "Editar". (A3).</p> <p>6 - Caso o usuário queira finalizar essa janela, deverá pressionar o botão "Sair".</p>
Fluxo Alternativo	<p>A1 - O usuário digitará nos campos as informações requisitadas e em seguida pressionará o botão "Gravar".</p> <p>A1.1 - Se tudo estiver correto, receberá uma mensagem que o Banco foi incluído com sucesso.</p> <p>A2 - Um campo de pesquisa ficará habilitado, que permite ao usuário escolher o parâmetro que deseja procurar (nome, id, etc).</p> <p>A2.1 - Após digitado, o usuário deverá pressionar o botão "Pesquisar", na aba que foi aberta, e então será feita a filtragem dos dados.</p> <p>A3 - Os campos ficarão habilitados para que seja feita a alteração necessária.</p> <p>A3.1 - Concluída a edição, o usuário deverá pressionar o botão "Gravar" para que as alterações sejam salvas.</p>

**Tabela 3 - Narrativa UC 2: Manter Banco**

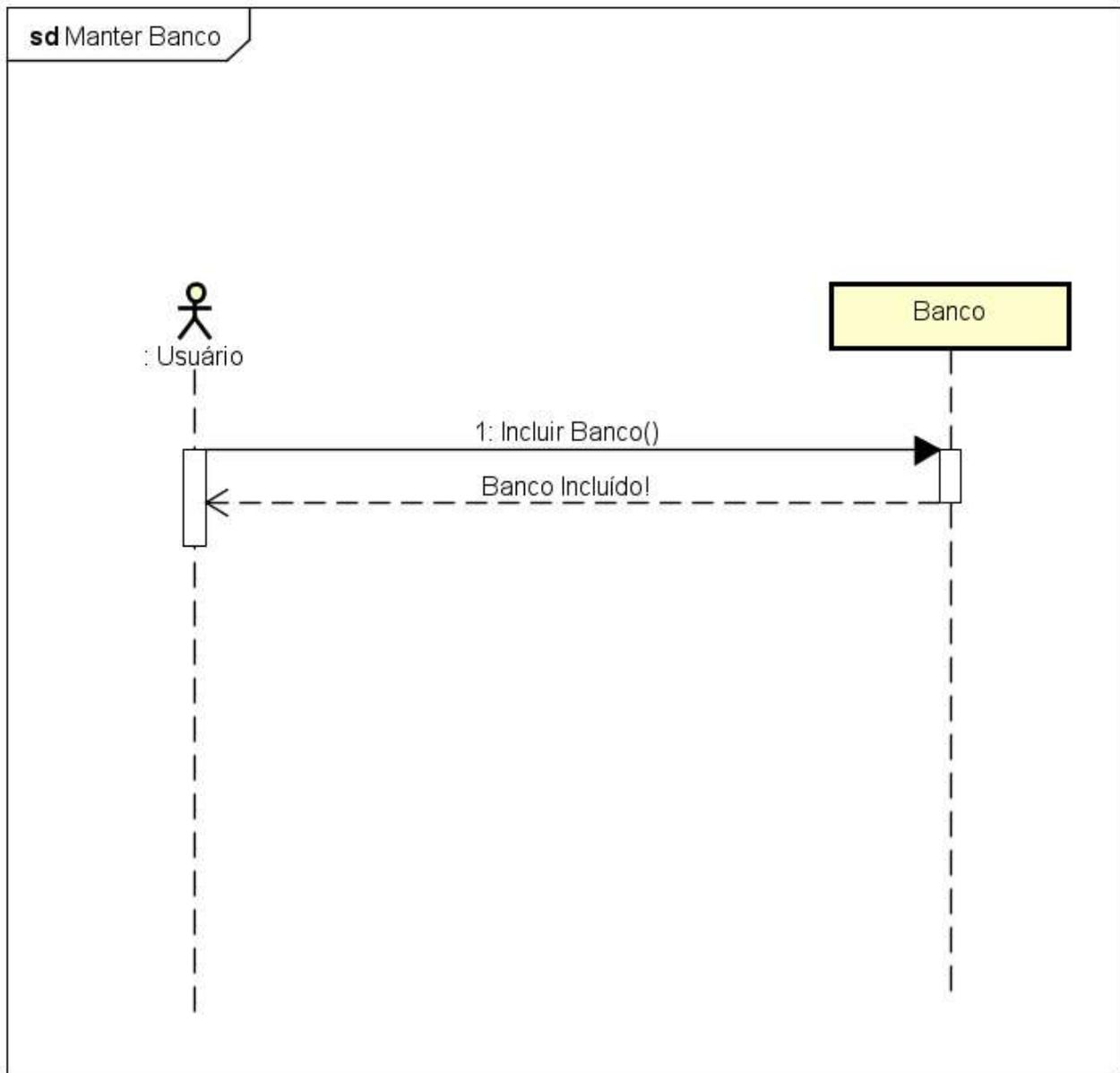


Figura 8 - Diagrama de Sequência: Manter Banco

### 3.10.3. Manter Campeonato

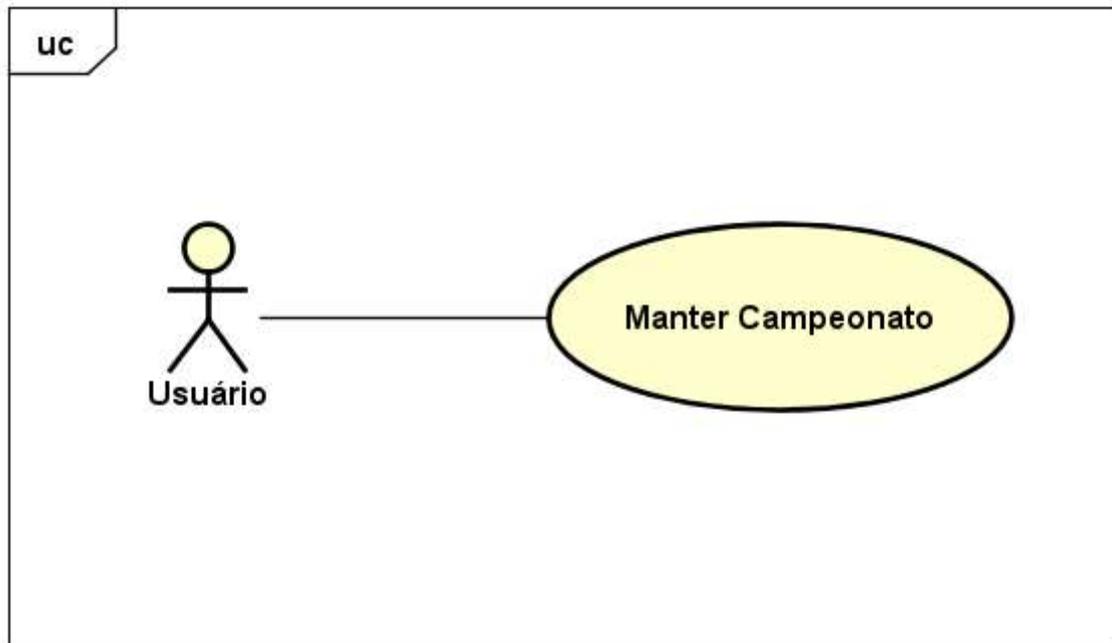
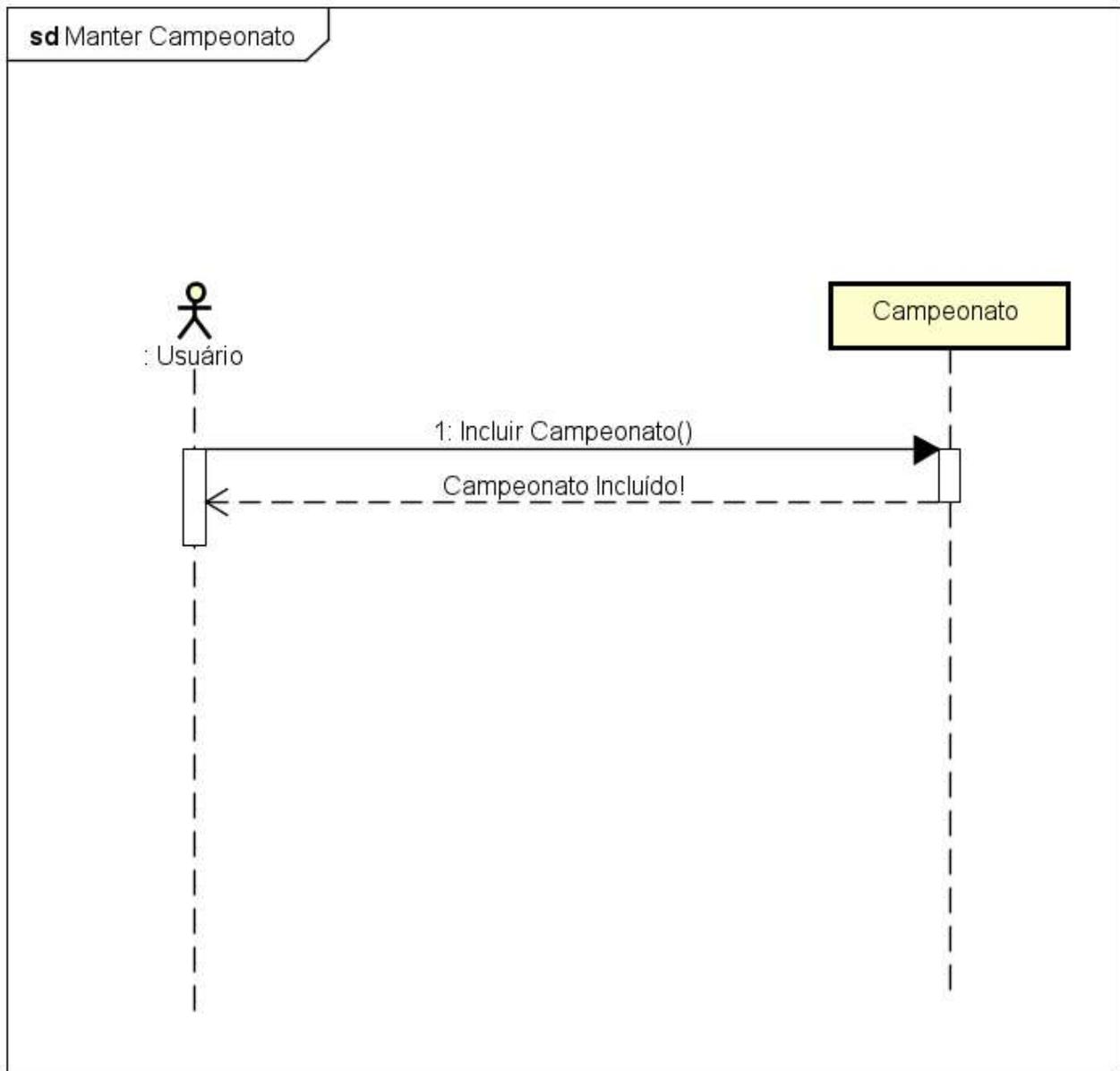


Figura 9 - UC 3: Manter Campeonato

Nome do UC 03	Manter Campeonato
Objetivo	Permitir ao usuário incluir, consultar ou alterar dados de um campeonato.
Ator	Usuário
Pré-Condições	Não existentes.
Fluxo Principal	<p>1 - O usuário seleciona o menu Campeonato e em seguida seleciona Campeonato.</p> <p>2 - Uma nova janela abrirá, com as informações.</p> <p>3 - Caso queira incluir um novo Campeonato, o usuário deverá pressionar o botão "Novo" (A1).</p> <p>4 - Caso queira consultar por um Campeonato, o usuário</p>

	<p>deverá pressionar o botão "Pesquisar" (A2).</p> <p>5 - Caso queira alterar os dados de um Campeonato existente, o usuário deverá selecionar o Campeonato desejado (duplo clique) e em seguida pressionar o botão "Editar". (A3).</p> <p>6 - Caso o usuário queira finalizar essa janela, deverá pressionar o botão "Sair".</p>
Fluxo Alternativo	<p>A1 - O usuário digitará nos campos as informações requisitadas e em seguida pressionará o botão "Gravar".</p> <p>A1.1 - Se tudo estiver correto, receberá uma mensagem que o Campeonato foi incluído com sucesso.</p> <p>A2 - Um campo de pesquisa ficará habilitado, que permite ao usuário escolher o parâmetro que deseja procurar (nome, id, etc).</p> <p>A2.1 - Após digitado, o usuário deverá pressionar o botão "Pesquisar", na aba que foi aberta, e então será feita a filtragem dos dados.</p> <p>A3 - Os campos ficarão habilitados para que seja feita a alteração necessária.</p> <p>A3.1 - Concluída a edição, o usuário deverá pressionar o botão "Gravar" para que as alterações sejam salvas.</p>

**Tabela 4 - Narrativa UC 3: Manter Campeonato**



**Figura 10 - Diagrama de Sequência: Manter Campeonato**

### 3.10.4. Manter Categoria

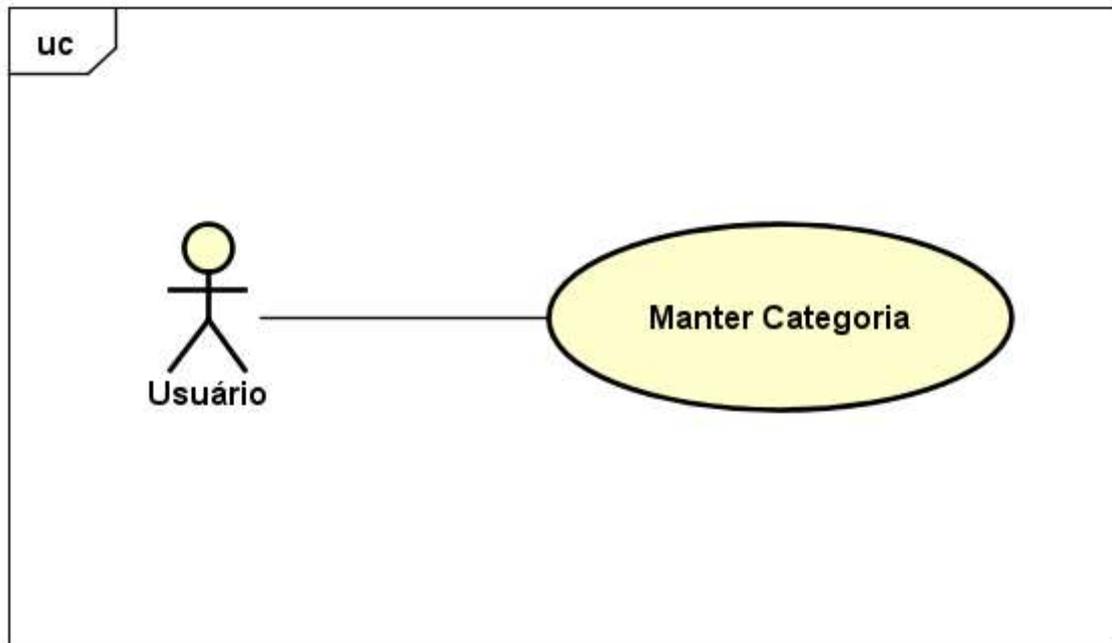
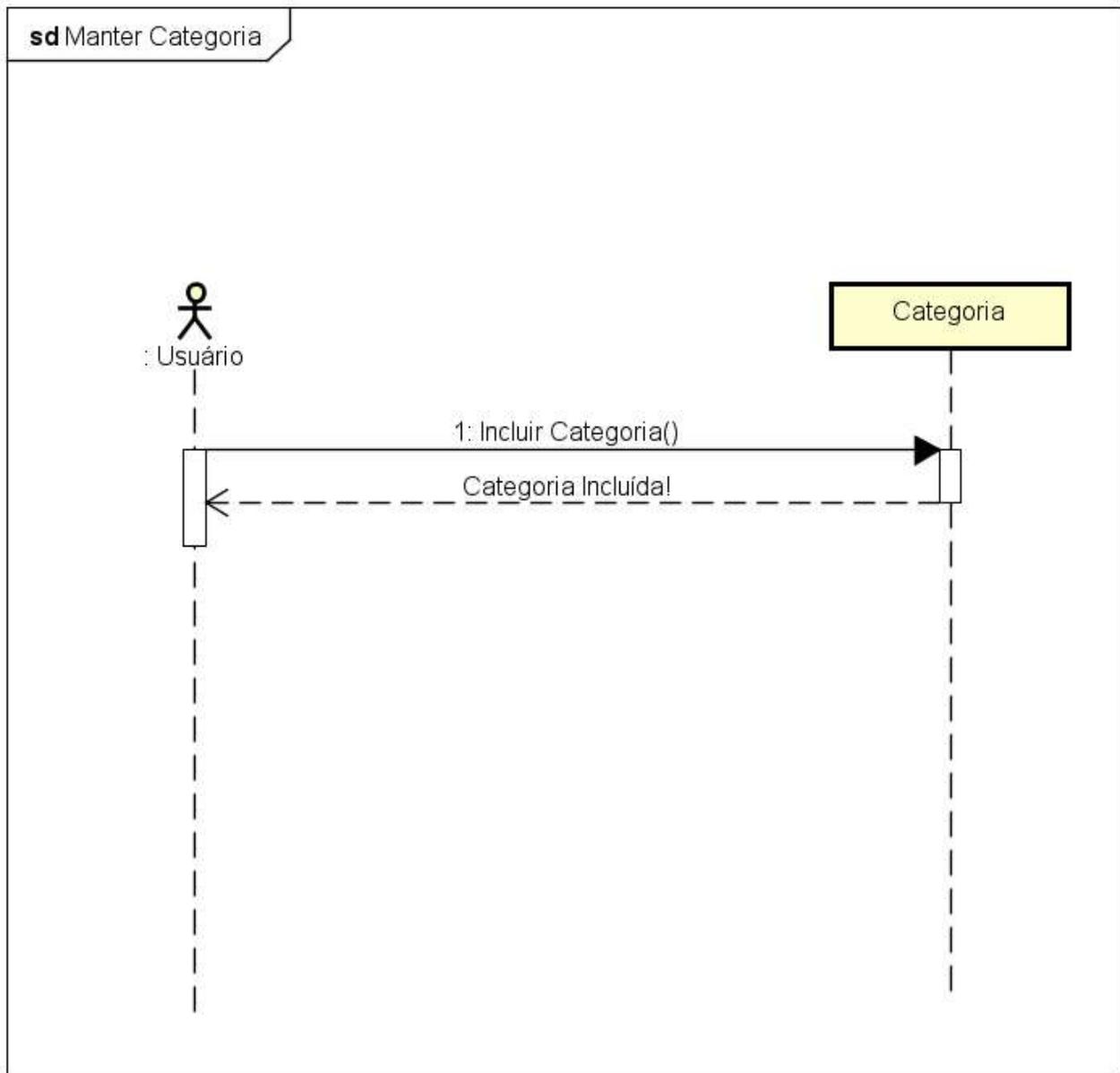


Figura 11 - UC 4: Manter Categoria

Nome do UC 04	Manter Categoria
Objetivo	Permitir ao usuário incluir, consultar ou alterar dados de uma categoria.
Ator	Usuário
Pré-Condições	Não existentes.
Fluxo Principal	<p>1 - O usuário seleciona o Campeonato e em seguida seleciona Categoria.</p> <p>2 - Uma nova janela abrirá, com as informações.</p> <p>3 - Caso queira incluir uma nova Categoria, o usuário deverá pressionar o botão "Novo" (A1).</p> <p>4 - Caso queira consultar por uma Categoria, o usuário deverá</p>

	<p>pressionar o botão "Pesquisar" (A2).</p> <p>5 - Caso queira alterar os dados de uma Categoria existente, o usuário deverá selecionar a Categoria desejada (duplo clique) e em seguida pressionar o botão "Editar". (A3).</p> <p>6 - Caso o usuário queira finalizar essa janela, deverá pressionar o botão "Sair".</p>
Fluxo Alternativo	<p>A1 - O usuário digitará nos campos as informações requisitadas e em seguida pressionará o botão "Gravar".</p> <p>A1.1 - Se tudo estiver correto, receberá uma mensagem que a Categoria foi incluída com sucesso.</p> <p>A2 - Um campo de pesquisa ficará habilitado, que permite ao usuário escolher o parâmetro que deseja procurar (nome, id, etc).</p> <p>A2.1 - Após digitado, o usuário deverá pressionar o botão "Pesquisar", na aba que foi aberta, e então será feita a filtragem dos dados.</p> <p>A3 - Os campos ficarão habilitados para que seja feita a alteração necessária.</p> <p>A3.1 - Concluída a edição, o usuário deverá pressionar o botão "Gravar" para que as alterações sejam salvas.</p>

**Tabela 5 - Narrativa UC 4: Manter Categoria**



**Figura 12 - Diagrama de Sequência: Manter Categoria**

### 3.10.5. Manter Cidade

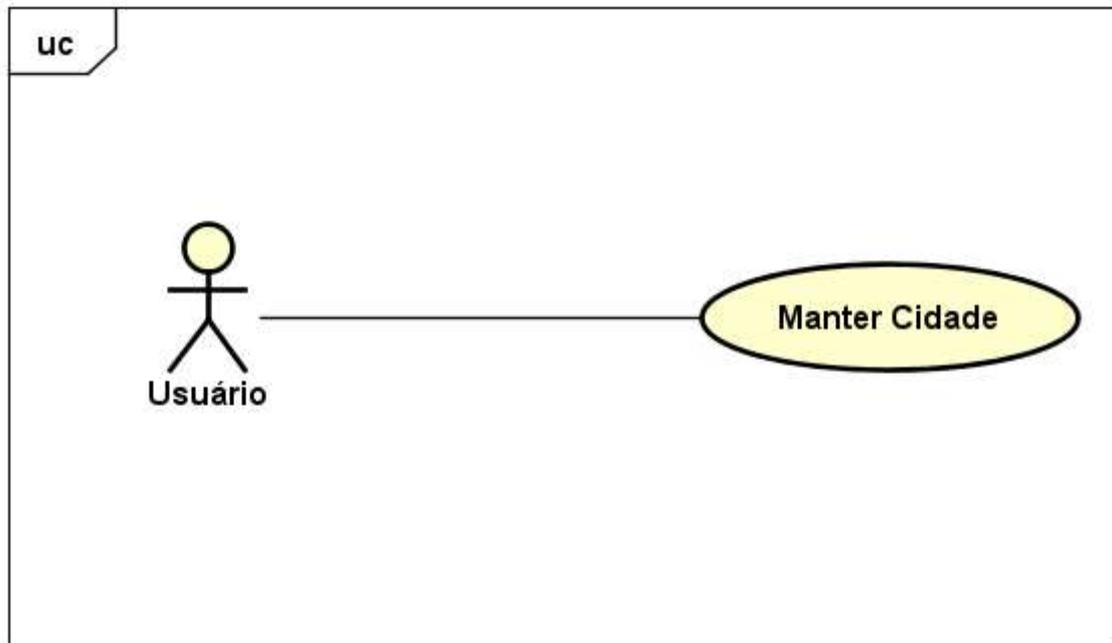
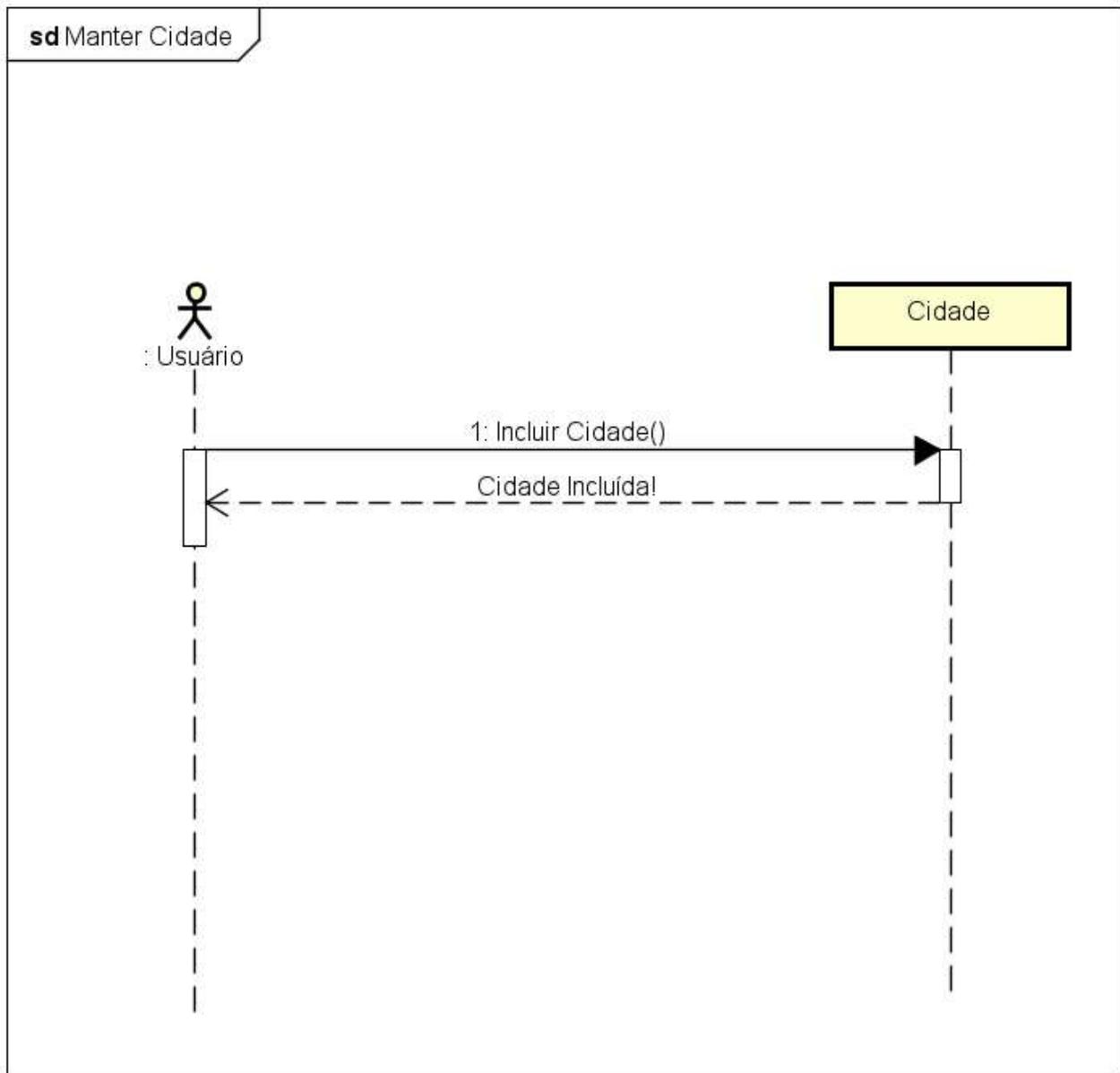


Figura 13 - UC 5: Manter Cidade

Nome do UC 05	Manter Cidade
Objetivo	Permitir ao usuário incluir, consultar ou alterar dados de uma cidade.
Ator	Usuário
Pré-Condições	Não existentes.
Fluxo Principal	<p>1 - O usuário seleciona no menu Cadastros e em seguida seleciona Cidade.</p> <p>2 - Uma nova janela abrirá, com as informações.</p> <p>3 - Caso queira incluir uma nova Cidade, o usuário deverá pressionar o botão "Novo" (A1).</p> <p>4 - Caso queira consultar por uma Cidade, o usuário deverá</p>

	<p>pressionar o botão "Pesquisar" (A2).</p> <p>5 - Caso queira alterar os dados de uma Cidade existente, o usuário deverá selecionar a Cidade desejada (duplo clique) e em seguida pressionar o botão "Editar". (A3).</p> <p>6 - Caso o usuário queira finalizar essa janela, deverá pressionar o botão "Sair".</p>
Fluxo Alternativo	<p>A1 - O usuário digitará nos campos as informações requisitadas e em seguida pressionará o botão "Gravar".</p> <p>A1.1 - Se tudo estiver correto, receberá uma mensagem que a Cidade foi incluída com sucesso.</p> <p>A2 - Um campo de pesquisa ficará habilitado, que permite ao usuário escolher o parâmetro que deseja procurar (nome, id, etc).</p> <p>A2.1 - Após digitado, o usuário deverá pressionar o botão "Pesquisar", na aba que foi aberta, e então será feita a filtragem dos dados.</p> <p>A3 - Os campos ficarão habilitados para que seja feita a alteração necessária.</p> <p>A3.1 - Concluída a edição, o usuário deverá pressionar o botão "Gravar" para que as alterações sejam salvas.</p>

**Tabela 6 - Narrativa UC 5: Manter Cidade**



**Figura 14 - Diagrama de Sequência: Manter Cidade**

### 3.10.6. Manter Competidor

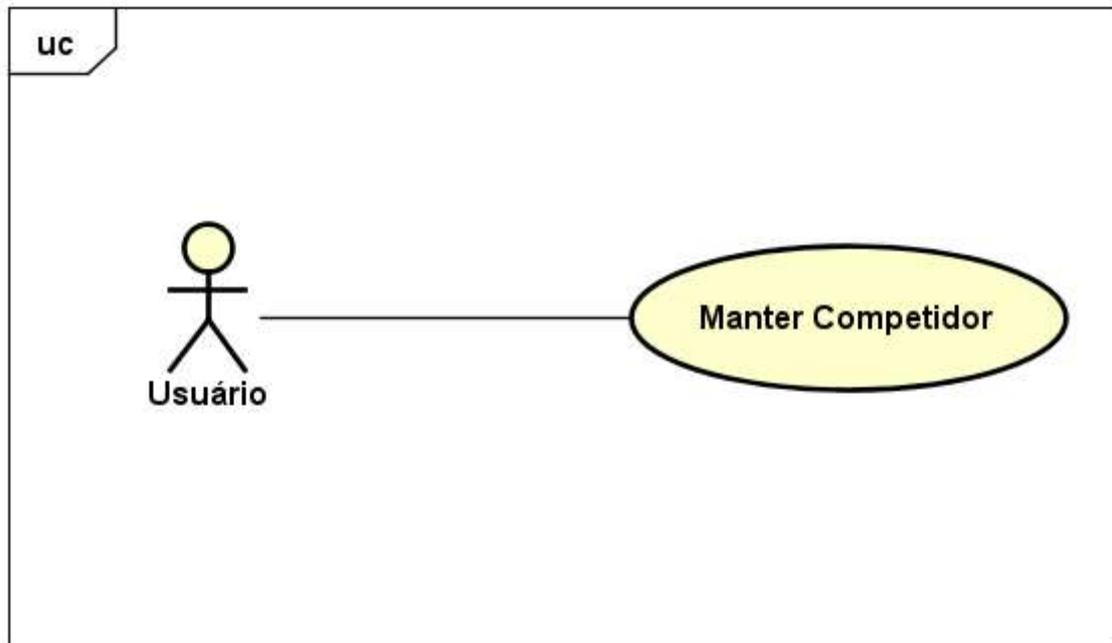
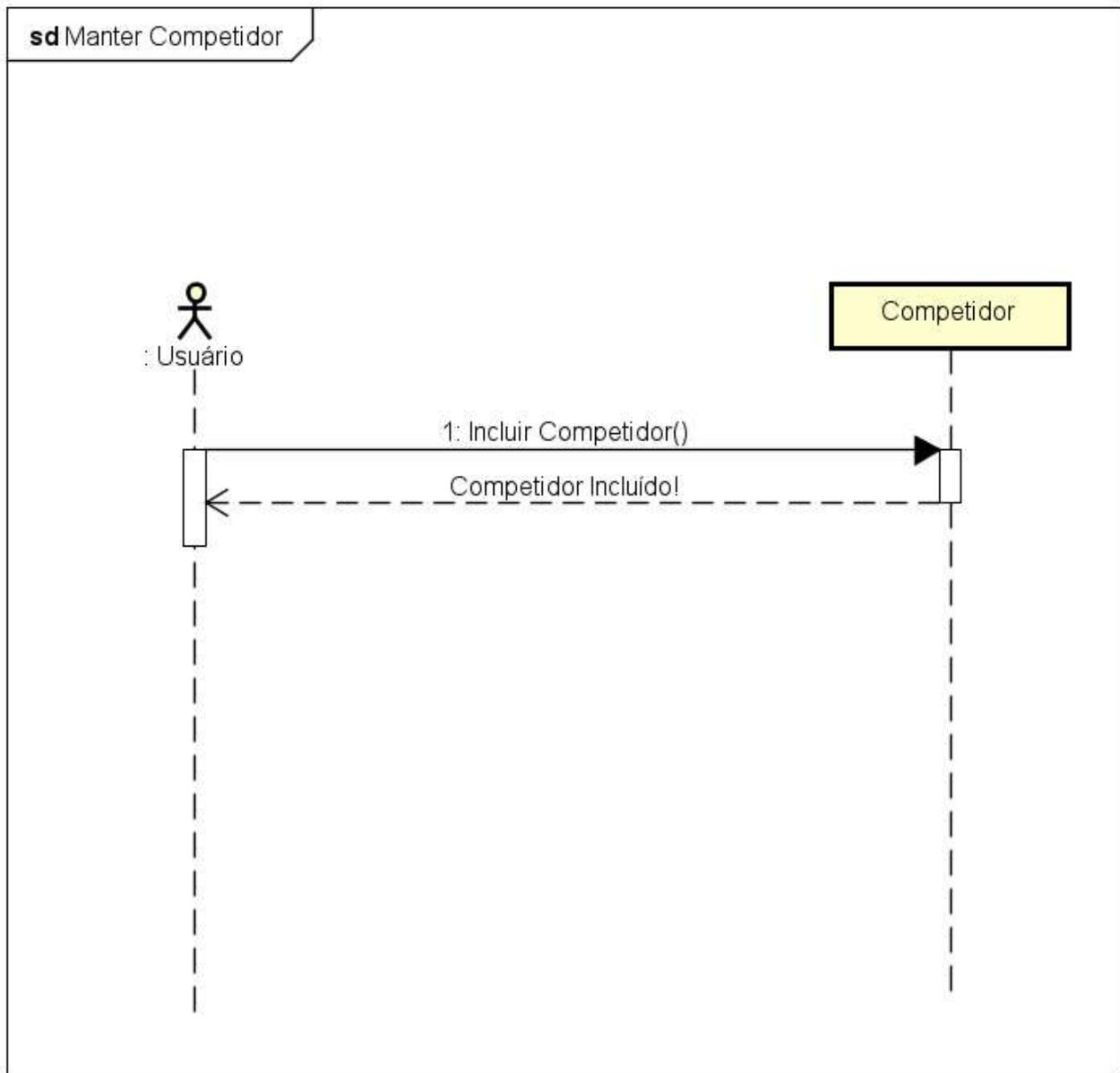


Figura 15 - UC 6: Manter Competidor

Nome do UC 06	Manter Competidor
Objetivo	Permitir ao usuário incluir, consultar ou alterar dados de um competidor.
Ator	Usuário
Pré-Condições	Não existentes.
Fluxo Principal	<p>1 - O usuário seleciona no menu Cadastros e em seguida seleciona Competidor.</p> <p>2 - Uma nova janela abrirá, com as informações.</p> <p>3 - Caso queira incluir um novo Competidor, o usuário deverá pressionar o botão "Novo" (A1).</p> <p>4 - Caso queira consultar por um Competidor, o usuário deverá</p>

	<p>pressionar o botão "Pesquisar" (A2).</p> <p>5 - Caso queira alterar os dados de um Competidor existente, o usuário deverá selecionar o Competidor desejado (duplo clique) e em seguida pressionar o botão "Editar". (A3).</p> <p>6 - Caso o usuário queira finalizar essa janela, deverá pressionar o botão "Sair".</p>
Fluxo Alternativo	<p>A1 - O usuário digitará nos campos as informações requisitadas e em seguida pressionará o botão "Gravar".</p> <p>A1.1 - Se tudo estiver correto, receberá uma mensagem que o Competidor foi incluído com sucesso.</p> <p>A2 - Um campo de pesquisa ficará habilitado, que permite ao usuário escolher o parâmetro que deseja procurar (nome, id, etc).</p> <p>A2.1 - Após digitado, o usuário deverá pressionar o botão "Pesquisar", na aba que foi aberta, e então será feita a filtragem dos dados.</p> <p>A3 - Os campos ficarão habilitados para que seja feita a alteração necessária.</p> <p>A3.1 - Concluída a edição, o usuário deverá pressionar o botão "Gravar" para que as alterações sejam salvas.</p>

**Tabela 7 - Narrativa UC 6: Manter Competidor**



**Figura 16 - Diagrama de Sequência: Manter Competidor**

### 3.10.7. Manter Etapa

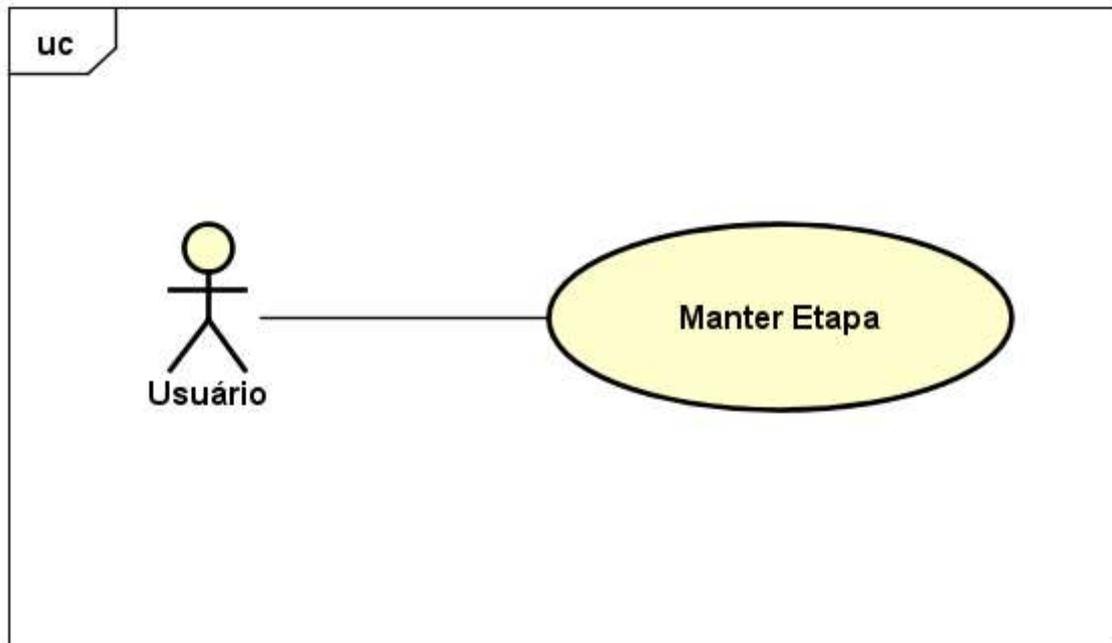


Figura 17 - UC 7: Manter Etapa

Nome do UC 07	Manter Etapa
Objetivo	Permitir ao usuário incluir, consultar ou alterar dados de uma etapa.
Ator	Usuário
Pré-Condições	Não existentes.
Fluxo Principal	<p>1 - O usuário seleciona o menu Campeonato e em seguida seleciona Etapa.</p> <p>2 - Uma nova janela abrirá, com as informações.</p> <p>3 - Caso queira incluir uma nova Etapa, o usuário deverá pressionar o botão "Novo" (A1).</p> <p>4 - Caso queira consultar por uma Etapa, o usuário deverá</p>

	<p>pressionar o botão "Pesquisar" (A2).</p> <p>5 - Caso queira alterar os dados de uma Etapa existente, o usuário deverá selecionar a Etapa desejada (duplo clique) e em seguida pressionar o botão "Editar". (A3).</p> <p>6 - Caso o usuário queira finalizar essa janela, deverá pressionar o botão "Sair".</p>
Fluxo Alternativo	<p>A1 - O usuário digitará nos campos as informações requisitadas e em seguida pressionará o botão "Gravar".</p> <p>A1.1 - Se tudo estiver correto, receberá uma mensagem que a Etapa foi incluída com sucesso.</p> <p>A2 - Um campo de pesquisa ficará habilitado, que permite ao usuário escolher o parâmetro que deseja procurar (nome, id, etc).</p> <p>A2.1 - Após digitado, o usuário deverá pressionar o botão "Pesquisar", na aba que foi aberta, e então será feita a filtragem dos dados.</p> <p>A3 - Os campos ficarão habilitados para que seja feita a alteração necessária.</p> <p>A3.1 - Concluída a edição, o usuário deverá pressionar o botão "Gravar" para que as alterações sejam salvas.</p>

**Tabela 8 - Narrativa UC 7: Manter Etapa**

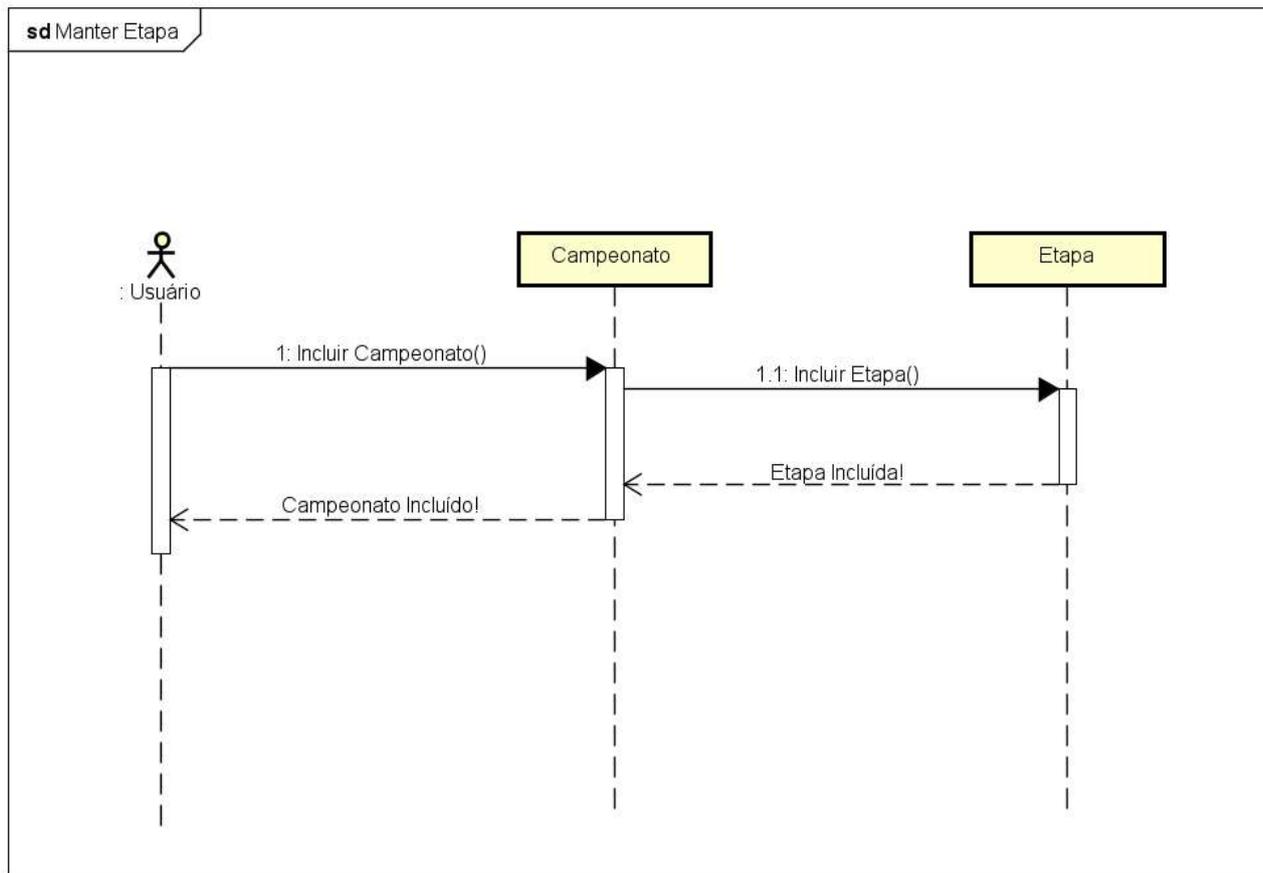


Figura 18 - Diagrama de Sequência: Manter Etapa

### 3.10.8. Manter Tipo Penalidade

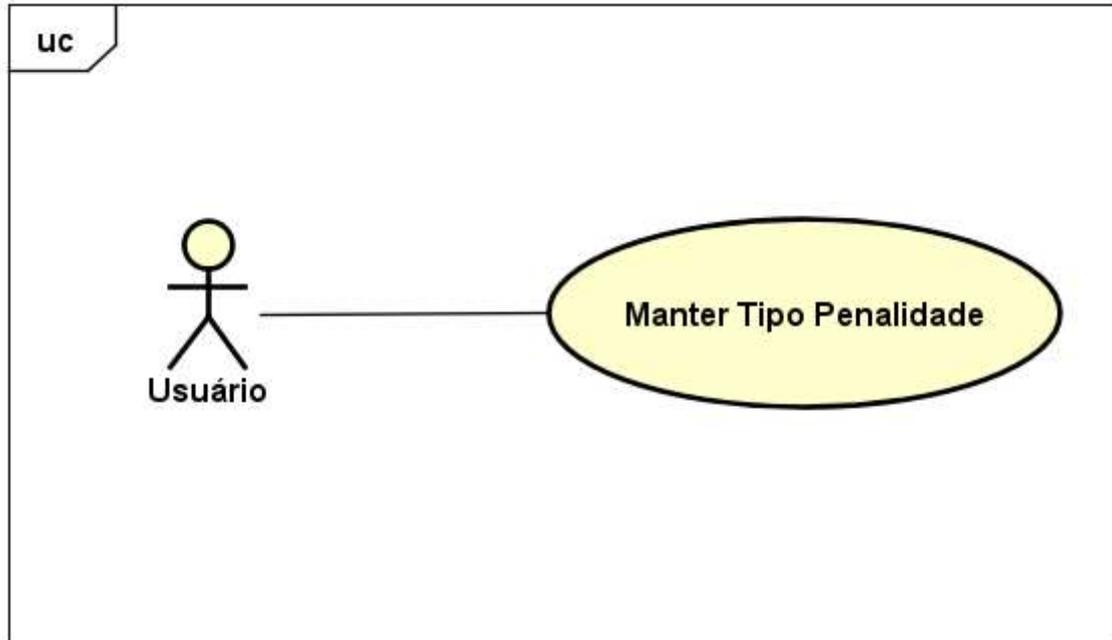
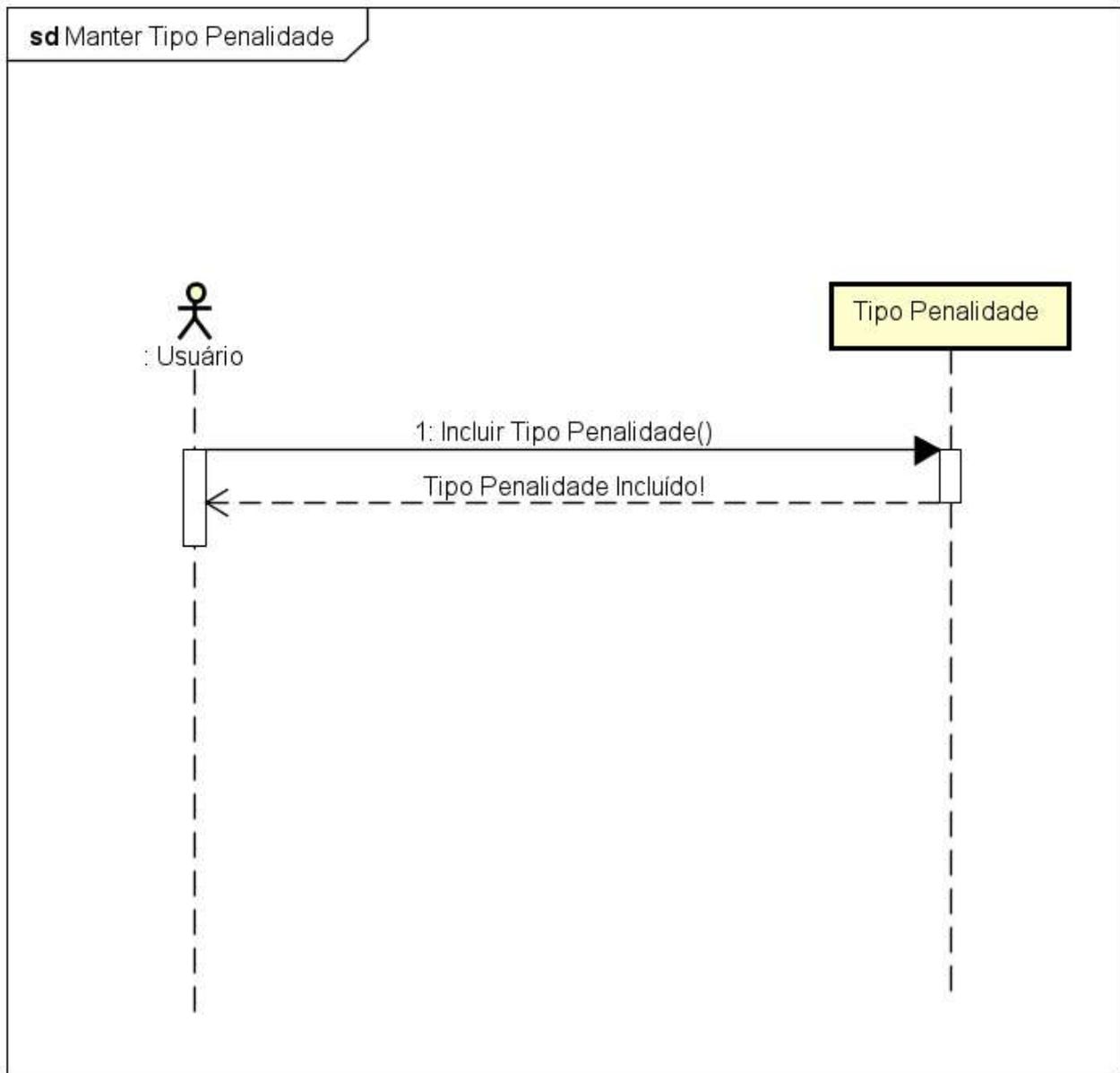


Figura 19 - UC 8: Manter Tipo Penalidade

Nome do UC 08	Manter Tipo Penalidade
Objetivo	Permitir ao usuário incluir, consultar ou alterar dados de um tipo de penalidade.
Ator	Usuário
Pré-Condições	Não existentes.
Fluxo Principal	<p>1 - O usuário seleciona no menu Cadastros e em seguida seleciona Tipo Penalidade.</p> <p>2 - Uma nova janela abrirá, com as informações.</p> <p>3 - Caso queira incluir um novo Tipo Penalidade, o usuário deverá pressionar o botão "Novo" (A1).</p> <p>4 - Caso queira consultar por um Tipo Penalidade, o usuário</p>

	<p>deverá pressionar o botão "Pesquisar" (A2).</p> <p>5 - Caso queira alterar os dados de um Tipo Penalidade existente, o usuário deverá selecionar o Tipo Penalidade desejado (duplo clique) e em seguida pressionar o botão "Editar". (A3).</p> <p>6 - Caso o usuário queira finalizar essa janela, deverá pressionar o botão "Sair".</p>
Fluxo Alternativo	<p>A1 - O usuário digitará nos campos as informações requisitadas e em seguida pressionará o botão "Gravar".</p> <p>A1.1 - Se tudo estiver correto, receberá uma mensagem que o Tipo Penalidade foi incluído com sucesso.</p> <p>A2 - Um campo de pesquisa ficará habilitado, que permite ao usuário escolher o parâmetro que deseja procurar (nome, id, etc).</p> <p>A2.1 - Após digitado, o usuário deverá pressionar o botão "Pesquisar", na aba que foi aberta, e então será feita a filtragem dos dados.</p> <p>A3 - Os campos ficarão habilitados para que seja feita a alteração necessária.</p> <p>A3.1 - Concluída a edição, o usuário deverá pressionar o botão "Gravar" para que as alterações sejam salvas.</p>

**Tabela 9 - Narrativa UC 8: Manter Tipo Penalidade**



**Figura 20 - Diagrama de Sequência: Manter Tipo Penalidade**

### 3.10.9. Gerar Relatórios

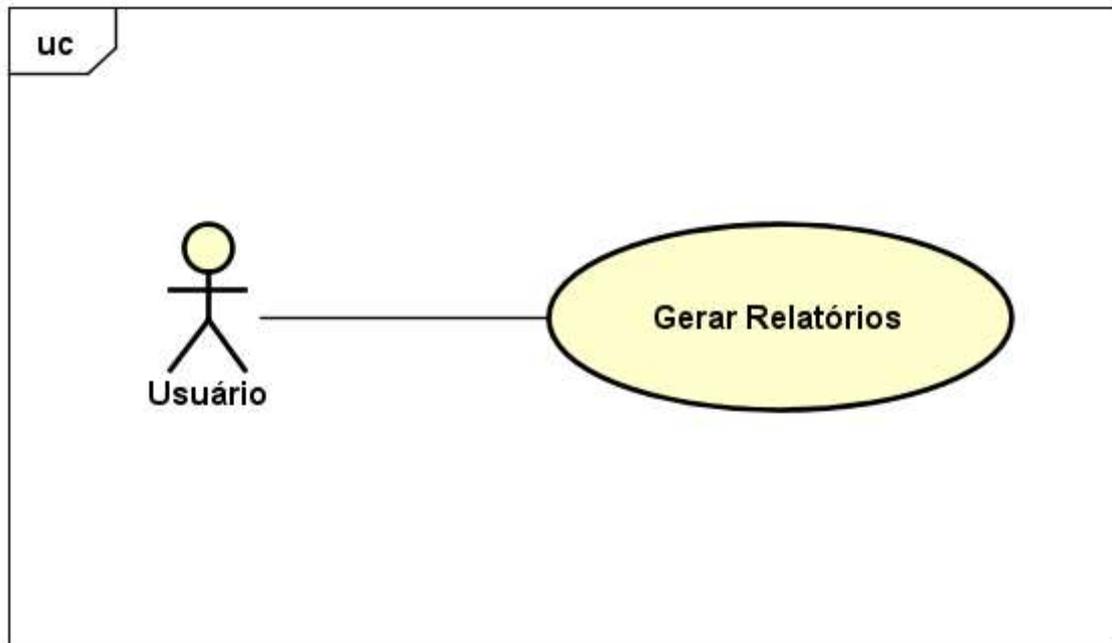
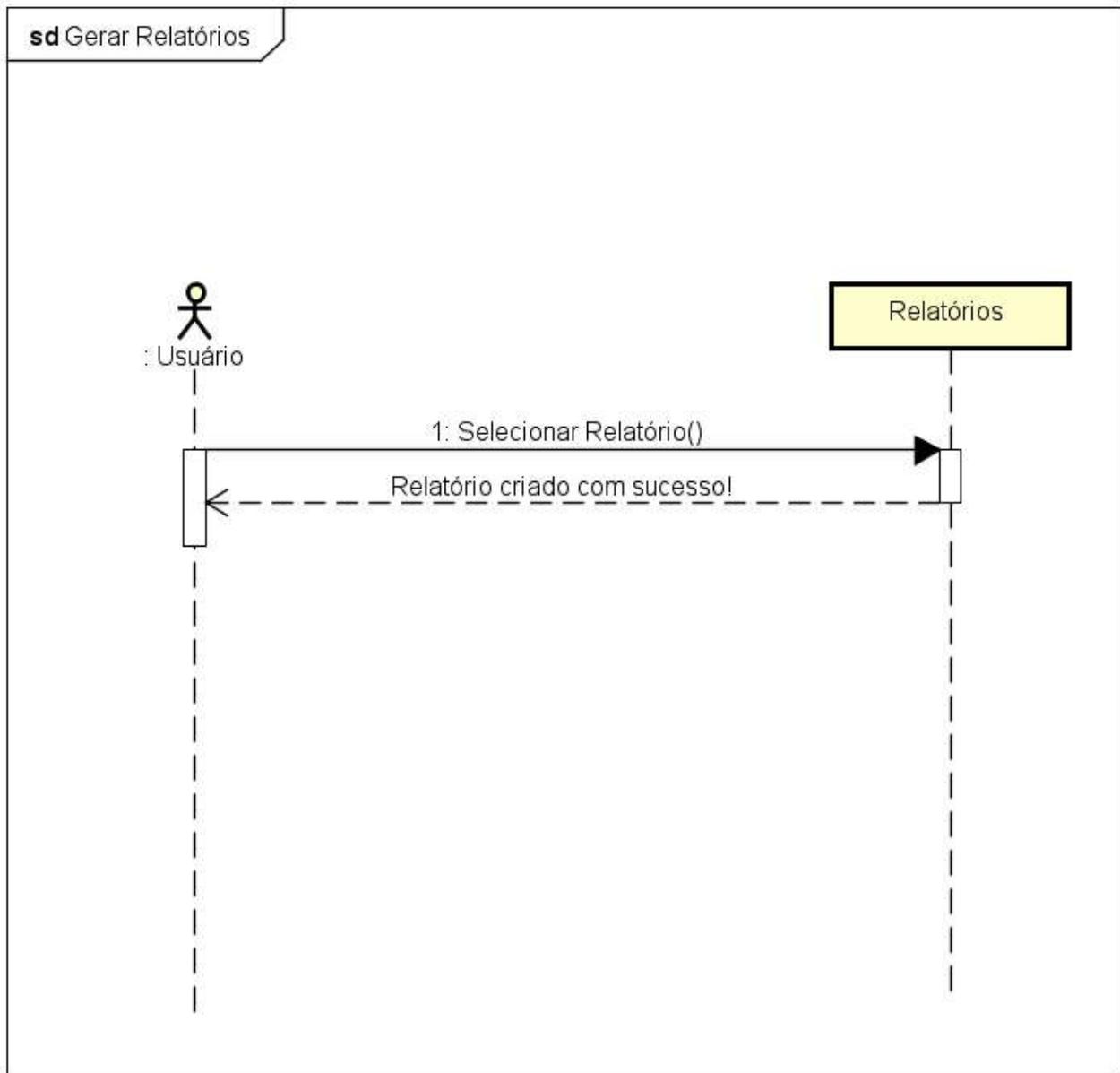


Figura 21 - UC 9: Gerar Relatórios

Nome do UC 09	Gerar Relatórios
Objetivo	Gerar relatórios para facilitar a visualização das informações geradas.
Ator	Usuário e Sistema
Pré-Condições	Ter informações no BD.
Fluxo Principal	<p>1 - Usuário pressiona no menu "Relatórios" e seleciona a opção desejada.</p> <p>2 - Caso pressione em "Etapa", uma nova tela será aberta (A1).</p> <p>3 - Caso pressione em "Gerais", uma nova tela será aberta (A2).</p>

Fluxo Alternativo	<p>A1 - Conforme o usuário seleciona a informação desejada, o sistema libera o próximo campo para opção.</p> <p>A1.1 - Após selecionar todas as informações necessárias, o usuário deverá pressionar o botão "Gerar", e após alguns segundos, o relatório estará pronto e abrirá automaticamente.</p> <p>A1.2 - Caso mude de idéia, o usuário poderá pressionar o botão "Voltar" a qualquer momento para fechar a tela.</p> <p>A2 - Na tela, será necessário selecionar o Campeonato ao qual se deseja emitir os relatórios.</p> <p>A2.1 - Usuário seleciona o tipo de relatório que deseja, e deverá pressionar o botão "Gerar", e após alguns segundos, o relatório estará pronto e abrirá automaticamente.</p> <p>A2.2 - Caso mude de idéia, o usuário poderá pressionar o botão "Voltar" a qualquer momento para fechar a tela.</p>
-------------------	---

**Tabela 10 - Narrativa UC 9: Gerar Relatórios**



**Figura 22 - Diagrama de Sequência: Gerar Relatórios**

### 3.10.10. Gerar Relatórios Gerais

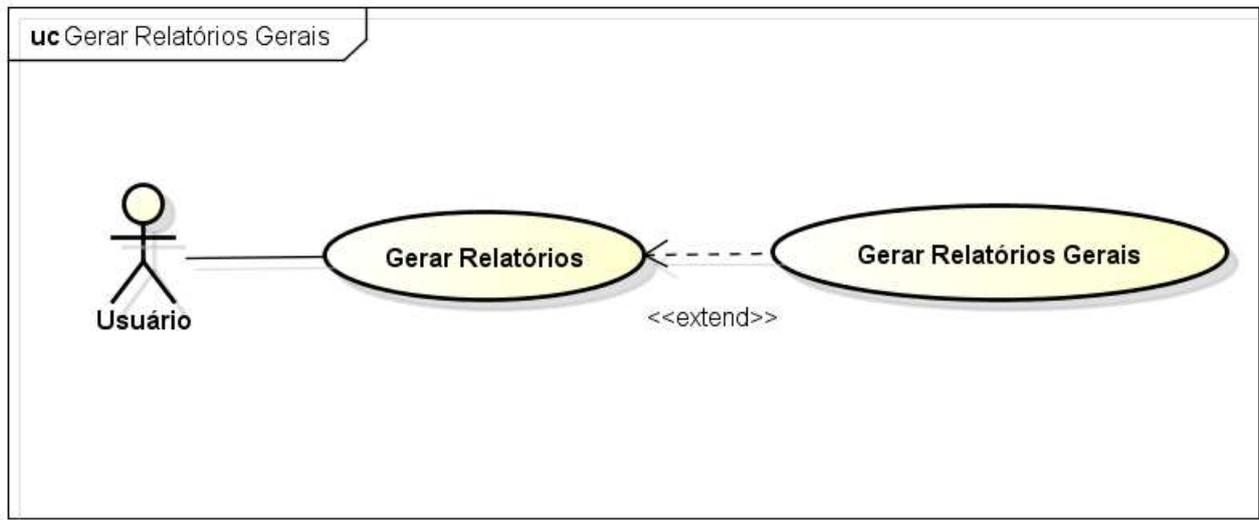
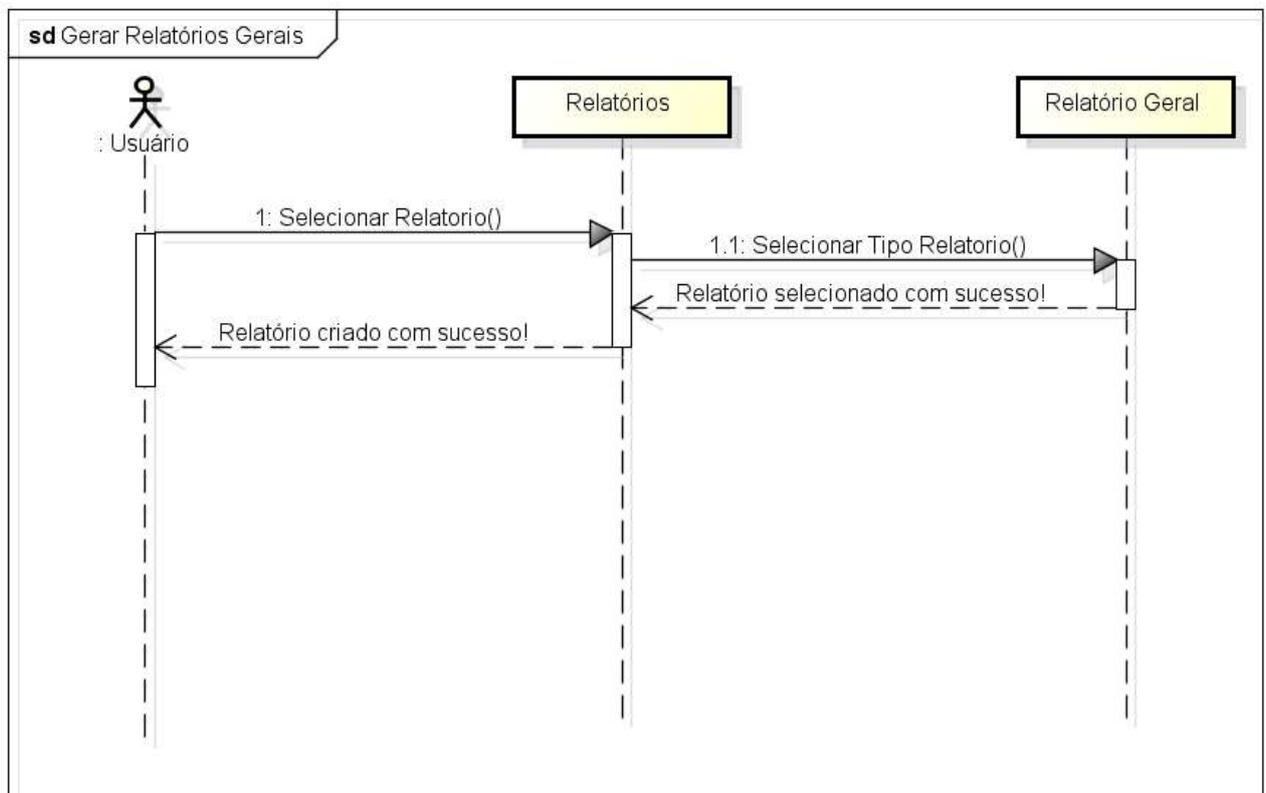


Figura 23 – UC 10 - Gerar Relatórios Gerais

Nome do UC 10	Gerar Relatórios Gerais
Objetivo	Gerar relatórios gerais para facilitar a visualização das informações cadastradas no BD.
Ator	Usuário e Sistema
Pré-Condições	Ter informações no BD.
Fluxo Principal	<p>1 - Usuário pressiona no menu "Relatórios" e seleciona a opção "Gerais".</p> <p>2 - Uma nova janela será aberta, aonde o usuário deverá escolher o Campeonato que deseja, e após, selecionar o tipo de relatório que deseja gerar.</p> <p>3 - Caso Selecione "Relatório Geral de Competidores" (A1).</p> <p>4 - Caso Selecione "Relatório Geral de Animais" (A1).</p> <p>5 – Caso Selecione "Relatório Geral de Cidades" (A1).</p> <p>6 – Caso Selecione "Relatório Geral de Passadas" (A1).</p>

Fluxo Alternativo	<p>A1 - Após selecionar o tipo de relatório desejado, o usuário deverá pressionar o botão "Gerar", e após alguns segundos, o relatório estará pronto e abrirá automaticamente.</p> <p>A1.1 - Caso mude de idéia, o usuário poderá pressionar o botão "Voltar" a qualquer momento para fechar a tela.</p>
-------------------	--

**Tabela 11 - Narrativa UC 10: Gerar Relatórios Gerais**



**Figura 24 - Diagrama de Sequência: Gerar Relatórios Gerais**

### 3.10.11. Gerar Relatório Etapa

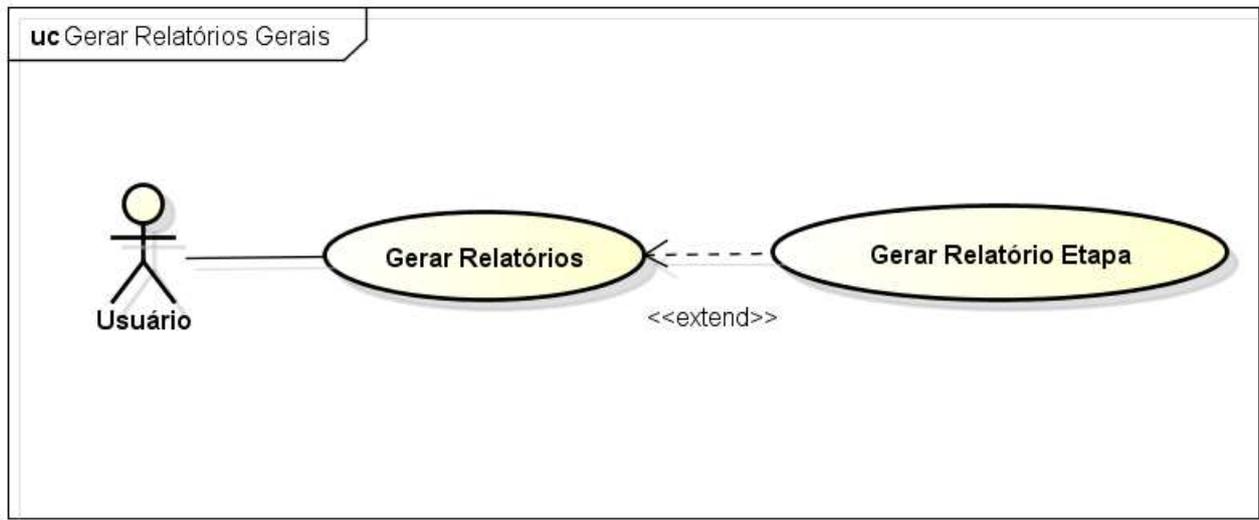
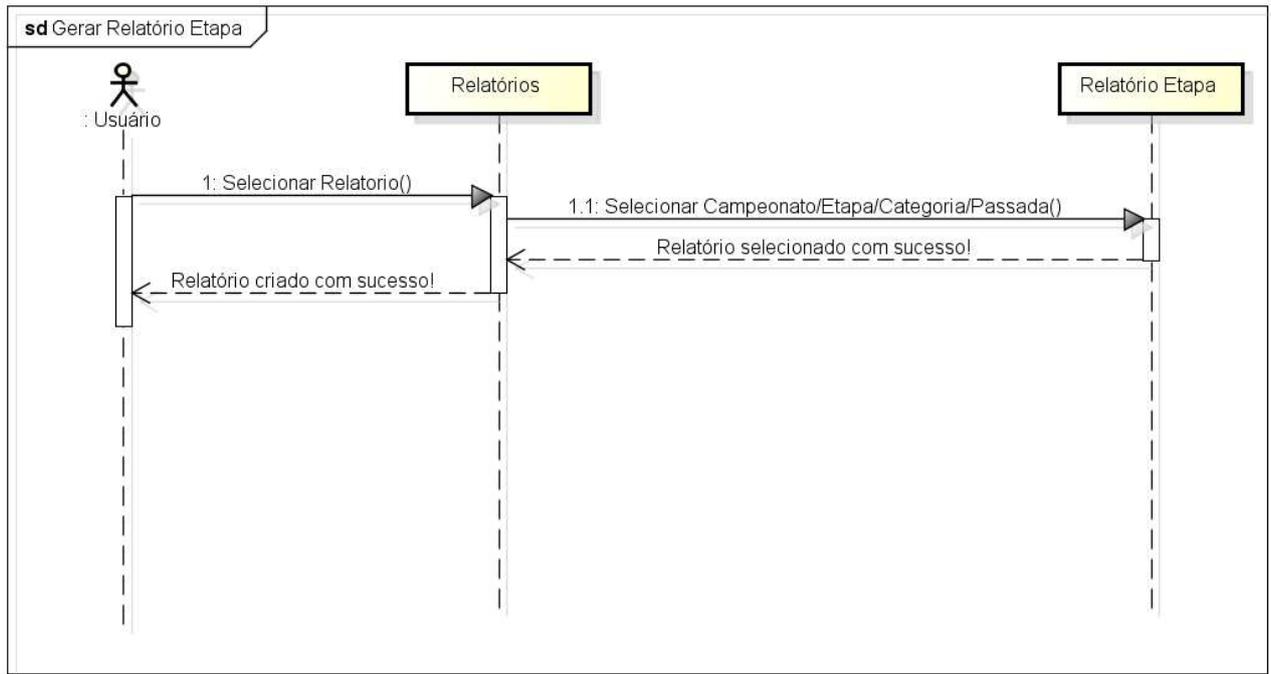


Figura 25 - UC 11 - Gerar Relatório Etapa

Nome do UC 11	Gerar Relatório Etapa
Objetivo	Gerar relatório etapa para facilitar a visualização do resultado de uma determinada etapa.
Ator	Usuário e Sistema
Pré-Condições	Ter informações no BD.
Fluxo Principal	<p>1 - Usuário pressiona no menu "Relatórios" e seleciona a opção "Etapa".</p> <p>2 - Uma nova janela será aberta, aonde o usuário deverá escolher: Campeonato, Etapa, Categoria, Dia, Passada e Tipo.</p> <p>3 – A ordem de seleção deverá ser da esquerda para direita e de cima para baixo (A1).</p> <p>4 – Para habilitar a próxima caixa, o usuário deverá pressionar a tecla TAB, para sair da caixa atual.</p> <p>5 – Após selecionadas todas as opções, o usuário deverá pressionar o botão "Gerar". Após concluído, o relatório abrirá</p>

	automaticamente.
Fluxo Alternativo	A1 - Caso mude de idéia, o usuário poderá pressionar o botão "Voltar" a qualquer momento para fechar a tela.

**Tabela 12 - Narrativa UC 11: Gerar Relatório Etapa**



**Figura 26 - Diagrama de Sequência: Gerar Relatório Etapa**

### 3.10.12. Gerar Backup

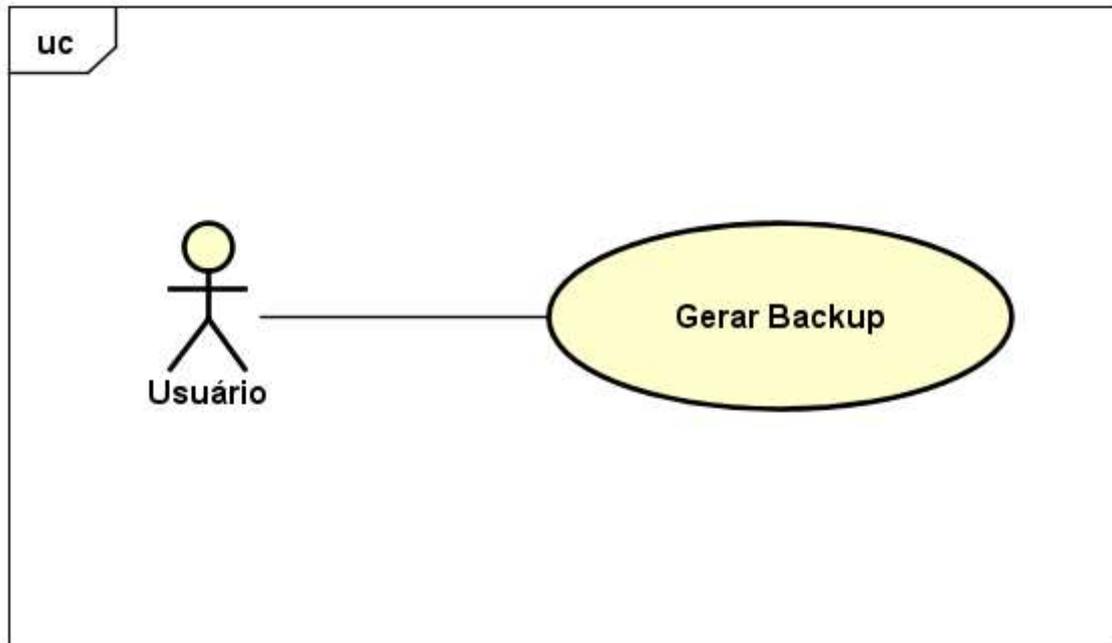
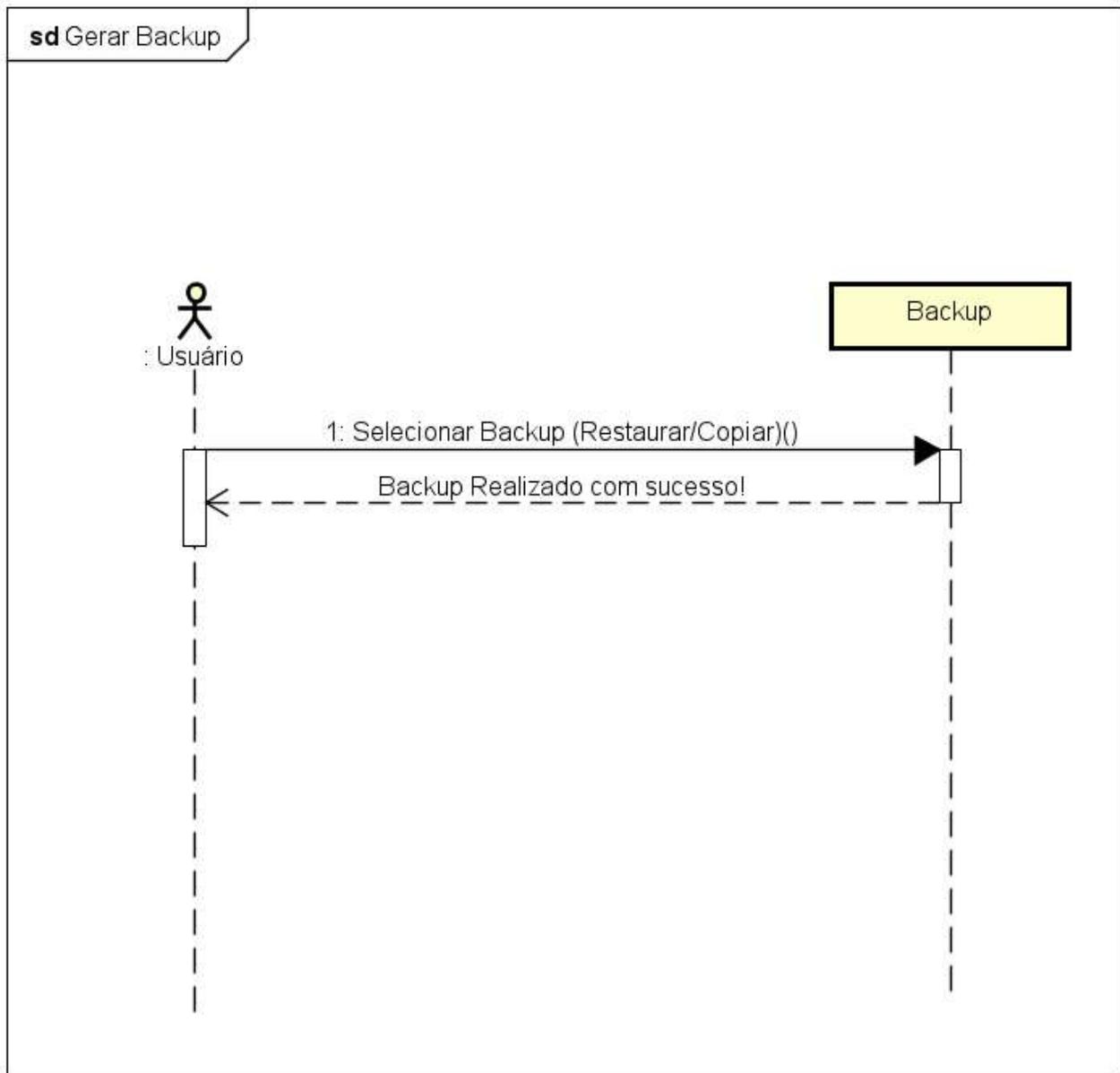


Figura 27 - UC 12: Gerar Backup

Nome do UC 12	Gerar Backup
Objetivo	Gerar backups dos dados.
Ator	Usuário
Pré-Condições	Ter informações no BD.
Fluxo Principal	<p>1 - Usuário pressiona no menu "Backup" e seleciona a opção desejada.</p> <p>2 - Caso pressione em "Copiar", uma nova tela será aberta (A1).</p> <p>3 - Caso pressione em "Restaurar", uma nova tela será aberta (A2).</p>
Fluxo Alternativo	A1 - O usuário deverá selecionar as informações necessárias

	<p>para criar o arquivo de Backup (Campeonato, Etapa, Diretório em que será salvo o arquivo gerado e o Nome que terá o arquivo).</p> <p>A1.1 - Após selecionadas todas as opções, o usuário deverá pressionar o botão "Gravar", para ser gerado o arquivo de backup.</p> <p>A1.2 - Caso mude de idéia, o usuário poderá pressionar o botão "Voltar" a qualquer momento para fechar a tela.</p> <p>A2 - O usuário deverá selecionar o diretório de origem do arquivo que deseja utilizar para restaurar os dados.</p> <p>A2.1 - Após selecionado o arquivo, o usuário deverá pressionar o botão "Restaurar" para dar início a ação.</p> <p>A2.2 - O sistema avisará ao usuário que todas as informações serão substituídas pelas do arquivo que foi selecionado para backup.</p> <p>A2.3 - Caso mude de idéia, o usuário poderá pressionar o botão "Voltar" a qualquer momento para fechar a tela.</p>
--	---

**Tabela 13 - Narrativa UC 12: Gerar Backup**



**Figura 28 - Diagrama de Sequência: Gerar Backup**

### 3.11. DIAGRAMA DE CLASSE

Trata-se de uma representação estrutural das classes e relações entre elas (Figura 29).

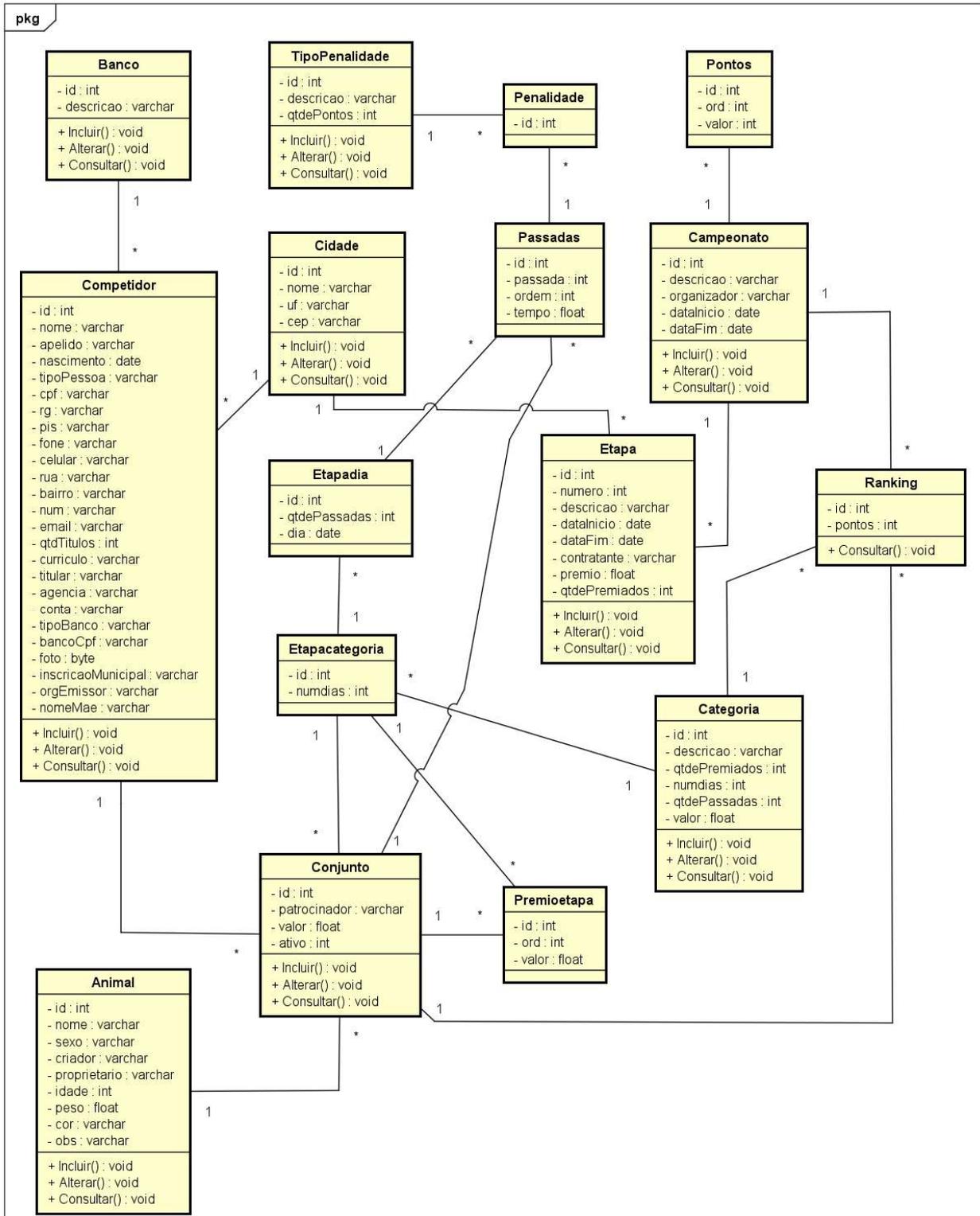


Figura 29 - Diagrama de Classes

### 3.12. DIAGRAMA DE ATIVIDADES

Demonstra os fluxos da funcionalidade, descrevendo a sequência da atividade por um cenário estabelecido (Figura 30).

#### 3.12.1. Gerar Passada

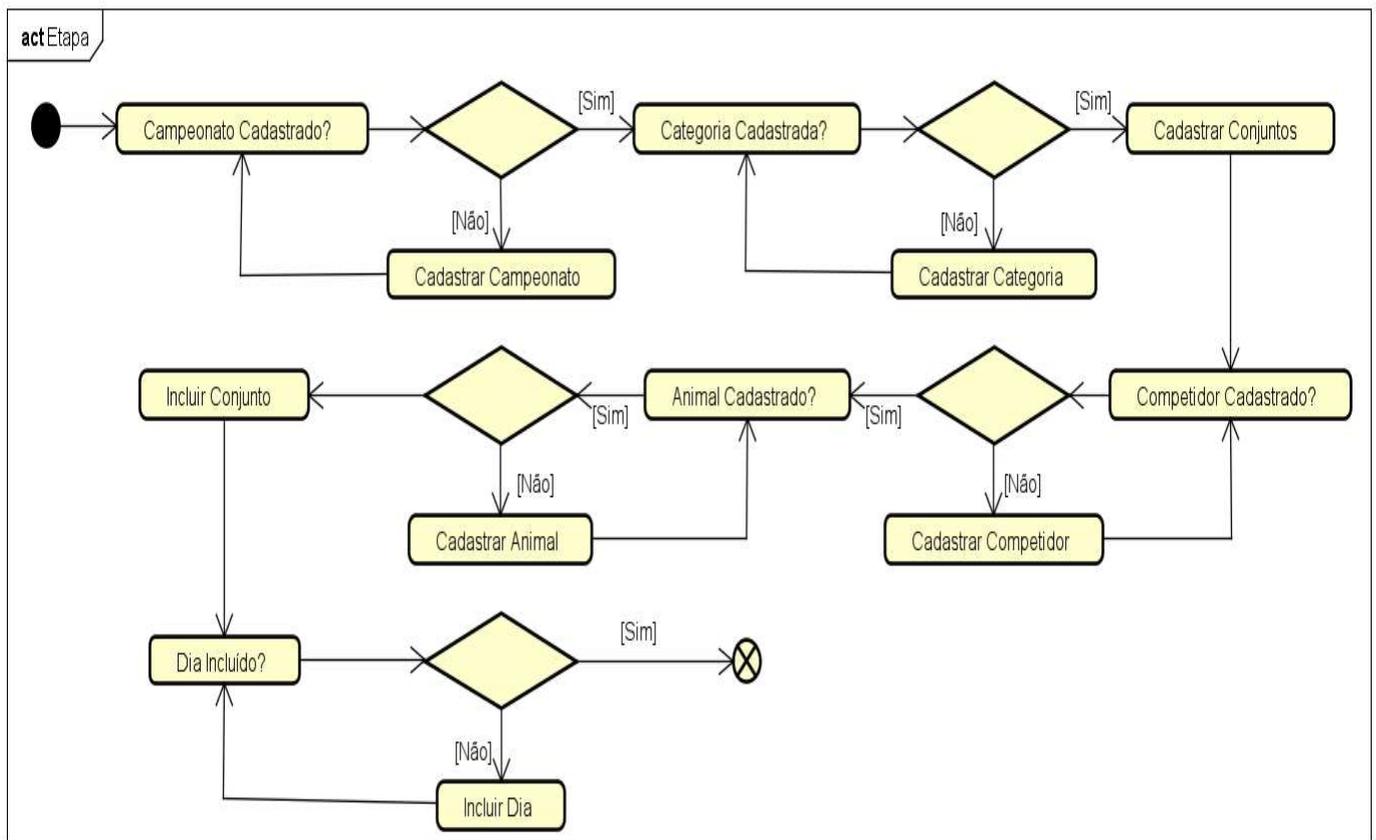


Figura 30 - Diagrama de Atividades: Gerar Passada

### 3.13. DIAGRAMA ENTIDADE-RELACIONAMENTO (DER)

O diagrama entidade-relacionamento (DER) tem como objetivo descrever, de forma totalmente abstrata, a arquitetura de Banco de Dados do Software proposto, que será desenvolvido (Figura 31).

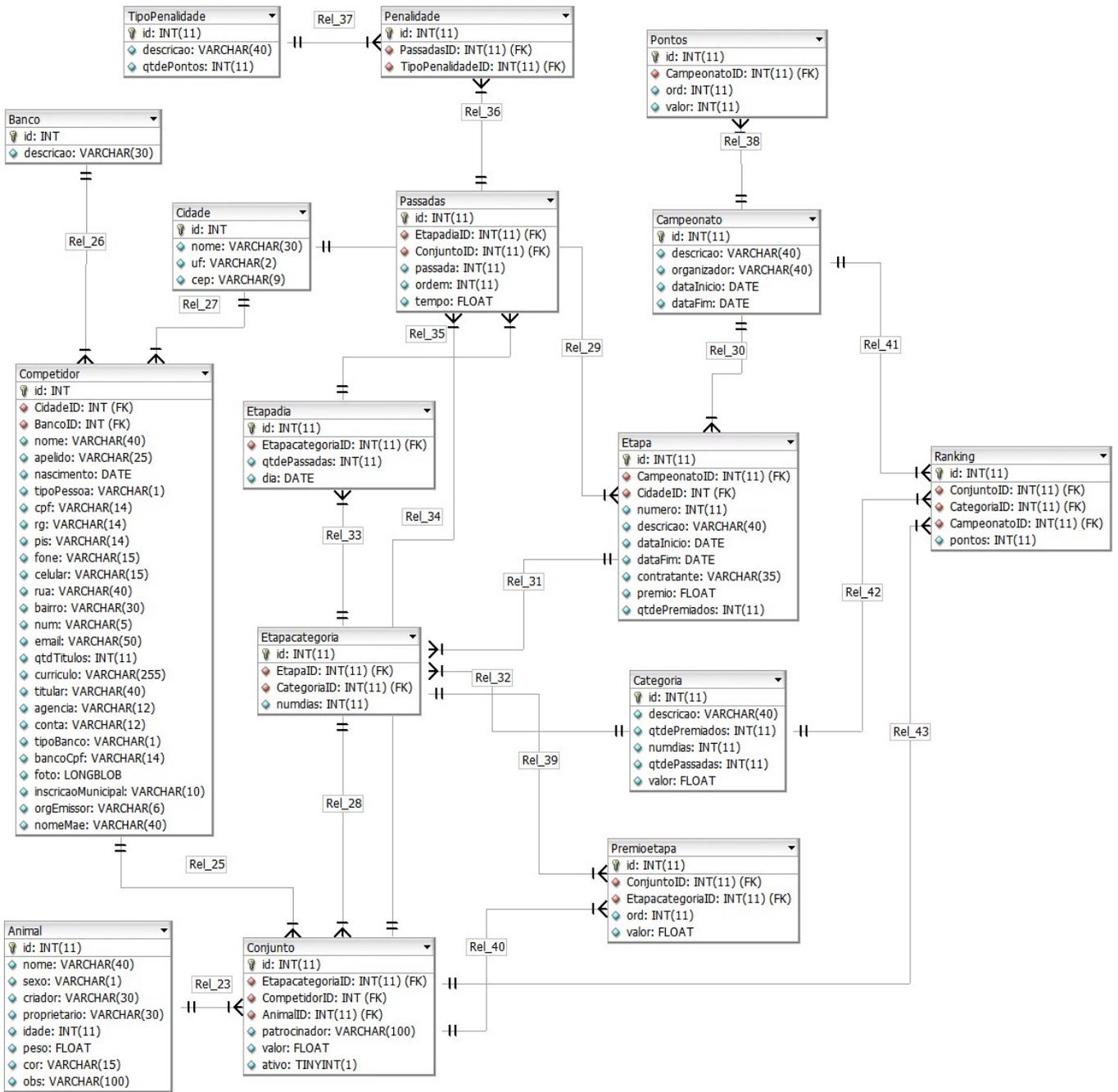


Figura 31 - Diagrama Entidade Relacionamento

## 4. IMPLEMENTAÇÃO

Para a implementação do sistema proposto foi utilizada a IDE da Microsoft, o Visual Studio 2019, a linguagem C# para codificação, o EntityFramework como camada de persistência e uma base de dados MySQL. Foi utilizado também a ferramenta XAMPP<sup>11</sup>, que proporciona a criação de um DB local para mantêr os dados relevantes das provas.

### 4.1. ORGANIZAÇÃO DO PROJETO

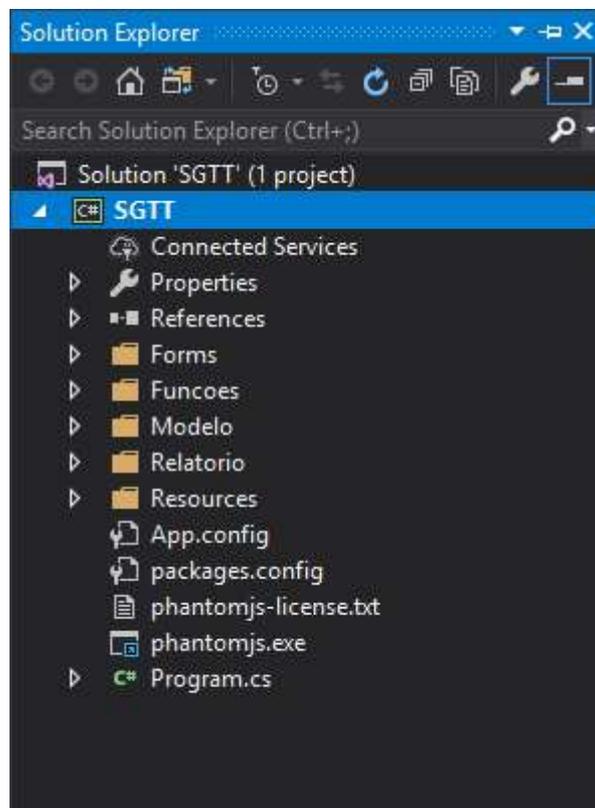


Figura 32 - Estrutura do Projeto

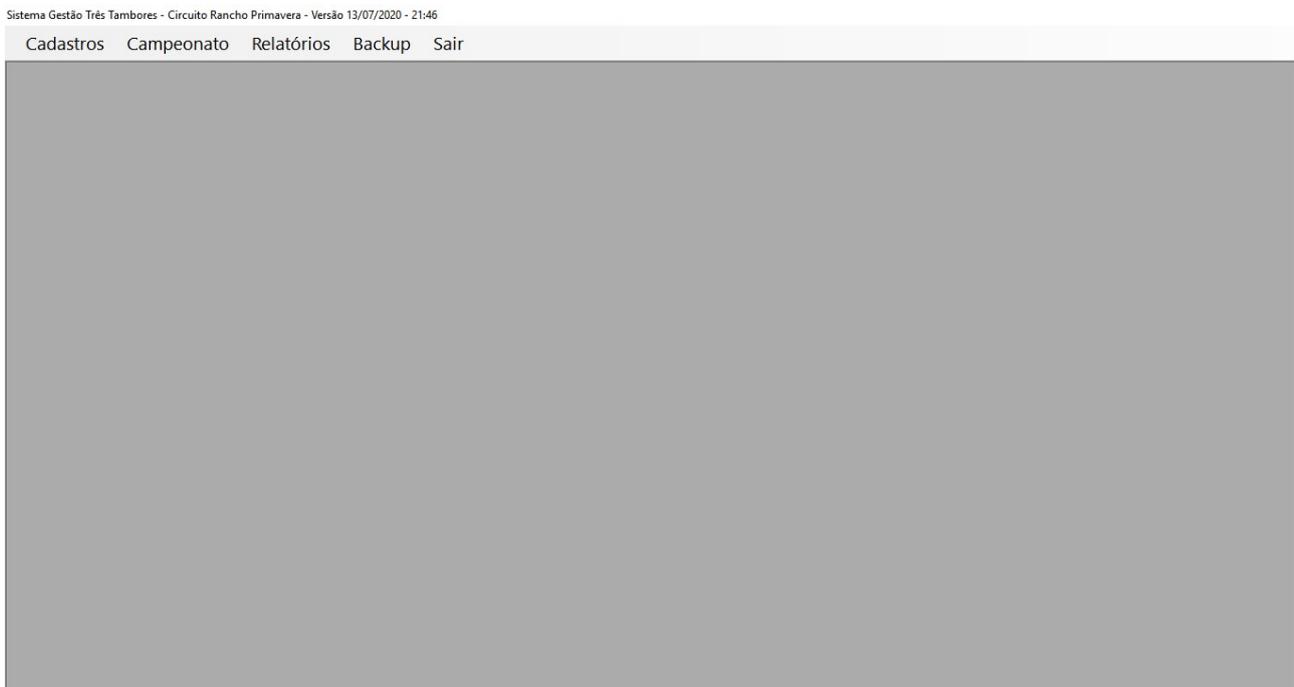
Para facilitar o desenvolvimento do projeto, melhorar sua organização e facilitar sua posterior manutenção, dividiu-se em algumas pastas (Figura 32):

- Forms: todas as interfaces gráficas que servirão de meio para que o usuário utilize a aplicação;

<sup>11</sup> [https://www.apachefriends.org/pt\\_br/index.html](https://www.apachefriends.org/pt_br/index.html)

- Funcoes: cálculos referentes a funcionalidades de prova e/ou cálculos diversos de rankings, pontuações, classificações e etc;
- Modelo: todas as classes utilizadas na aplicação;
- Relatorio: códigos responsáveis pela emissão de relatórios;
- Resources: pasta padrão de um projeto C#, onde ficam localizadas imagens que serão utilizadas na interface, tal como ícones que foram personalizados.

## 4.2. TELA INICIAL



**Figura 33 - Tela Inicial SGTT**

Na Figura 33, podemos ver a tela inicial da aplicação, onde é possível verificar os principais menus. Foram divididos conforme sua maior necessidade e relevância, para que seja de fácil utilização para o usuário, sem que seja necessário um treinamento intensivo.

### 4.3. CADASTRO ANIMAL

**Manter Animal**

ID:

Nome:

Sexo:

Cor:

Idade:

Peso:

Obs:




Criador:

Proprietário:

	ID	Nome	Sexo	Cor	Idade	Peso	Proprietário
▶	1	Alazão	M	Preto	3	350	Almir
	2	Sweet	F	Marrom	4	280	Rogério Paill

**Figura 34 - Cadastro Animal**

Na Figura 34, podemos observar a tela para cadastro de animais. Para se chegar nesta tela é necessário escolher a opção Cadastros e posteriormente a opção Animal. Podemos observar que já existem dois animais na lista, que podem ser editados e/ou removidos clicando duas vezes no respectivo animal e clicando no botão Remover após.

Nas próximas telas de cadastros básicos que aqui serão demonstradas, pode-se perceber uma certa padronização entre elas, objetivo esse para facilitar o uso da aplicação.

#### 4.4. CADASTRO DE BANCO

Manter Bancos

## Manter Bancos

ID:

Descrição:

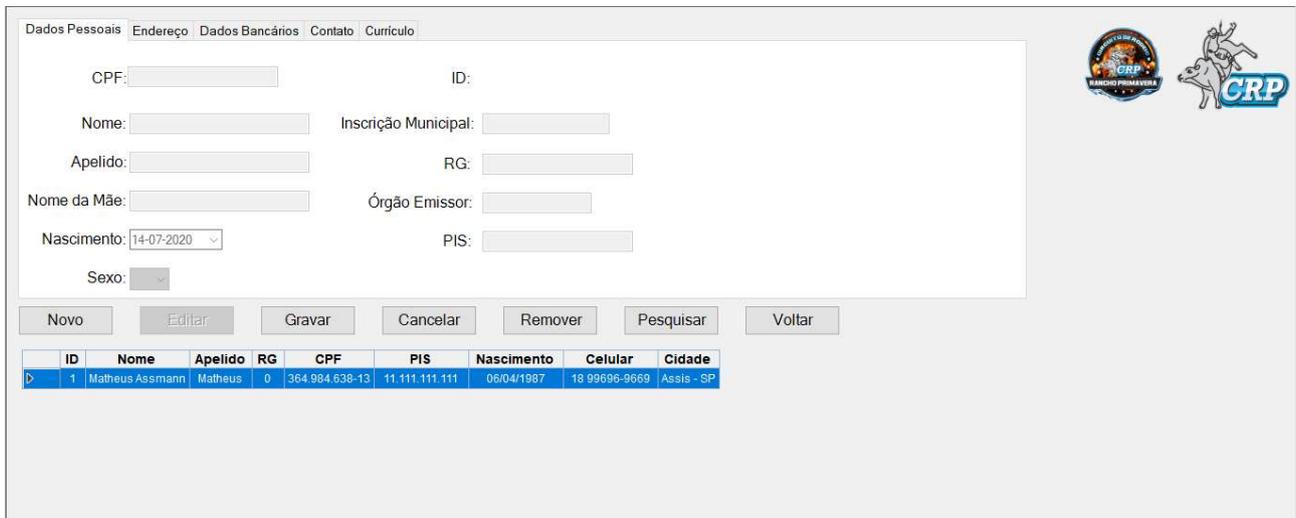
	ID	Descrição
▶	BB	Banco do Brasil
	BND	Banco Não Cadastrado
	ITU	Itaú
	NUB	Nubank



Figura 35 - Cadastro de Banco

Responsável para realizar cadastro de bancos (Figura 35), para que tenha-se conhecimento devido a premiações. Para acessar esta tela é necessário escolher a opção Banco dentro do menu de Cadastros.

## 4.5. CADASTRO DE COMPETIDOR



ID	Nome	Apelido	RG	CPF	PIS	Nascimento	Celular	Cidade
1	Matheus Assmann	Matheus	0	364.984.638-13	11.111.111.111	06/04/1987	18 99696-9669	Assis - SP

**Figura 36 - Cadastro Competidor 1**

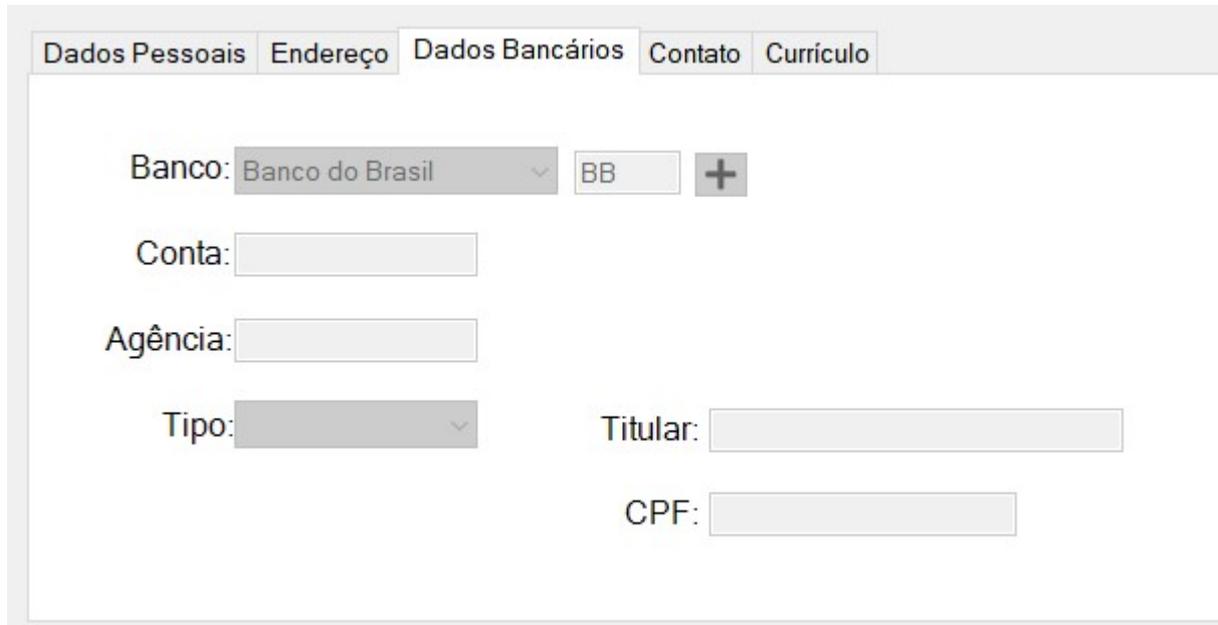
Cadastro onde ficam localizadas todas as informações de um competidor (Figura 36). Para acessá-lo é necessário escolher a opção Competidor dentro do menu de Cadastros.

### 4.5.1. CONTINUAÇÃO CADASTRO DE COMPETIDOR 1



**Figura 37 - Cadastro Competidor 2**

#### 4.5.2. CONTINUAÇÃO CADASTRO DE COMPETIDOR 2

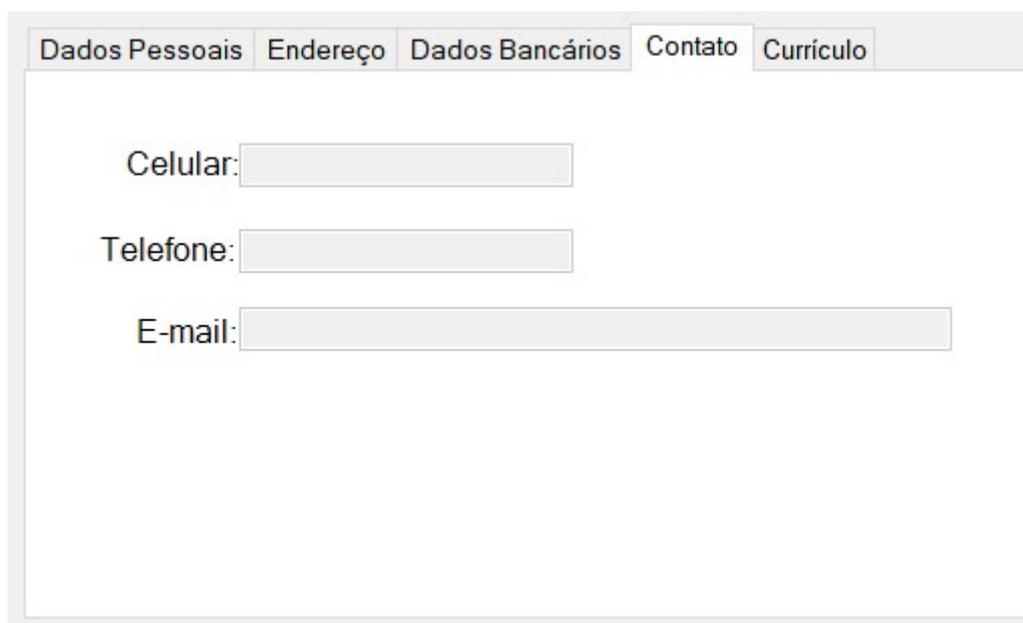


The image shows a web form for entering banking information. At the top, there are five tabs: 'Dados Pessoais', 'Endereço', 'Dados Bancários', 'Contato', and 'Currículo'. The 'Dados Bancários' tab is selected. Below the tabs, the form contains the following fields:

- Banco:** A dropdown menu showing 'Banco do Brasil' with a downward arrow, followed by a small box containing 'BB' and a plus sign (+).
- Conta:** A text input field.
- Agência:** A text input field.
- Tipo:** A dropdown menu with a downward arrow.
- Titular:** A text input field.
- CPF:** A text input field.

Figura 38 - Cadastro Competidor 3

#### 4.5.3. CONTINUAÇÃO CADASTRO DE COMPETIDOR 3



The image shows a web form for entering contact information. At the top, there are five tabs: 'Dados Pessoais', 'Endereço', 'Dados Bancários', 'Contato', and 'Currículo'. The 'Contato' tab is selected. Below the tabs, the form contains the following fields:

- Celular:** A text input field.
- Telefone:** A text input field.
- E-mail:** A text input field.

Figura 39 - Cadastro Competidor 4

#### 4.5.4. CONTINUAÇÃO CADASTRO DE COMPETIDOR 4

Dados Pessoais	Endereço	Dados Bancários	Contato	Curriculo
Qtde de Títulos: <input type="text"/>		Frente		Perfil
Descrição: <input type="text"/>				

Figura 40 - Cadastro Competidor 5

#### 4.6. CADASTRO DE CAMPEONATO

Manter Campeonato




ID: \_\_\_\_\_

Descrição:

Organizador:

Data Início:

Data Fim:

	ID	Descrição	Organizador	Data Início	Data Fim
▶	1	teste	matheus	13/07/2020	17/07/2020
	2	Campeonato Paulista	Rogério Paitl	14/07/2020	20/07/2020

Figura 41 - Cadastro de Campeonato

Para se chegar a tela visualizada acima é necessária selecionar a opção Campeonato, dentro do menu de Campeonato (Figura 41).

#### 4.7. CADASTRO DE CATEGORIA

Manter Categoria

ID:

Descrição:

Qtde de Premiados:

Nº de Dias:

Qtde de Passadas:

Valor:

	ID	Descrição	Qtde Premiados	Nº Dias	Qtde Passadas	Valor
▶	1	Feminino	10	3	3	R\$150,00
	2	Free Horse	10	1	1	R\$100,00
	3	Juvenil	10	1	3	R\$130,00

Figura 42 - Cadastro de Categoria

Para que seja possível acessar esta tela é necessário selecionar a opção de Categoria, dentro do menu de Campeonato (Figura 42).

## 4.8. CADASTRO DE ETAPA

Etapa Categoria Dia Passadas

ID:

Campeonato: Campeonato Paulista Contratante:

Número:  Cidade: Assis - SP

Descrição:  Prêmio R\$:

Data Início: 14/07/2020  Qtde Premiados:

Data Fim: 14/07/2020  Tipo Cálculo:

Novo Editar Cancelar Gravar Remover Pesquisar

ID	Campeonato	Nº da Etapa	Data Início	Data Fim	Contratante	Cidade	Prêmio	Qtde Premiados	Cálculo
2	Campeonato Paulista	2	14/07/2020	15/07/2020	Rogério Paitl	Assis - SP	R\$3.000,00	3	0
1	teste	1	13/07/2020	17/07/2020	matheus	Assis - SP	R\$50.000,00	10	0

**Figura 43 - Cadastro de Etapa**

Esta tela é responsável pela maior funcionalidade do sistema e será discutida detalhadamente para melhor entendimento. Para título de curiosidade, o código responsável por essa funcionalidade estará disponível no final do trabalho. Para acessar essa função é necessário selecionar a opção Etapa, dentro do menu de Campeonato (Figura 43).

Podemos observar que informações como campeonato e cidade devem ter sido cadastrados anteriormente para que se possa cadastrar uma nova etapa.

#### 4.8.1. Continuação cadastro de Etapa 1

Adicionar Categoria

Categoria: Feminino +

Nº de Dias: 3

Adicionar

Prêmio

	ID	Categoria	Nº Dias	
▶	1	Feminino	3	X

Figura 44 - Cadastro de Etapa 1

Após cadastrada uma nova etapa, como explicado anteriormente, é necessário que o usuário a selecione na lista abaixo com um duplo clique, como se fosse editá-la, e após selecionar a aba de Categoria exibida na parte superior da tela (Figura 44).

Aqui é necessário selecionar uma categoria e a quantidade de dias que será disputado. O ícone localizado a direita da categoria selecionada, (+) funciona como um atalho, que abre a tela para cadastro de categoria, sem que seja necessária voltar manualmente.

Incluindo a categoria e a quantidade de dias, o usuário deverá pressionar o botão Adicionar, para que ela seja visível na lista abaixo. Com um duplo clique na categoria desejada, será redirecionado para a aba de Dia, localizada a direita desta.

## 4.8.2. Continuação cadastro de Etapa 2

Etapa Categoria Dia Passadas

Campeonato: **Campeonato Paulista**

Etapa: **2 - Eliminatória de Assis - Assis - SP**

Categoria: **Feminino**

Conjunto

Competidor:  + Animal:  +

Nome	Cidade
Matheus Assmann	Assis-SP

Nome	Sexo
Alazão	M
Sweet	F

Qtde Competidores: 1

Conjunto:

Competidor	Animal	Patrocinador	Valor
Matheus Assmann	Alazão	SimpleTech	R\$150,00 <span style="color: red;">X</span>

Patrocinador:

Inscrição R\$:  Gravar

Novo Dia

Total Inscrições R\$: 150

ID	Dia	Passadas
1	14/07/2020	1 <span style="color: red;">X</span> Passadas

**Figura 45 - Cadastro de Etapa 2**

Nesta tela, será possível ver a relação de todos os competidores e animais cadastrados no sistema. Podemos ver um campo de pesquisa dinâmico em todas as informações desta tela (competidor, animal e conjunto), sendo que, conforme for digitada a informação necessária, o campo respectivo vai filtrando somente os resultados encontrados, facilitando encontrar o dado desejado no caso de uma grande quantidade de informações cadastradas. Lembrando também do ícone de (+) localizado a direita do campo de pesquisa em competidor e animal, tendo a mesma funcionalidade de facilitador de cadastramento utilizado anteriormente em outras telas (Figura 45).

Para se incluir um conjunto, é necessário selecionar o respectivo competidor e animal em suas listas, o campo de patrocinador é opcional e é recuperado o valor da inscrição que foi previamente cadastrado no valor no cadastro de categoria. Selecionando o competidor e animal desejado, o usuário então deverá pressionar o botão Gravar, para que o conjunto seja criado e exibido a direita, caso não haja uma exceção<sup>12</sup>. Caso tenha sido incluído um conjunto errado, basta pressionar o botão vermelho (X) para que seja removido.

<sup>12</sup> Na prova de Três Tambores, um competidor pode participar com vários animais, porém um animal só participará com um único competidor. Caso seja selecionado um animal para mais de um competidor, o sistema avisará com uma mensagem de erro, e não finalizará a transação de inclusão de conjunto.

Após incluídos os conjuntos que participarão da prova, o usuário deverá pressionar o botão de Novo Dia, para que sejam iniciados os dias da prova, dias estes que foram selecionados no início deste tópico de Etapa, ao cadastrá-la.

Com os dias de prova criados, quantidade de passadas selecionadas, o usuário deverá pressionar o botão Passadas, para gerenciar as passadas daquele dia.

### 4.8.3. Continuação cadastro de Etapa 3

The screenshot shows a web interface for managing 'Passadas' (runs) for a specific event. The interface includes the following elements:

- Navigation:** Tabs for 'Etapa', 'Categoria', 'Dia', and 'Passadas'.
- Event Information:**
  - Campeonato: Campeonato Paulista
  - Etapa: 2 - Eliminatória de Assis - Assis - SP
  - Categoria: Feminino
  - Dia: 14/07/2020
- Logos:** Logos for 'CRP' and 'CRP BRASIL' are visible in the top right.
- Form Fields:**
  - 'Passada' dropdown menu set to '1'.
  - 'Quantidade Competidores' input field set to '1'.
  - 'Tempo de Corte' input field.
  - 'Qtd Classificados' input field with an 'Alterar' button.
- Action Buttons:** 'Nova Passada', 'Gravar', 'Antes Passada', 'Depois Passada', 'Desclassificar', 'Reclassificar', and 'Tela Classificação'.
- Table:** A table with columns 'Posº', 'Competidor', 'Animal', and 'Passada 1'. It contains one entry:
 

Posº	Competidor	Animal	Passada 1
1	Matheus Assmann	Alazão	0,000
- Right Panel:** A panel with columns 'ID', 'Compete', and 'Animal'.

Figura 46 - Cadastro de Etapa 3

Nesta tela é possível gerenciar todos os tempos e passadas do dia selecionado. Todos os conjuntos cadastrados anteriormente neste dia, serão visualizados nesta lista. Para iniciar a primeira passada é necessário pressionar o botão Nova Passada e em seguida Gravar. Com isso serão vinculados os conjuntos a este dia. Na aba de Tempo de Corte, é possível selecionar a quantidade de competidores que serão classificados para uma próxima passada, basta pressionar no botão alterar e definir a quantidade (10) e sair da caixa de texto com a tecla TAB. Com isso o sistema filtrará com base na quantidade escolhida, calculando o tempo de corte necessário para um conjunto faça parte dos classificados. Caso um competidor seja classificado com mais de uma animal, ele deverá escolher com qual animal continuará na prova, e dará lugar para um outro competidor suba, no caso alguém que estivesse em décimo primeiro lugar, passará a décima e será classificado (Figura 46).

Após selecionar o conjunto que permanecerá na disputa, o conjunto adjacente deverá ser selecionado com um único clique e em seguida pressionar o botão Desclassificar, para que não faça mais parte da disputa.

Com isso, será possível atribuir o tempo da Passada 1 para cada conjunto. Para melhorar a interação do sistema com o público que acompanha a prova, é possível gerar telas para exibição em dispositivos audio-visuais. Para usufruir desta funcionalidade, o usuário deverá selecionar o conjunto com um único clique e após pressionar o botão Antes Passada.

Para gerar o seguinte telão, podemos observar um trecho do código abaixo (Figura 47):

#### 4.8.3.1. gerarAntesPassada()

```
public static void gerarAntesPassada(int idPassada)
{
    Modelo.SGTTContexto contexto = new Modelo.SGTTContexto();
    Modelo.Passadas passada = contexto.Passadas.Find(idPassada);
    string nomeArq = "Banner/aPassada.html";
    using (StreamWriter html = new StreamWriter(nomeArq))
    {
        gerarInicioHtmlBanner(html, "aPassada.css", "aPassada.jpg");
        antesMontariaHtml(html, passada);
        gerarFimHtmlBanner(html);
    }
    string nomeImagem = "aPassada";
    phantomImage("aPassada", nomeImagem, 1920, 1080, true);
}
```

Figura 47 - Código Antes Passada

Podemos observar o nome da função, juntamente com o parâmetro necessário para que ela funcione corretamente. Para cada tela, também é necessário um arquivo .jpg (fundo da imagem) e um arquivo .css (responsável pelo alinhamento dos elementos). As funções gerarInicioHtmlBanner(), na Figura 48, e a gerarFimHtmlBanner(), na Figura 49, simplesmente são responsáveis em fechar as tags corretamente, para que não haja repetição desnecessária de código.

#### 4.8.3.2. gerarInicioHtmlBanner()

```
public static void gerarInicioHtmlBanner(StreamWriter html, string cssNome, string imgSrc)
{
    html.WriteLine("<html>");
    html.WriteLine("<head>");
    html.WriteLine("<meta http-equiv=\"Content-Type\" content=\"text/html; charset=utf-8\">");
    html.WriteLine("<meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0\" >");
    html.WriteLine("<link rel=\"stylesheet\" type=\"text/css\" href=\"css/\" + cssNome + \"\" />");
    html.WriteLine("</head>");
    html.WriteLine("<body>");
    html.WriteLine("<div class=\"image\">");
    html.WriteLine("<img src=\"img/\" + imgSrc + \"\" />");
}
```

Figura 48 - gerarInicioHtmlBanner()

#### 4.8.3.3. gerarFimHtmlBanner()

```
public static void gerarFimHtmlBanner(StreamWriter html)
{
    html.WriteLine("</div>");
    html.WriteLine("</body>");
    html.WriteLine("</html>");
}
```

Figura 49 - gerarFimHtmlBanner()

Com essa função, utilizando a biblioteca StreamWriter, é possível criar um html linha a linha, com os parâmetros desejados, no caso do exemplo, com o idPassada. Ambas as funções de geração de tela Antes e Depois utilizam este mesmo corpo, tendo como diferença apenas seu conteúdo. Podemos observar um trecho de código da função antesMontariaHtml(), na Figura 50, e na sequência, um trecho de código da função depoisMontariaHtml(), na Figura 51.

#### 4.8.3.4. antesMontariaHtml()

```

string arq = @"Banner\img\frente\" + passada.conjunto.competidor.id + ".png";
if (File.Exists(arq))
    html.WriteLine("<img class=\"imgCowgirl\" src=\"" + arq.Replace(@"Banner\", "") + "\"/>");
else if (passada.conjunto.competidor.tipoPessoa == "F")
{
    html.WriteLine("<img class=\"imgCowgirlSombra\" src=\"img/cowGirl.png\"/>");
}
else
{
    html.WriteLine("<img class=\"imgCowgirlSombra\" src=\"img/cowboy.png\"/>");
}
html.WriteLine("<div class=\"dEtapaCidade\">" + passada.conjunto.etapaCategoria.etapa.numero + "ª ETAPA" + " - PASSADA " + passada.passada + "</div>");
if (passada.conjunto.competidor.nome.Count() < 20)
{
    html.WriteLine("<div class=\"dNome\">" + passada.conjunto.competidor.nome + "</div>");
}
else if (passada.conjunto.competidor.nome.Count() >= 20 && passada.conjunto.competidor.apelido != "")
{
    html.WriteLine("<div class=\"dNome\">" + passada.conjunto.competidor.apelido + "</div>");
}
else if (passada.conjunto.competidor.nome.Count() >= 20 && passada.conjunto.competidor.apelido == "")
{
    html.WriteLine("<div class=\"dNome20\">" + passada.conjunto.competidor.nome + "</div>");
}
html.WriteLine("<div class=\"dCidade\">" + passada.conjunto.competidor.cidade.nome + " - " + passada.conjunto.competidor.cidade.uf + "</div>");
html.WriteLine("<div class=\"dColocacao\">" + pos.ToString() + "º Pos." + "</div>");

```

Figura 50 - Trecho antesMontariaHtml()

#### 4.8.3.5. depoisMontariaHtml()

```

html.WriteLine("<div class=\"dColocacao\">" + pos.ToString() + "º Pos." + "</div>");
html.WriteLine("<div class=\"passada\">PASSADA: " + passada.passada + "</div>");
if (acumulado > 0)
{
    if (passada.passada == 3)
    {
        if (etapa.tipocalculo == 0)
        {
            html.WriteLine("<div class=\"dTextoAcumulado\">" + "ACUMULADO" + "</div>");
        }
        else if (etapa.tipocalculo == 1)
        {
            html.WriteLine("<div class=\"dTextoAcumulado\">" + "ACUMULADO" + "</div>");
        }
        html.WriteLine("<div class=\"dAcumulado2\">" + acumulado.ToString("0.000") + "</div>");
    }
    else
    {
        if (etapa.tipocalculo == 0)
        {
            html.WriteLine("<div class=\"menorTempo\">" + impAcumulado + "</div>");
            html.WriteLine("<div class=\"dTextoAcumulado\">" + "MENOR TEMPO" + "</div>");
        }
        else if (etapa.tipocalculo == 1)
        {
            html.WriteLine("<div class=\"dAcumulado2\">" + acumulado.ToString("0.000") + "</div>");
            html.WriteLine("<div class=\"dTextoAcumulado\">" + "ACUMULADO" + "</div>");
        }
    }
}
else html.WriteLine("<div class=\"dAcumulado\">" + "-" + "</div>");
html.WriteLine("<div class=\"dDiferenca\">" + passada.tempo.ToString("0.000") + "</div>");
html.WriteLine("<div class=\"dTempo\">TEMPO PASSADA" + "</div>");

```

Figura 51 - Trecho depoisMontariaHtml()

#### 4.8.4. Tela Antes Passada



Figura 52 - Antes Passada 1

Com isso fica fácil a visualização do competidor, seu animal e os tempos de prova. Por se tratar da tela de ANTES da passada (Figura 52), nenhuma informação será exibida, e no atual momento este conjunto estará liderando a prova. O logo localizado a esquerda é fixo na própria imagem gerada, e o logo a direita é o respectivo a etapa atual. Caso o competidor não tenha nenhuma foto cadastrada no sistema, a imagem acima foi determinada como padrão.

Para lançar a nota que o conjunto teve na passada, basta pressionar abaixo do cabeçalho de Passada 1, com isso, o campo ficará habilitado, sendo o padrão dos tempos com três casa decimais, separados por vírgula (0,000).

#### 4.8.5. Tela Depois Passada

Depois de lançado o tempo, é possível visualizar uma nova tela referente ao tempo que o conjunto teve nesta passada. Para visualizá-la, deve-se selecionar novamente o conjunto com um único clique e após pressionar o botão Depois Passada (Figura 53).

Figura 53 - Depois Passada 1

Por se tratar da primeira passada, o tempo atual do competidor é também o seu menor tempo. Com isso ele lidera essa passada.

#### 4.8.6. Segunda Passada

Etapa Categoria Dia Passadas

Campeonato: **Campeonato Paulista**

Etapa: **2 - Eliminatória de Assis - Assis - SP**

Categoria: **Feminino**

Dia: **14/07/2020**

Passada: **2**

Quantidade Competidores: **1**

Tempo de Corte:

Qtd Classificados:

Tempo de Corte:

	Posº	Competidor	Animal	Passada 1	Passada 2	Menor Tempo
▶	1	Matheus Assmann	Alazão	19,450	0,000	19,450

Figura 54 - Segunda Passada

O procedimento na segunda passada é muito similar a da primeira. O usuário deverá pressionar em Nova Passada e posteriormente em Gravar. Com isso o campo de Passada 2 ficará visível assim como o de Menor Tempo, (que geralmente é o menor tempo dentre as duas primeiras passadas) visível na Figura 54.

#### 4.8.7. Tela Antes Passada 2



Figura 55 - Antes Passada 2

Antes de iniciar a segunda passada, o sistema trará o tempo obtido na primeira passada, assim como o tempo necessário para liderar a prova (Figura 55).

#### 4.8.8. Tela Depois Passada 2



Figura 56 - Depois Passada 2

Neste caso, a passada do competidor recebeu o tempo de 120,000. Chamamos esta ocorrência de SAT (Sem Aproveitamento Técnico), e recebe este tempo por padrão. Ocorre em casos de ter derrubado algum tambor, ou de ter machucado o animal durante o percurso (Figura 56).

### 4.8.9. Terceira Passada (Final)

Etapa Categoria Dia Passadas

Campeonato: **Campeonato Paulista**

Etapa: **2 - Eliminatória de Assis - Assis - SP**

Categoria: **Feminino**

Dia: **14/07/2020**

Passada

Quantidade Competidores

Tempo de Corte:

Qtd Classificados:

Tempo de Corte:

	Pos°	Competidor	Animal	Passada 1	Passada 2	Menor Tempo	Acumulado	Passada 3
▶	1	Matheus Assmann	Alazão	19,450	120,000	19,450	19,450	0,000

**Figura 57 - Terceira Passada**

Para criar a terceira passada, repetiremos mais uma vez os comando de pressionar em Nova Passada e em seguida em Gravar. Assim o campo de Acumulado e Passada 3 ficarão visíveis. Perceba que mesmo não tendo um bom desempenho na segunda passada, o menor tempo do competidor ainda permanece competitivo, com isso ele ainda terá chances de vencer (Figura 57).

#### 4.8.10. Antes Passada 3



Figura 58 - Antes Passada 3

Na tela de antes da passada da final é exibido o melhor tempo do competidor (entre as duas primeiras passadas), na Figura 58, acima.

#### 4.8.11. Depois Passada 3



Figura 59 - Depois Passada 3

Finalizada a terceira e última passada, essa é a tela final para o competidor naquele dia de prova. Podemos ver que o seu tempo acumulado é o resultado do seu menor tempo mais o tempo em que teve na última passada (Figura 59).

#### 4.8.12. Tela Classificação



Figura 60 - Tela Classificação

Depois de finalizada cada passada, é possível gerar uma tela com o ranking da competição até o atual momento. Para gerar essa tela é necessário pressionar o botão Tela Classificação, e após selecionar a tela para ser gerada com as respectivas posições (Figura 60).

## 5. CONCLUSÃO

Devido a constante evolução da tecnologia, é unânime em se concluir que empresas que não utilizam sistemas de informação no seu dia-a-dia correm grandes riscos de serem “engolidas” pelos seus concorrentes.

Ao se tratar do evento de Três Tambores, não é diferente. O objetivo principal desse trabalho foi de desenvolver e oferecer um sistema capaz de lidar com todo o gerenciamento de prova, para auxiliar àqueles responsáveis por sua operação e também para exibir informações para os espectadores.

No início do desenvolvimento deste projeto, devido a necessidade de ter-se algo palpável rapidamente, não foi realizado um levantamento de requisitos muito detalhado, deixando de lado a criação de alguns diagramas de UC, sequência e etc. Devido a isso, e a falta de objetividade no início do projeto, por se tratar de algo novo, houve a necessidade de retrabalho em algumas funcionalidades do sistema, a fim de deixá-las mais fáceis e intuitivas para àquele que for operá-lo. Pode-se dizer que desenvolvimento e análise, após essas dificuldade iniciais, caminharam lado a lado, visto que a compreensão de todo o funcionamento de prova tornou-se mais claro para todas as partes envolvidas no projeto.

A utilização do C# e .NET foi vital para a entrega rápida do sistema, visto que a linguagem oferece uma grande facilidade na criação da interface gráfica (telas), economizando muitas linhas de código que a própria IDE do Visual Studio é capaz de gerá-los para diminuir a carga sobre o programador. Destaca-se também pela fácil manutenção do código e fácil depuração (encontrar erros). Até o atual momento, de todas as linguagens que utilizei, o C# é, sem dúvida, um grande aliado para se criar um sistema com interface gráfica de maneira simplificada e rápida, sem a necessidade de profundos conhecimentos da linguagem.

Com a finalização deste sistema, espera-se agilizar todo o processo de uma prova de Três Tambores para a equipe do CRP. Visto que quando era feito de maneira manual, o tempo para cadastrar participantes era demasiado longo e caótico. Além de auxiliar durante toda a prova em si, gerando relatórios para facilitar a visualização da classificação

e com a geração das imagens para os telões, fazendo com que todos os espectadores da prova, acompanhem todas as suas etapas facilmente.

O sistema ainda é muito novo, e, com certeza existem diversas melhorias que podem ser implementadas para que ele auxilie cada vez mais o gerenciamento de uma prova. Integrações com sistemas web/móvel, capacidade de um cadastramento de participantes antecipado (via web) e importação/exportação de dados via excel são algumas melhorias que já foram pensadas em serem implementadas, e com certeza melhorarão muito mais a fluidez e sincronia entre sistema e prova.

## REFERÊNCIAS BIBLIOGRÁFICAS

DOMINGUES, Jenifer Teixeira. **A importância dos sistema de informação gerencial para as empresas.** Disponível em <<http://www.administradores.com.br/artigos/academico/a-importancia-dos-sistema-de-informacao-gerencial-para-as-empresas/78358/>> Publicado em 25 Jun. 2014. Acesso em: 10 Nov, 2019.

LAUDON, Kenneth C.; LAUDON, Jane Price; **Sistemas de Informação Gerenciais.** São Paulo: Pearson Prentice Hall, 2010.

LOH, Stanley; **31 tipos de sistemas de informação - 31 maneiras de a tecnologia da informação ajudar as organizações.** Porto Alegre, 2014.

MANZANO, José Augusto N. G. **MySQL 5.5 interativo: guia essencial de orientação e desenvolvimento** / José Augusto N. G. Manzano. -- 1. ed. -- São Paulo : Érica, 2011.

MATOS, M. P; BERMEJO, P.H.S; SALM JUNIOR, J.F. **Gerência de Risco em Projetos de Software:** Baseada nos modelos de Processos de Referência PMBOK, CMMI, MPS.BR, TenStep e ISO 12207. 1.Ed. Rio de Janeiro: Editora Ciência Moderna Ltda, 2010.

PEREIRA E SILVA, Ricardo; **Como modelar com um UML 2.** Visual Books, 2009.

STELLMAN, Andrew; GREENE, Jennifer; **Use a cabeça! C#.** Alta Books, 2011.

Tudo o que você precisa saber sobre a prova de três tambores. **Blog Rodeo West**, 2020. Disponível em: <<https://blog.rodeowest.com.br/mundo-country/tudo-sobre-prova-de-tres-tambores/#:~:text=O%20primeiro%20tambor%20deve%20ser,para%20a%20linha%20de%20chegada/>> Acesso em: 19 de ago. de 2020.

## 6. ANEXOS

### 6.1. ANEXO A – CRONOGRAMA DE TAREFAS REALIZADAS

Cronograma	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago
Definição do Pré-Projeto	■										
Entrega do Pré-Projeto	■	■									
Análise e Validação de Requisitos		■	■	■							
Análise e Especificação		■	■	■	■						
Diagramas UML e MER		■	■	■	■	■					
Entrega da Qualificação		■	■	■	■	■	■				
Preparação para Qualificação					■	■	■				
Exame de Qualificação						■	■	■	■		
Desenvolvimento do Sistema		■	■	■	■	■	■	■	■	■	■
Implementação e Testes		■	■	■	■	■	■	■	■	■	■
Apresentação ao Orientador				■	■	■	■	■	■	■	■
Solicitação de Defesa										■	■
Entrega do TCC											■
Apresentação Final											■

### 6.2. CÓDIGO DE NOVAPASSADA

#### 6.2.1. Primeira Parte

```
private void newPassada(int numpassada, int selecionados)
{
    Modelo.SGTTContexto contexto = new Modelo.SGTTContexto();
    lstPassadas = contexto.Passadas.Where(passada => passada.etapaDiaID == idEtpDia && passada.conjunto.ativo == true).OrderBy(passada =>
    passada.tempo).ToList();
    if (numpassada == 1)
    {
        var dados = from conjunto in contexto.Conjunto.Where(conjunto => conjunto.etapacategoriaID == idEtpCategoria).ToList()
        select new
        {
            id = conjunto.id,
            competidor = conjunto.competidor.nome,
            animal = conjunto.animal.nome,
            tempo = 0
        };

        dgvPassadas.DataSource = dados.ToList();
        txtQtdCompete.Text = dados.Count().ToString();
        lstPassadas.Clear();

        foreach (var dado in dados)
        {
            Modelo.Passadas passada = new Modelo.Passadas();
            passada.passada = 0;
            passada.etapaDiaID = idEtpDia;
            passada.conjuntoID = dado.id;
            passada.tempo = 0;
            lstPassadas.Add(passada);
        }
    }
}
```

### 6.2.2. Segunda Parte

```
else //passada 2 ou 3
{
    numpassada--;
    List<Modelo.Resultado> lstResult = Relatorio.Consultas.consultaResultado(idEtpDia, numpassada);
    lstResult = lstResult.Where(a => a.conjunto.ativo == true).ToList();

    if (numpassada == 1)
    { //passada2
        lstResult.OrderBy(r => r.tempo1);
    } //gerando passada 2
    if (numpassada == 2)
    {
        if (tipoCalculo == 1) // 1 = somatória
        {
            lstResult.OrderBy(r => r.somaTempo);
        }
        else if (tipoCalculo == 0) // 0 = classificatória
        {
            lstResult.OrderBy(r => r.menorTempo);
        }
    }
    if (numpassada == 3)
    {
        if (tipoCalculo == 1)
        {
            lstResult.OrderBy(r => r.somaTempo3);
        }
        else if (tipoCalculo == 0)
        {
            lstResult.OrderBy(r => r.somaTempo);
        }
    }
}
```

### 6.2.3. Terceira Parte

```
int cont = 0;
var dados = from passadas in lstResult

    select new
    {
        id = ++cont,
        numpassada = numpassada,
        etapadia = idEtpDia,
        conjunto = passadas.conjunto,
        conjuntoid = passadas.conjunto.id,
        competidor = passadas.conjunto.competidor.nome,
        tempo = (tipoCalculo==1)? ((numpassada == 1) ? passadas.tempo1 : passadas.menorTempo): ((numpassada == 1) ? passadas.tempo1 :
            passadas.tempo2),
        animal = passadas.conjunto.animal.nome,
        somaTempo = passadas.somaTempo
    };

lstPassadas.Clear();
int contar = 0;
if (numpassada == 1)
{
    dados.OrderBy(r => r.tempo);
}
if (numpassada == 2)
{
    if (tipoCalculo == 1) // 1 = somatória
    {
        dados.OrderBy(r => r.somaTempo);
    }
    else if (tipoCalculo == 0) // 0 = classificatória
    {
        dados.OrderBy(r => r.tempo);
    }
}
```

#### 6.2.4. Quarta Parte

```
if (numpassada == 3)
{
    if (tipoCalculo == 1)
    {
        dados.OrderBy(r => r.somaTempo);
    }
    else if (tipoCalculo == 0)
    {
        dados.OrderBy(r => r.somaTempo);
    }
}
foreach (var dado in dados)
{
    if (contar < selecionados)
    {
        Modelo.Passadas passada = new Modelo.Passadas();
        passada.id = dado.id;
        passada.passada = dado.numpassada;
        passada.etapaDiaID = dado.etapadia;
        passada.conjuntoID = dado.conjuntoid;
        passada.tempo = dado.tempo;
        passada.conjunto = dado.conjunto;
        lstPassadas.Add(passada);
        contar++;
    }
}
var dadosGrid = from passada in lstPassadas.ToList()

                select new
                {
                    id = passada.id,
                    etapadia = passada.etapaDiaID,
                    conjuntoid = passada.conjuntoID,
                    competidor = passada.conjunto.competidor.nome,
                    animal = passada.conjunto.animal.nome,
                    tempo1 = passada.tempo,
                    menorTempo = passada.menorTempo,
                    somaTempo = passada.somaTempo
                };
```

### 6.2.5. Quinta Parte

```
if (numpassada == 1)
{
    dgvPassadas.DataSource = dadosGrid.OrderBy(d => d.tempo1).ToList();
}
if (numpassada == 2)
{
    if (tipoCalculo == 1)
    {
        dgvPassadas.DataSource = dadosGrid.OrderBy(d => d.somaTempo).ToList();
    }
    else if (tipoCalculo == 0)
    {
        dgvPassadas.DataSource = dadosGrid.OrderBy(d => d.menorTempo).ToList();
    }
}
if (numpassada == 3)
{
    if (tipoCalculo == 1)
    {
        dgvPassadas.DataSource = dadosGrid.OrderBy(d => d.somaTempo).ToList();
    }
    else if (tipoCalculo == 0)
    {
        dgvPassadas.DataSource = dadosGrid.OrderBy(d => d.somaTempo).ToList();
    }
}
txtQtdCompete.Text = dadosGrid.Count().ToString();
}
}
```