Universidade do Minho
Escola de Engenharia

Licenciatura em Engenharia Informática

22/23

Simulador de Campeonatos de Automobilismo

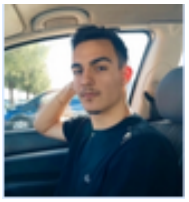
Entrega intermédia 1

Grupo 33

a94595
João Pedro Martins da Silva Freitas Cardoso

a96351
Luís Alberto Barreiro Araújo

a92945
José Miguel Carvalho Neiva



a95338
Mauricio Miranda Pereira

a87983
Pedro Calheno Pinto

<https://github.com/luisAraujo11/DSS-22-23>

Índice

1. Introdução

Com este trabalho prático pretende-se conceber um sistema que permita simular campeonatos de automobilismo, que segundo o enunciado deverá assemelhar-se a algo como o F1 Manager (famoso jogo de gestão de corridas).

Na prática este simulador/jogo permitirá que utilizadores registados como jogadores consigam competir em provas automobilísticas, ao mesmo tempo permitirá também, a utilizadores registados como administradores, criar/customizar um número diverso de variáveis, tais como, a classe do carro, o seu modelo, o piloto e as suas respetivas habilidades/perícias próprias, inclusive o próprio circuito que futuramente poderá jogar.

Esta primeira fase tem como objetivo modelar este sistema em modelos UML. Serão para isso apresentados o modelo de domínio e de use cases.

Para tentar perceber melhor como funciona este tipo de simulador, observamos online, parte da forma como é estruturado o jogo anteriormente referido, F1 Manager.

Por fim, após analisar todos os cenários providenciados no enunciado do trabalho, procedemos à realização dos modelos de domínio e use cases.

2. Modelo de domínio

Através de uma análise do enunciado proposto, detectamos as seguintes entidades principais:

- Administrador
- Jogador
- Campeonato
- Carro
- Circuito
- Piloto

Existem ainda as entidades, que podemos considerar “secundárias”, ou seja, que não são tão importantes para a estrutura principal do modelo, tais como:

- Resultados
- Acontecimento
- Volta
- Corrida
- Registo
- Cilindrada
- Distância
- Curva
- Reta
- Chicane
- GDU
- Potência
- Afinação

- Marca
- CTS
- SVA
- Informações
- Resultados
- Pneus
- C1
- C2
- GT
- SC
- Modelo
- PAC

- ModoMotor
- Acontecimento
- C1Hibrido
- C2Hibrido
- GTHibrido
- Sitio
- Sujeito
- Descrição
- MotorEletrico
- MotorCombustao
- Fiabilidade

De todas as entidades que observamos, temos duas principais, que irão interagir com o sistema proposto, o **Administrador** e o **Jogador**. No caso do **Administrador**, este faz login e tem como papel gerir um conjunto de entidades contidas no sistema, particularmente adicionar, remover ou alterar um ou mais **Carros**, **Pilotos**, **Campeonatos** e **Circuitos**. No caso do **Jogador**, após fazer login, este pode escolher, ou não, **Pilotos**, **Campeonatos**, **Pneus** e **Carros**.

A entidade **Campeonato** contém **Corridas** e determina os **Resultados** finais.

Temos também a entidade **Carro** que possui **Pneus**, uma dada **Cilindrada** associada, um **PAC**, uma **Fiabilidade**, um **Modelo,** um **motorCombustao** (que é definido por um **ModoMotor**) e uma **Marca**. Inclui também as classes **C1**, **C2**, **GT** e **SC**. Por sua vez, estas classes estão associadas às suas subclasses, **C1Hibrido**, **C2Hibrido** e **GTHibrido**, das quais também está associada uma **Potência** (relacionada juntamente com o **motorCombustao**). Ainda sobre as classes, temos a **Afinação** (exceto o para os **GT** e **SC**).

Na entidade Piloto, este apenas é identificado pelas suas perícias, **CTS** e **SVA**.

Abrangemos também a entidade **Corrida**, que providencia **Informações** e é composta por **Voltas** e **Circuitos**. A **Volta** é por sua vez composta por **Acontecimentos**, dos quais, estão descritos por um **Sítio**, um **Sujeito** e uma **Descrição**. Quanto ao **Circuito**, este mede uma **Distância** e é composto por **Curvas**, **Chicanes** e **Retas**, que são caracterizadas pela **GDU**.

Por último, podemos falar da entidade **Registo**, que vai apontar o **PAC**, o **Jogador**, o **Campeonato**, o **Circuito**, o **Carro** e o **Piloto**. Esta entidade terá como objetivo informar, a qualquer momento da simulação, após o início da mesma, os registos feitos sobre as entidades referidas anteriormente.

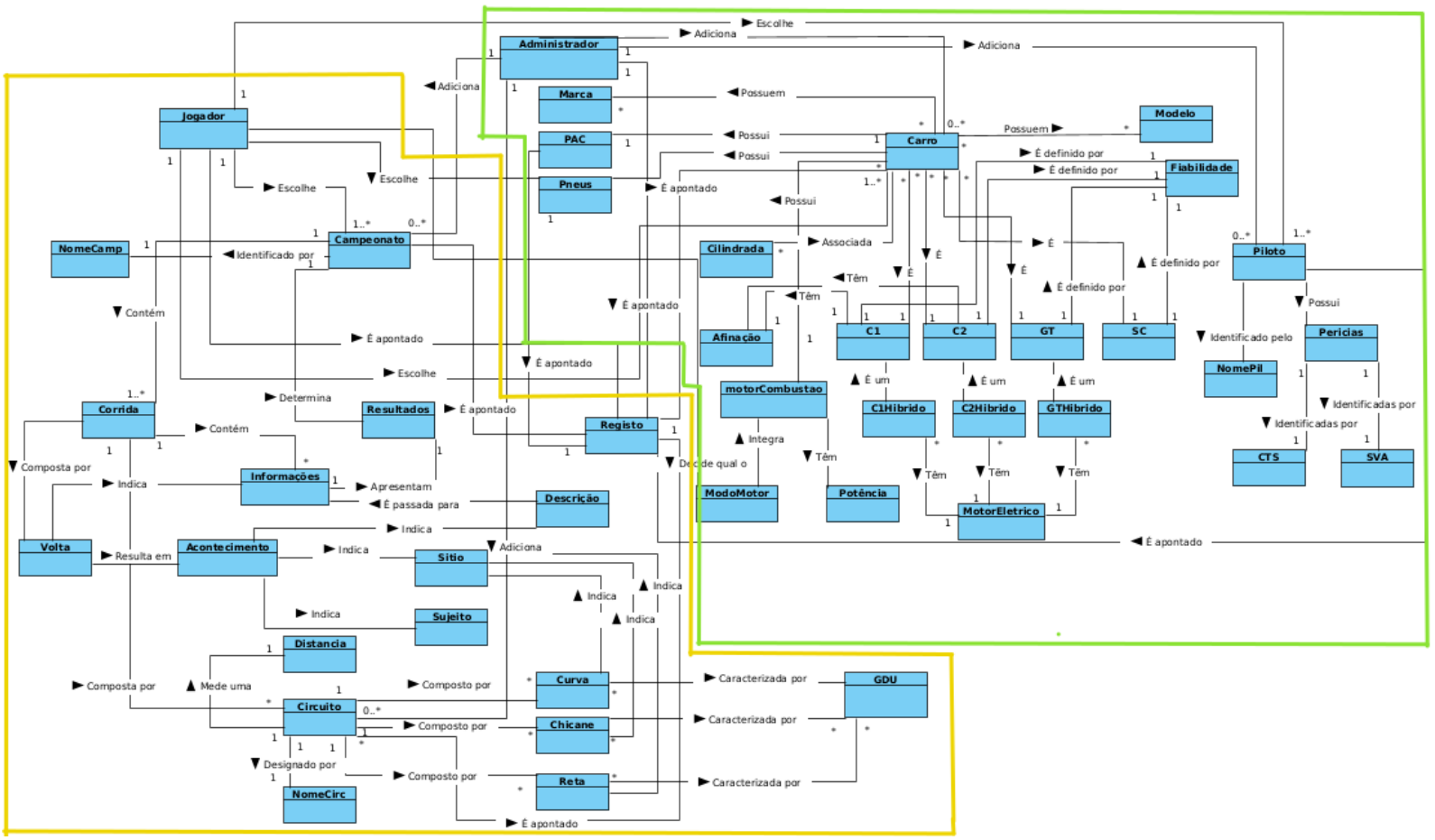


Fig.1 - Diagrama de Classes / Modelo de Domínio

Nota: A entidade **Potência** deveria estar ligada por um relacionamento à entidade **MotorEletrico**, apesar de estar corrigido no modelo do github, a imagem acima apresenta essa falha. A **Fiabilidade** também deveria estar ligada diretamente ao carro.

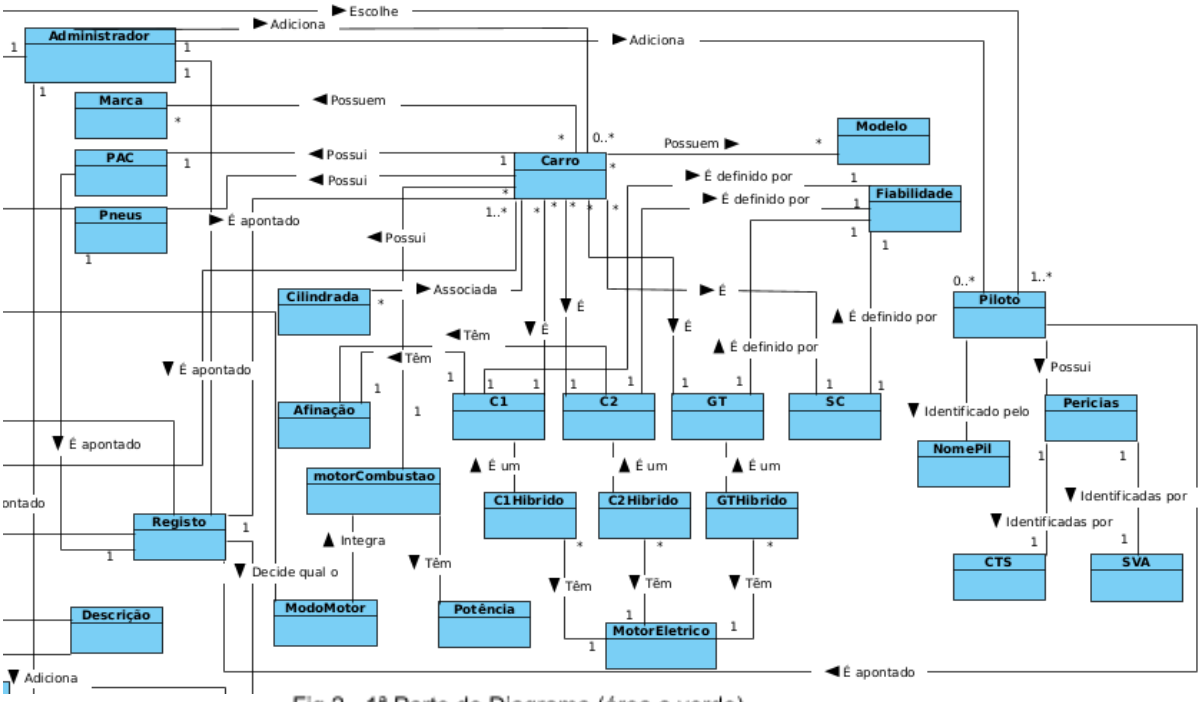


Fig.2 - 1ª Parte do Diagrama (área a verde)

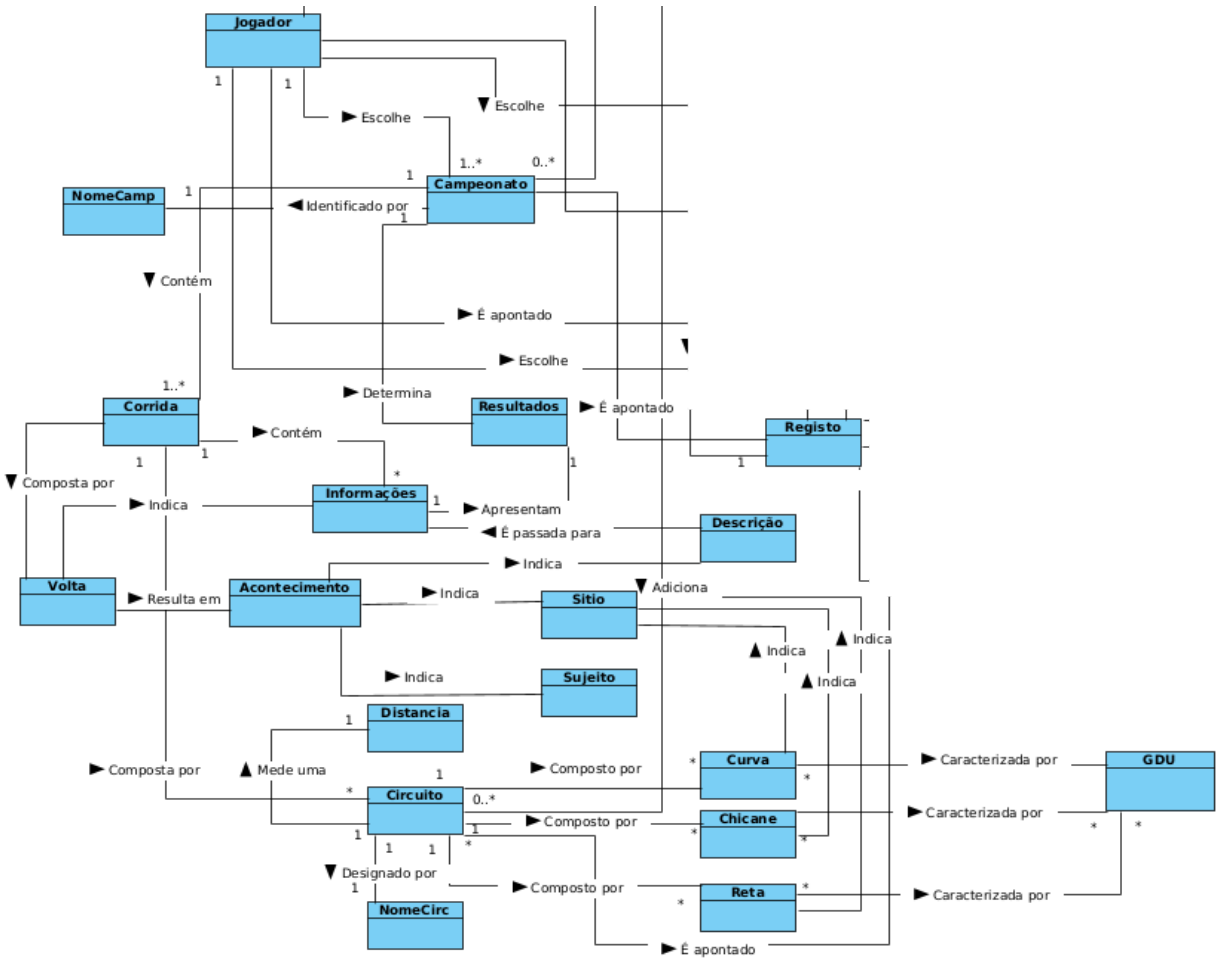


Fig.3 - 2ª Parte do Diagrama (área a azul amarelo)

3. Modelo de Use Case

3.1 Atores

Use Case para cada ator:

Jogador:

1. Simular um Campeonato
2. Consultar os Resultados
3. Escolher um ou mais Campeonatos

Administrador:

1. Gerir Piloto
    - Adicionar Piloto
    - Remover Piloto
    - Alterar Piloto
2. Gerir Campeonato
    - Adicionar Campeonato
    - Remover Campeonato
    - Alterar Campeonato
3. Gerir Carro
    - Adicionar Carro
    - Remover Carro
    - Alterar Carro
4. Gerir Circuito
    - Adicionar Circuito
    - Remover Circuito
    - Alterar Circuito

3.2. Use Cases

Aqui estão definidos os nossos Use Cases (em formato textual)

3.2.1 Administrador

Piloto

Use case: Adicionar Piloto

Descrição: O administrador regista um novo Piloto
Cenários: Cenário 4
Pré-condição: Ator está autenticado
Pós-condição: Piloto fica disponível para jogar

Fluxo normal:
1. Ator introduz nome, CTS, SVA.
2. Sistema regista o piloto

Use case: Alterar Piloto

Descrição: O administrador altera um Piloto
Cenários: Cenário 4
Pré-condição: Piloto já existe
Pós-condição: Piloto fica alterado

Fluxo normal:
1. Ator introduz nome a alterar
2. Sistema guarda alterações do piloto

Fluxo Alternativo(1) (altera CTS)
1.1. Ator introduz um novo CTS
1.2 Volta para o passo 2

Fluxo Alternativo(2) (altera SVA)
2.1 Ator introduz um novo SVA
2.2 Volta para o passo 2

Use case: Eliminar Piloto

Descrição: O administrador elimina um Piloto

Cenários: Cenário 4

Pré-condição: Piloto existe

Pós-condição: Piloto já não existe

Fluxo normal:

1.Ator escolhe nome a alterar

2.Sistema elimina o Piloto

Fluxo Exceção(1)(Nome não existe) (2)

2.1 Sistema verifica que nome não existe

2.2 Sistema termina o processo

Campeonato

Use Case: Adicionar Campeonato

Descrição: O administrador regista um novo campeonato

Cenários: Cenário 1

Pré-condição: Ator está autenticado

Pós-condição: Foi adicionado um campeonato à lista de campeonatos

Fluxo normal:

1. Ator escolhe um nome para o campeonato
2. Sistema regista o campeonato

Fluxo Exceção(1)(Nome já existe) (passo 1)

1.1 Sistema verifica que nome do campeonato já existe

1.2 Sistema termina o processo

Use Case: Eliminar Campeonato

Descrição: O administrador remove um campeonato já existente

Cenários: Cenário 1

Pré-condição: O campeonato têm de existir
Pós-condição: O campeonato deixa de existir

Fluxo normal:

1. Ator escolhe o nome do campeonato que deseja excluir
2. Sistema elimina o campeonato escolhido

Fluxo Exceção(1)(Nome não existe) (passo 1)

2.1 Sistema verifica que nome não existe
2.2 Sistema termina o processo

Use Case: Alterar Campeonato

Descrição: O administrador altera um campeonato já existente
Cenários: Cenário 1

Pré-condição: O campeonato tem de existir
Pós-condição: O campeonato é alterado

Fluxo normal:

1. Ator escolhe o nome do campeonato que deseja alterar
2. Sistema altera o campeonato selecionado

Fluxo Exceção(1)(Nome não existe) (2)

2.1 Sistema verifica que nome não existe
2.2 Sistema termina o processo

Carro

Use Case: Adicionar Carro

Descrição: O administrador regista um novo carro
Cenários: Cenário 3
Pré Condição : Autor está autenticado

Pós Condição : O sistema fica com mais um carro disponível para jogar

Fluxo Normal:

1. Sistema apresenta categorias disponíveis
2. Actor escolhe classe, marca, modelo, cilindrada e potência
3. Sistema verifica que o carro é da Classe C1 (e que pode ser híbrido)
4. Sistema atribui fiabilidade aleatoriamente ao carro
5. Ator indica que o carro é não híbrido
6. Ator indica a afinação do Carro
7. Actor indica PAC
8. Sistema regista carro

Fluxo Alternativo (1) [carro é SC] (passo 2)

3.1 Sistema verifica que carro é da Classe SC

3.2 Sistema atribui fiabilidade aleatoriamente ao carro

3.3 Actor indica PAC

3.4 Regressa a 8

Fluxo Alternativo (2) [carro é C2] (passo 3)

3.1 Sistema verifica que carro é da Classe C2

3.2 Sistema atribui fiabilidade aleatoriamente ao carro

3.3 Ator indica que o carro é não híbrido

3.4 Ator indica a afinação do Carro

3.5 Ator indica PAC

3.6 Regressa a 8

Fluxo Alternativo (3) [carro é GT] (passo 3)

3.1 Sistema verifica que carro é da Classe GT

3.2 Sistema atribui fiabilidade aleatoriamente ao carro

3.3 Ator indica que o carro é não híbrido

3.4 Ator indica PAC

3.5 Regressa a 8

Fluxo Alternativo (4) [carro é Híbrido] (passo 5)

5.1 Actor indica que é híbrido e indica potência do motor elétrico

5.2 Ator indica a afinação do Carro

5.3 Actor indica PAC

5.4 Regressa a 8

Use case: Remover Carro

Descrição: O administrador remove um carro

Cenários: Cenário 3

Pré-condição: Ator está autenticado

Pós-condição: Carro deixa de estar disponível para jogar

Fluxo normal:

1. Ator escolhe marca e modelo

2. Sistema apaga carro

Fluxo de exceção (Carro não existe) (passo 1)

1.1 Sistema termina processo

Use case: Alterar Carro

Descrição: O administrador altera um carro

Cenários: Cenário 3

Pré-condição: Ator está autenticado

Pós-condição: Carro fica com características alteradas

Fluxo normal:

1. Ator escolhe marca e modelo
2. Sistema apresenta lista de características que podem ser mudadas
3. Ator seleciona a desejada
4. Ator escolhe novos valores para as características selecionadas
5. Sistema verifica se valor introduzido está dentro dos possíveis
6. Sistema atualiza características do carro

Fluxo alternativo(1) Sistema verifica que valor inserido não é permitido (passo 5)

5.1 Sistema pede que ator insira outro valor

5.2 Ator insere valor novo

5.3 Sistema verifica que valor introduzido está dentro dos possíveis

5.3 Regressa ao passo 6

Circuito

Use case: Adicionar circuito

Descrição: O administrador regista um novo circuito

Cenários: Cenário 2

Pré-condição: Administrador está autenticado

Pós-condição: Circuito fica disponível para ser escolhido

Fluxo normal:

1. Administrador cria um circuito e dá-lhe um nome

2. Administrador indica o comprimento do circuito, número de curvas e chicanes

3. Sistema com essa informação calcula o número de retas

4. Sistema apresenta a lista de curvas, rectas e GDUs disponíveis (impossível, possível e difícil)

5. Administrador escolhe um GDU para cada uma das curvas e rectas

6. Sistema define GDU da chicane como difícil

7. Administrador indica o número de voltas e regista o circuito

8. Sistema adiciona o circuito à lista de circuitos disponíveis

Fluxo de Exceção (1) (Circuito já existente)

2.1 Sistema verifica que um circuito com o mesmo nome já existe

2.2 Sistema termina o processo

Use case: Remover circuito

Descrição: O administrador remove um circuito já existente

Cenários: Cenário 2

Pré-condição: Administrador está autenticado e o circuito já existe
Pós-condição: Circuito não está na lista de circuitos disponíveis

Fluxo normal:
1. Administrador escolhe o nome do circuito que quer eliminar
2. Sistema remove o circuito da lista de circuitos disponíveis

Fluxo de Exceção (1) (Circuito não existente)
2.1 Sistema verifica que o circuito a ser retirado não existe
2.2 Sistema termina o processo

Use case: Alterar circuito

Descrição: O administrador altera um circuito já existente
Cenários: Cenário 2
Pré-condição: Administrador está autenticado e o circuito já existe
Pós-condição: Os atributos do circuito são alterados

Fluxo normal:
1. Administrador escolhe o nome do circuito que quer alterar
2. Administrador insere os novos dados
3. Sistema grava o circuito com as alterações

Fluxo de Exceção (1) (Circuito não existente)
2.1 Sistema verifica que o circuito a ser retirado não existe
2.2 Sistema termina o processo

3.2.2 Jogador

Use case: Escolher Campeonato

Descrição: O jogador escolhe Campeonato, um Carro e um Piloto

Cenários: Cenário 5

Pré-condição: O jogador está autenticado

Pós-condição: Jogador está pronto para a corrida

Fluxo normal:

1. Jogador escolhe um campeonato da lista de campeonatos existentes
2. O jogador escolhe um piloto e um carro
3. Jogador pode alterar a afinação do carro
4. Jogador escolhe o tipo de pneus (macio, duro ou chuva) e o modo do motor (conservador, normal ou agressivo)
5. Sistema regista as opções escolhidas

Fluxo Alternativo (1) [Caso os carros sejam Classe GT ou SC] (passo 3)

5.1 O Jogador não altera a afinação do carro

5.2 Regressa a 4

Use case: Simulação

Descrição: Racing Manager simula o Campeonato

Cenários: Cenário 5

Pré-condição: Jogador está pronto

Pós-condição: Campeonato é simulado

Fluxo normal:

1. Sistema analisa as características do campeonato, do circuito, dos carros e dos pilotos
2. Jogador inicia a corrida
3. Sistema escolhe aleatoriamente a situação meteorológica
4. Sistema simula a corrida
5. Sistema indica acontecimentos (avarias, ultrapassagens e avarias) por volta

6. Sistema verifica acontecimentos que foram identificados através de um Sítio, um Sujeito e uma Descrição

7. Sistema atualiza e indica a ordem dos carros no final de cada volta

8. Sistema verifica se a corrida acabou

9. Sistema regista a pontuação da corrida nas informações

10. Regressa a 3

Fluxo Alternativo (1) [Sistema não indica acontecimentos] (passo 5)

4.1 Durante a volta, não foram registados acontecimentos

4.2 Regressa a 6

Fluxo Alternativo (2) [Já foram realizadas todas as corridas] (passo 8)

3.1 Sistema regista a pontuação da Corrida nas informações

3.2 Sistema apresenta os resultados finais

3.3 Sistema termina o processo

Use case: Consultar Resultados

Descrição: O jogador consulta os resultados de um campeonato

Cenários: Cenário 5

Pré-condição: Jogador está autenticado

Pós-condição: True

Fluxo normal:

1. O Jogador seleciona o campeonato

2. O sistema apresenta os resultados

Fluxo de Exceção (1) [caso o campeonato não tenha iniciado] (passo 1)

1.1 O Sistema verifica que o campeonato ainda não iniciou

1.2 O Jogador escolhe outro campeonato, se existir

4. Diagrama de Use Case

Com base nos atores e nos use cases identificados procedemos à realização do diagrama de use cases. O Administrador do Sistema tem como tarefa a gerência dos pilotos, dos campeonatos, dos circuitos e dos carros, podendo adicionar, alterar e remover cada uma destas entidades. Ao jogador compete-lhe escolher o campeonato que quer jogar, assim como o carro e o piloto que quer utilizar para as suas corridas. Pode ainda iniciar a simulação das corridas, consultar as informações (voltas, posição relativa de cada carro e descrição dos acontecimentos), escolher o tipo de pneus (macio, duro ou chuva) e o modo do motor (conservador, normal ou agressivo) e definir a afinação do carro. Por último tem a capacidade de consultar os resultados finais do campeonato.

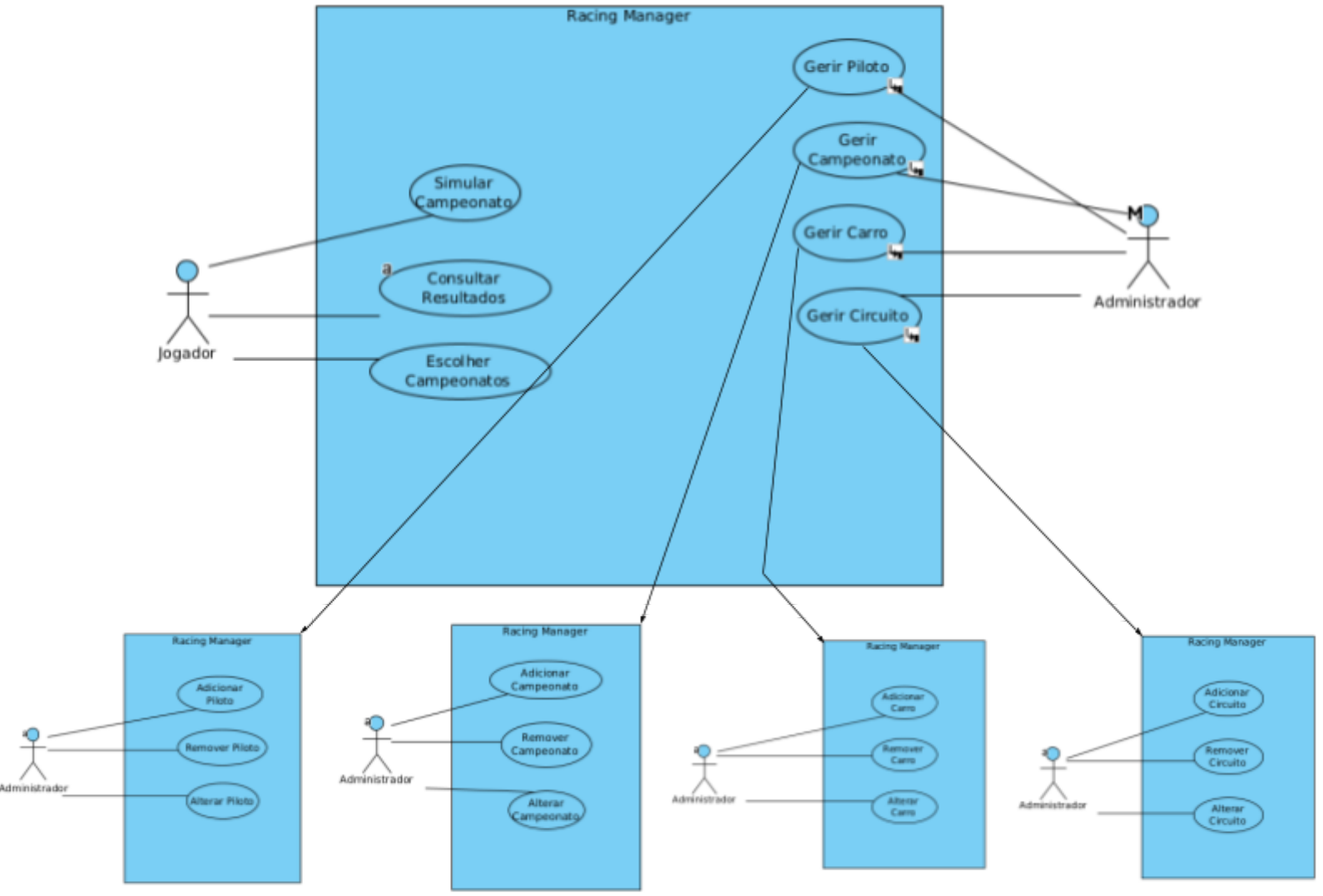


Fig.4 Diagrama de Use Cases

5. Conclusão

Nesta fase inicial do trabalho prático de Desenvolvimento de Sistemas de Software tratamos pela primeira vez da realização de diagramas de domínio e diagramas de use cases. Após esta fase do trabalho, além de ganharmos experiência a modelar, estamos preparados para fazer a implementação do sistema de uma forma mais estruturada e correta. O modelo de domínio ajuda a esquematizar uma possível organização das classes e do código da fase final do projeto. Dá-nos ainda uma visão geral sobre as relações entre as entidades, e a sua multiplicidade, o que nos permite facilmente compreender o problema. Já os use cases e o respectivo diagrama de use cases dão-nos discernimento sobre as funcionalidades da simulação que vamos implementar. Permite, além disso, identificar como os atores vão interagir com o sistema, assim como as modificações que estes vão realizar. Acreditamos ter criado um modelo firme e bem estruturado para assim a próxima fase do projeto poder ser viável.