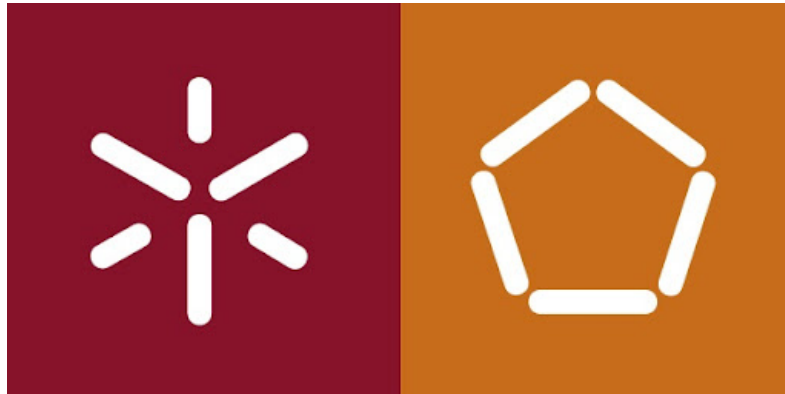


# Universidade do Minho

Departamento de Informática

Curso : MIEI / LEI



## Guião 1 - Relatório de DSS

Grupo nº2

Alexandre Eduardo Vieira Martins A93242

Guilherme Rodrigues do Outeiro Cunha Marques A94984

Hugo dos Santos Martins A95125

José Eduardo da Cunha Rocha A97270

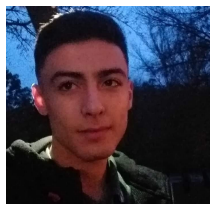
João Bernardo Teixeira Escudeiro A96075

<https://github.com/Eduard0Rocha/ProjetoDSSGrupo2>

19 de outubro de 2022



Hugo Martins



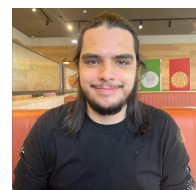
José Rocha



João Escudeiro



Alexandre Martins



Guilherme Marques

# Índice

<b>Introdução</b>	<b>2</b>
<b>Objetivos</b>	<b>2</b>
<b>Modelação de domínio</b>	<b>3</b>
Campeonato	3
Piloto	3
Carro	3
Circuito	4
Jogador e administrador	4
<b>Descrição dos Use Cases</b>	<b>5</b>
<b>Análise crítica dos resultados obtidos</b>	<b>6</b>
<b>Conclusão</b>	<b>6</b>
<b>Anexos</b>	<b>7</b>

# Introdução

Este trabalho tem como principal inspiração o F1 Manager, tendo então como principal objetivo, a simulação de corridas. O desafio proposto aos jogadores é, então, a escolha de uma combinação (configuração de carro e combinação com o piloto) que se adeque o melhor possível à pista e as condições climáticas. Quanto melhor a combinação feita, maiores são as hipóteses de ganhar. Vamos também possibilitar a criação de campeonatos, circuitos, pilotos e carros (baseados nas combinações possíveis).

Tendo isto em conta vamos realizar o projeto em várias fases. A presente fase do projeto tem como foco principal a realização do diagrama correspondente à modelação de domínio, assim como do diagrama de *Use Cases* correspondente aos vários cenários encontrados.

## Objetivos

Nesta primeira fase, e sendo esta a mais importante pois vai servir de alicerce para todo o trabalho ao longo do semestre, requer uma atenção especial.

Caminhando paralelamente com as aulas teóricas e práticas, neste início de projeto, deu-se particular relevância à conceção do modelo de domínio com as respectivas entidades relevantes e ao modelo de *Use Case* que contenha os requisitos e funcionalidades propostas por nós para o sistema.

Logo os dois maiores objetivos nesta fase são a modelação do modelo de domínio e o modelo dos *Use Case*.

# Modelação de domínio

## Campeonato

Uma das entidades deste projeto é o campeonato. Um campeonato será a entidade que ultimamente fará possível realizar a simulação de corridas, visto que a única maneira de se efetuar a simulação é através de um campeonato.

Um campeonato será criado por um utilizador após este fazer o login como administrador, este dar-lhe-á um nome e associará os circuitos pretendidos.

As entidades primárias que estão ligadas ao campeonato são:

- Nome do campeonato;
- Lista de Campeonatos;
- Corrida;

## Piloto

O piloto é das entidades mais simples mas vai definir como é que o carro vai ser usado, isto porque o estilo de condução do piloto vai em grande parte definir o seu desempenho na pista, o seu SVA vai delinear a frequência de tentativas de ultrapassagem e os riscos que está disposto a tomar. A CTS vai ser uma forma do jogador saber que performance esperar dado um determinado estado meteorológico. Entidade primárias ligadas ao piloto:

- Nome do piloto
- SVA (Segurança vs. Agressividade)
- CTS (Chuva vs. Tempo seco)

## Carro

Uma das entidades com maior relevo neste projeto é o carro. Os carros estão divididos por classes (C1, C2, GT e SC) de acordo com a sua cilindrada. Diretamente do texto é possível retirar a informação de que há uma entidade que se relaciona apenas com os carros das classes C1, C2 e GT que é o motor elétrico (presença ou ausência) que tem associada uma potência. Há também uma entidade que se relaciona apenas com os carros da categoria C2 que é a afinação mecânica do carro e que se refere à *downforce*.

Em termos de entidades comuns a todos os carros, retiramos diretamente do texto as seguintes:

- Potência do motor de combustão;
- Cilindrada;
- PAC ;
- Estabilidade em curva;
- Tipo de pneus;
- Fiabilidade;
- Marca;
- Modelo;

- Modo de funcionamento do motor.

Através de uma análise dos textos, podemos também deduzir outras entidades associadas que são relevantes na modelação do conhecimento sobre o carro:

- Distância percorrida;
- Integridade do carro;
- Estado dos pneus;
- Tempo de corrida;
- Velocidade.
- Tempo de Corrida

Algumas destas entidades podem, numa fase de implementação, ser atributos dos jogadores, porém, consideramos que, em termos abstratos de representação do conhecimento, são entidades relacionadas ao carro.

## ***Circuito***

O circuito também é uma entidade com bastante relevância neste projeto , pois sem eles não faria sentido falar em corridas nem mencionar as outras entidades como o carro , o piloto, etc.

Quanto aos circuitos, todas as entidades são comuns, pelo que as seguintes são as que conseguimos retirar após uma leitura detalhada do enunciado:

- Nome do circuito;
- Distância;
- N° de curvas, chicanes, retas;
- Lista de curvas, chicanes, retas;
- Cada curva, chicane, reta tem um GDU.

Também consideramos relevantes os seguintes:

- Estado climatérico;
- Localização
- N° de voltas.

## ***Jogador e administrador***

Por fim, também temos as entidades que utilizam e manipulam diretamente o programa, jogador e administrador, pois serão elas as responsáveis pelo uso e administração de todo o sistema.

O administrador tem permissão para adicionar/remover/criar pilotos, campeonatos, circuitos e carros e o jogador pode efetuar as escolhas de piloto, carro e algumas afinações mediante o circuito, pode também selecionar o campeonato e por fim jogá-lo.

As entidades associadas ao jogador são:

- Nome;
- Pontos no campeonato;
- Posição no campeonato;
- Posição na corrida.

Um utilizador pode ou não fazer *login* como administrador ou jogador e estes diferentes papéis dar-lhe-ão diferentes permissões dentro do jogo.

Nos anexos , está apresentado o modelo de domínio , que apesar de não muito perceptível , surge um ficheiro com a extensão .vpp com o nome :”Etapa1-Grupo2 “ , que contém o diagrama.

## Descrição dos *Use Cases*

Nos cenários 1, 2, 3 e 4, são relatadas todas as ações que o administrador pode efetuar no sistema e que são:

- Efetuar *login*;
- Adicionar campeonatos;
- Adicionar circuitos;
- Adicionar carros;
- Adicionar pilotos.

Através da informação presente no texto e de antecipação de possíveis decisões que o administrador pode querer tomar bem como possíveis equívocos que este possa apresentar (nomeadamente em relação aos intervalos de valores da cilindrada e atributos de carros e pilotos cujos valores se encontrem entre 0 e 1) criamos os fluxos alternativos e de exceção.

No cenário 5 são relatados 4 *Use Cases* diferentes:

- Configurar campeonato;
- Configurar corridas;
- Simular corridas;
- Resultado final.

Os primeiros dois estão relacionados com ações que o jogador terá de realizar. Este pode ou não estar autenticado, o que apenas interfere na atribuição (ou falta de atribuição) de pontos de *ranking* global.

No *Use Case* “Configurar campeonato” após o fluxo normal o jogador terá realizado a configuração de um campeonato com sucesso. Porém existe um fluxo alternativo que acontece no caso de o jogador não gostar dos circuitos do campeonato selecionado.

No *Use Case* seguinte após o fluxo normal o jogador terá configurado a corrida com sucesso sendo ambos os fluxos alternativos relacionados com a afinação do carro ( o jogador escolhe não afinar o carro ou o carro selecionado não pode ser afinado).

O último *Use Case* é referente ao “Resultado Final” este será ultimamente executado pelo sistema após ocorrerem todas as simulações. O fluxo de exceção é referente ao facto de o jogador não fazer login o que fará com que a sua pontuação na corrida não seja contabilizada no ranking global. Por fim, o fluxo alternativo acontece caso o jogador não tenha pontos suficientes para subir de posição no ranking global.

# Análise crítica dos resultados obtidos

Através da contribuição de todos os elementos do grupo, os objetivos principais desta fase do projeto foram alcançados na íntegra.

Consideramos que a modelação de domínio engloba todas as entidades relevantes encontradas no enunciado e outras que foram deduzidas, que são importantes para uma mais completa modelação do conhecimento sobre as entidades.

Após discutir bastante sobre este assunto, consideramos que a versão “*premium*” do sistema será algo a ter em consideração apenas na próxima fase (realização da arquitetura conceptual), pois é uma requisito do sistema e não uma “entidade”, nem um atributo propriamente dito. Assim esta parte não consta no modelo de domínio, apenas é abordado de uma forma ligeira no diagrama de *Use Cases*.

Relativamente ao segundo tópico - os *Uses Cases* - pensamos ter cobrido todos os casos de uso possíveis do futuro sistema, bem como os casos alternativos e de exceção.

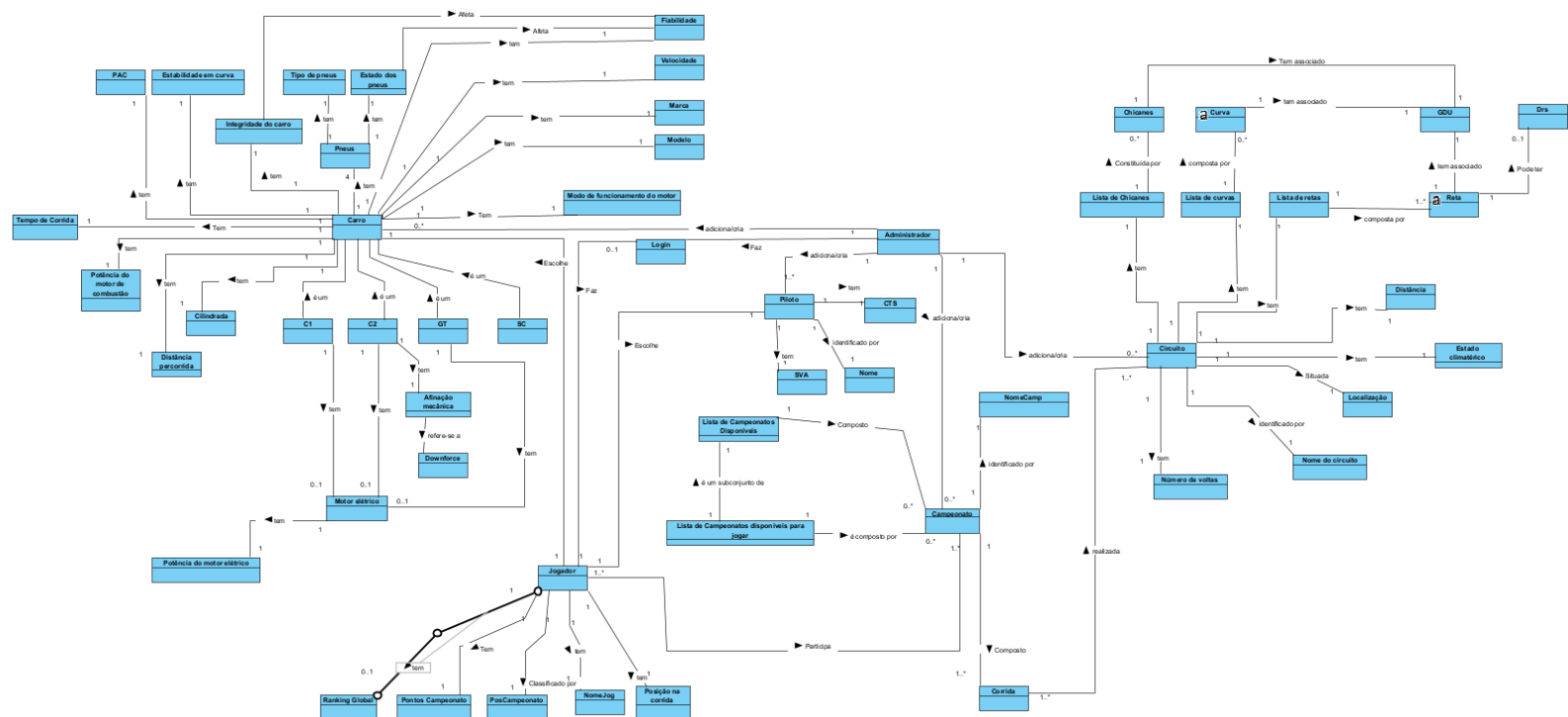
De um modo geral, e com base neste relatório, em que constam os anexos respectivos ao diagrama de entidades, bem como o diagrama de use cases, conseguimos suportar os textos de descrição que servem para “explicar” de uma forma sucinta o que foi pensado e a forma como foi realizada cada uma das partes.

## Conclusão

A principal dificuldade encontrada pelo grupo foi a construção do diagrama correspondente à modelação de domínio, uma vez que esta foi dividida pelo grupo de acordo com os cenários, o que dificultou a união de todos os componentes.

Apesar do pequeno obstáculo, acreditamos ter alcançado os resultados esperados, construindo uma boa base para as fases posteriores deste projeto.

## Anexos



**Figura 1.** Modelo de domínio (Apesar de não ser muito perceptível o documento vpp segue anexado).



## **Use Case 0 : "Autenticar"**

Descrição : Administrador ou Jogador (premium ou não) faz login no sistema

Cenário 0

Pré Condição : True ;

Pós Condição : O utilizador está autenticado;

Fluxo normal :

1. O utilizador insere que é um utilizador já existente no sistema
2. O utilizador seleciona se quer fazer login como jogador ou administrador;
3. O utilizador insere as credenciais .
4. Sistema verifica que as credenciais são válidas;
5. O utilizador está autenticado no sistema como administrador/jogador;

Fluxo Alternativo 1- [Utilizador insere que é um jogador novo (passo 1)]

- 1.1. Sistema pede credenciais novas
- 1.2. Utilizador insere as credenciais
- 1.3. Sistema pede credenciais novamente
- 1.4. Utilizador insere as credenciais novamente
- 1.5. Sistema verifica que as credenciais são iguais
- 1.5. Sistema regista um novo utilizador
- 1.6. Regressa a 2

Fluxo Alternativo 2- [Utilizador insere credenciais diferentes (passo 1.5)]

- 1.5.1. Sistema informa que as credenciais são diferentes
- 1.5.2. Regressa a 1.1

Fluxo Exceção 3[[Utilizador cancela operação após ter inserido credenciais diferentes] (passo 1.5.1)]

- 1.5.1.1 Utilizador decide cancelar operação
- 1.5.1.2 Sistema Cancela operação

Fluxo Exceção 4[[Utilizador cancela operação após o sistema não validar as credenciais] (passo 4)]

- 4.1 Utilizador decide cancelar operação
- 4.2 Sistema Cancela operação

## **Use Case 1 : "Criação de campeonatos"**

Pré-Condição: O utilizador está autenticado como administrador

Pós-Condição: O sistema fica com um novo campeonato jogável

Fluxo Normal:

1. Utilizador dá nome ao campeonato
2. Sistema verifica que o nome do campeonato é válido
3. Utilizador escolhe os circuitos que vão pertencer ao campeonato
4. Sistema verifica se os circuitos são válidos
5. Sistema cria campeonato

Fluxo Alternativo 1 [nome de campeonato inválido] (passo 2):

- 2.1. Sistema informa do ocorrido
- 2.2. Administrador indica um novo nome
- 2.3. Sistema verifica que o nome é válido
- 2.4. Regressa a 3

Fluxo de Exceção 2 [Utilizador cancela após atribuir um nome errado] (passo 2):

- 2.1.1. Utilizador cancela criação de campeonato
- 2.1.2. Sistema abandona criação do campeonato

Fluxo Alternativo 3 [Circuitos adicionados são inválidos] (passo 4):

- 4.1. Sistema informa utilizador dos circuitos inválidos
- 4.2. Utilizador escolhe novos circuitos
- 4.3. Sistema verifica que novos circuitos são válidos
- 4.4. Regressa a 5

Fluxo de Exceção 4 [Utilizador cancela após introduzir circuitos inválidos] (passo 4)

- 4.2.1. Utilizador cancela criação do campeonato
- 4.2.2. Sistema cancela a operação

## Use Case 2 : "Adicionar Circuito"

Descrição : Administrador adiciona um circuito

Cenário 2

Pré Condição : O utilizador está autenticado como administrador;

Pós Condição : O circuito fica disponível para ser utilizado em Jogos

Fluxo normal:

1. Administrador adiciona o nome do circuito
2. Sistema verifica que o nome do novo circuito não existe no programa
3. Administrador adiciona a distância da pista, nº de curvas e nº de chicanes
4. Sistema verifica que a distância da pista, nº de curvas e nº de chicanes são válidos
5. Sistema informa qual o nº de retas e a lista das curvas
6. Administrador escolhe o GDU de cada curva
7. Sistema verifica que os GDU são válidos
8. Sistema indica o nº de voltas do circuito
9. Sistema regista o novo circuito

Fluxo Alternativo 1 [Nome já existe]

- 2.1 Sistema informa que o nome já existe
- 2.2 Sistema pede a informação novamente
- 2.3 Administrador indica a informação pedida
- 2.4 Sistema verifica que o nome é válido
- 2.5 Regressa a 3

Fluxo Exceção 2 [administrador cancela operação após o sistema não validar o nome]

- 2.3.1 Administrador escolhe cancelar a operação
- 2.3.2 Sistema cancela a operação

Fluxo Alternativo 3 [ distância da pista, nº de curvas e nº de chicanes não válido]

- 4.1 Sistema informa que a informação indicada não é válida
- 4.2 Sistema pede a informação novamente
- 4.3 Administrador indica a informação pedida
- 4.4 Sistema verifica se é válida
- 4.5 Regressa a 5

Fluxo Exceção 4 [administrador cancela operação após o sistema não validar a distância da pista, nº de curvas e nº de chicanes]

- 4.3.1 Administrador escolhe cancelar a operação
- 4.3.2 Sistema cancela a operação

Fluxo Alternativo 5 [GDU não é válido]

- 7.1 Sistema informa que o GDU indicado não é válido
- 7.2 Sistema pede a informação novamente
- 7.3 Administrador indica a informação pedida
- 7.4 Sistema verifica que o GDU é válido
- 7.5 Regressa a 8

Fluxo Exceção 6 [administrador cancela operação após o sistema não validar o GDU]

- 7.3.1 Administrador escolhe cancelar a operação
- 7.3.2 Sistema cancela a operação

### **Use Case 3 : "Adicionar um carro"**

Descrição: Administrador adiciona um carro

Cenário 3

Pré-condição: O utilizador está autenticado como administrador;

Pós-condição: O carro fica disponível para ser utilizado em jogos

Fluxo normal:

1. Administrador consulta as categorias disponíveis
2. Administrador escolhe a categoria
3. Administrador indica a marca, modelo, cilindrada e potência do motor de combustão
4. Sistema verifica que a cilindrada é válida considerando a categoria
5. Administrador adiciona motor elétrico
6. Administrador indica potência do motor elétrico
7. Administrador indica o PAC do carro
8. Sistema verifica que o PAC é válido
9. Sistema guarda a informação do carro

Fluxo Alternativo 1 [cilindrada não é válida] (passo 4)

- 4.1 Sistema informa que a cilindrada indicada não é válida
- 4.2 Sistema pede a informação novamente
- 4.3 Administrador indica a informação pedida
- 4.4 Sistema verifica que a cilindrada é válida considerando a categoria
- 4.5 Regressa a 5

Fluxo Exceção 2 [administrador cancela operação após o sistema não validar a cilindrada] (passo 4.3)

- 4.3.1 Administrador escolhe cancelar a operação
- 4.3.2 Sistema cancela a operação

Fluxo Alternativo 3 [administrador não adiciona motor elétrico] (passo 5)

- 5.1 Administrador não adiciona motor elétrico
- 5.2 Regressa a 7

Fluxo Alternativo 4 [PAC não é válido] (passo 8)

- 8.1 Sistema informa que o PAC indicado não é válido
- 8.2 Sistema pede a informação novamente
- 8.3 Administrador indica a informação pedida
- 8.4 Sistema verifica que o PAC é válido
- 8.5 Regressa a 9

Fluxo Exceção 5 [administrador cancela operação após o sistema não validar o PAC] (passo 8.3)

- 8.3.1 Administrador escolhe cancelar a operação
- 8.3.2 Sistema cancela a operação

#### **Use Case 4 : "Adicionar Pilotos"**

Descrição : Administrador adiciona um piloto

Cenário 4

Pré Condição : O utilizador está autenticado como administrador;

Pós Condição : O piloto fica disponível para ser utilizado em jogos

Fluxo normal:

1. Administrador adiciona o nome do piloto
2. Sistema verifica que o nome do piloto não existe no programa
3. Administrador adiciona o nível de perícia CTS
4. Sistema verifica que CTS é válido
5. Administrador adiciona o critério SVA
6. Sistema verifica que o SVA é válido
7. Sistema guarda a informação do novo piloto

Fluxo Alternativo 1 [Nome já existe ] (passo 2)

- 2.1 Sistema informa que o nome já existe
- 2.2 Sistema pede a informação novamente
- 2.3 Administrador indica a informação pedida
- 2.4 Sistema verifica que o nome é válido
- 2.5 Regressa a 5

Fluxo Exceção 2 [administrador cancela operação após o sistema não validar o nome ] (passo 2.3)

- 2.3.1 Administrador escolhe cancelar a operação
- 2.3.2 Sistema cancela a operação

Fluxo Alternativo 3 [CTS não é válido] (passo 4)

- 4.1 Sistema informa que o CTS indicado não é válido
- 4.2 Sistema pede a informação novamente
- 4.3 Administrador indica a informação pedida
- 4.4 Sistema verifica que o CTS é válido
- 4.5 Regressa a 5

Fluxo Exceção 4 [administrador cancela operação após o sistema não validar o CTS ] (passo 4.3)

- 4.3.1 Administrador escolhe cancelar a operação
- 4.3.2 Sistema cancela a operação

Fluxo Alternativo 5 [SVA não é válido] (passo 6)

- 6.1 Sistema informa que o SVA indicado não é válido
- 6.2 Sistema pede a informação novamente
- 6.3 Administrador indica a informação pedida
- 6.4 Sistema verifica que o SVA é válido
- 6.5 Regressa a 7

Fluxo Exceção 6 [administrador cancela operação após o sistema não validar o SVA ] (passo 6.3)

- 6.3.1 Administrador escolhe cancelar a operação
- 6.3.2 Sistema cancela a operação

## **Use Case 5 : "Configurar campeonato"**

Descrição: Jogador configura um campeonato para jogar

Cenário: 5

Pré-Condição: Existência de campeonatos, de carros e de pilotos.

Pós-Condição: Campeonato configurado pelo utilizador para jogar

Fluxo Normal:

1. Sistema apresenta a lista de campeonatos e pede ao jogador que escolha um
2. Jogador escolhe um campeonato da lista
3. Jogador analisa os circuitos que compõem o campeonato
4. Jogador escolhe um carro de uma dada categoria com base na análise realizada
5. Jogador analisa o desempenho dos pilotos
6. Jogador escolhe o piloto com base na análise
7. Outros utilizadores podem juntar-se ao campeonato configurado

Fluxo Alternativo 1 [Jogador não gosta das pistas no campeonato escolhido] (passo 3):

- 3.1 Jogador cancela o ação
- 3.2 Sistema mostra a lista de campeonatos ao jogador e pede que escolha outro
- 3.3 Jogador escolhe outro campeonato
- 3.4 Jogador analisa os circuitos que compõem o campeonato
- 3.5 Regressa a 4

## **Use Case 6 : "Configurar corridas"**

Descrição: Jogador configura as corridas

Cenário: 5

Pré-Condição: Configuração de campeonato concluída

Pós-Condição: O utilizador configurou as corridas para jogar

Fluxo Normal:

1. Sistema dá início ao campeonato
2. Sistema apresenta o circuito e as condições meteorológicas da primeira corrida
3. O jogador altera as afinações do carro com base nas condições
4. O jogador escolhe o tipo de pneus e o modo do motor de acordo com as condições

Fluxo Alternativo 1 [O Jogador decide não afinar o carro] (passo 3):

- 3.1 Regressa a 4

Fluxo Alternativo 2 [O carro escolhido não pode ser afinado] (passo 3):

- 3.1 Regressa a 4

### **Use Case 7 : " Apresentar Resultado Final"**

Descrição: Sistema apresenta o resultado final

Cenário: 5

Pré-Condição: Simulação de todas as corridas de um campeonato realizadas

Pós-Condição: O Sistema apresenta o resultado final

Fluxo Normal:

1. O sistema apresenta as classificações finais do campeonato
2. O jogador faz login no sistema
3. Com base na classificação do jogador o sistema adiciona os pontos respectivos à pontuação global
4. O jogador tem pontos suficientes para subir de posição no ranking global

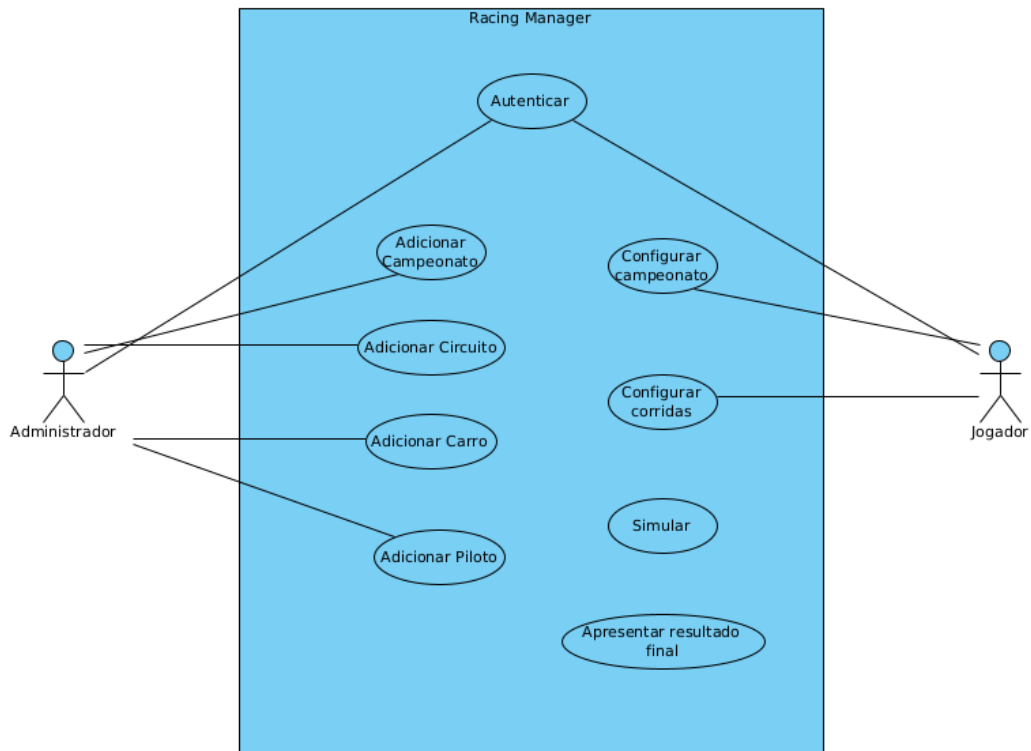
Fluxo Exceção 1 [O jogador não tem conta ou não faz login] (passo 2):

- 2.1 O sistema não contabiliza os seus pontos
- 2.2 O sistema cancela a operação

Fluxo Alternativo 2[O jogador não tem pontos suficientes para subir de posição no ranking global] (passo 4):

- 4.1 O jogador mantém o ranking global





**Figura 2.** Diagrama de *Use Cases*.

Estes anexos consistem na especificação dos vários Use Cases que surgiram após uma leitura detalhada do enunciado, bem como em o próprio diagrama que , de uma forma geral ajudam a perceber todas as funcionalidades propostas e pensadas para o sistema , bem como servem de base para as próximas fases de avaliação de requisitos e modelação conceptual.