



Universidade do Minho  
Escola de Engenharia  
Licenciatura em Engenharia Informática

Ano Letivo de 2002/2003  
**Repositório Github**  
<https://github.com/cgustavop/DSS-PL>

# Racing Manager

20 de outubro de 2022

# DSS



Vasco Manuel Araújo  
Andrade de Oliveira  
96361



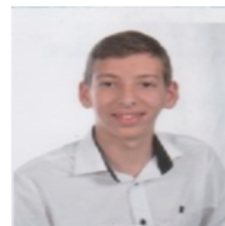
Carlos Gustavo  
Silva Pereira  
96867



Cláudio Alexandre  
Freitas Bessa  
97063



Carlos Emanuel  
Leite Machado  
97114



Tiago André  
Mendes Oliveira  
97254

# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
<b>2</b>	<b>Objetivos</b>	<b>2</b>
<b>3</b>	<b>Modelo de Domínio</b>	<b>3</b>
3.1	Entidades presentes no Modelo . . . . .	3
3.2	Leitura do Modelo de Domínios . . . . .	4
<b>4</b>	<b>Use case</b>	<b>5</b>
4.1	Diagrama de <i>use cases</i> . . . . .	6
4.2	Leitura do Diagrama de use cases . . . . .	7
4.3	Atores e <i>use cases</i> . . . . .	8
4.4	Especificações de use case . . . . .	9
<b>5</b>	<b>Conclusão</b>	<b>13</b>
	<b>Anexos</b>	<b>14</b>

# Lista de Figuras

3.1	Modelo de domínio . . . . .	3
4.1	Diagrama dos <i>use cases</i> . . . . .	6

# 1 Introdução

Racing Manager é um simulador de corridas local, onde um ou mais jogadores competem em campeonatos de forma simulada pelo sistema. Nesta aplicação existem três atores principais, os gestores, os administradores e por fim os jogadores. Os administradores são utilizadores escolhidos pelos gestor para adicionar *features* ao simulador. Dentro dessas *features* temos como exemplo, criação de campeonatos, adição de circuitos, adição de carros e adição de pilotos entre outros. Os jogadores, por sua vez, são capazes de competirem entre si localmente em campeonatos disponíveis para os mesmos. Possuem a opção de escolherem o seu carro, piloto e adaptarem-se, afinando os carros, consoante as condições dos circuitos geradas pelo sistema. Ao fim de cada corrida os jogadores presentes, são submetidos a um sistema de *ranking*, caso estejam autenticados, para um *ranking* final no fim do campeonato.

Tal como outros sistemas de *software* na indústria, possui duas versões do jogo, sendo uma delas a versão base e a versão *premium*. A sua grande diferença é possível ser visualizada durante a simulação da corrida, onde as posições relativas entre cada competidor e outras situações, ultrapassagens, despistes, entre outros, em vez de serem atualizadas ao fim de cada volta ou no fim de cada segmento, são atualizadas em tempo real.

**Área de Aplicação:** Desenho e arquitetura de sistemas de *software*.

**Palavras-Chave:** Base de Dados, Jogos, Aplicação, Simulação

## 2 Objetivos

Nesta fase introdutório do projeto *Racing Manager* é pretendido a execução de algumas tarefas cruciais para todo o sistema de *software*:

- Compreensão do solicitado no enunciado.
- Criação modelo de domínio com as entidades relevantes
- Criação do modelo de *use cases*, diagrama mais as especificações, com as funcionalidades propostas

## 3 Modelo de Domínio

O Modelo de Domínio descreve as entidades do contexto em que o sistema deve ser implementado. Um modelo como este é importante para estabelecer algumas regras sobre as entidades e pensar como estas vão funcionar no respetivo sistema.

Para a resolução do Modelo de Domínios começamos por identificar, através do enunciado apresentado, as entidades do sistema e de seguida as suas associações, com as suas respetivas multiplicidades. É de apontar que o desenvolvimento não foi linear e, como tal, foi sofrendo alterações ao longo do tempo.

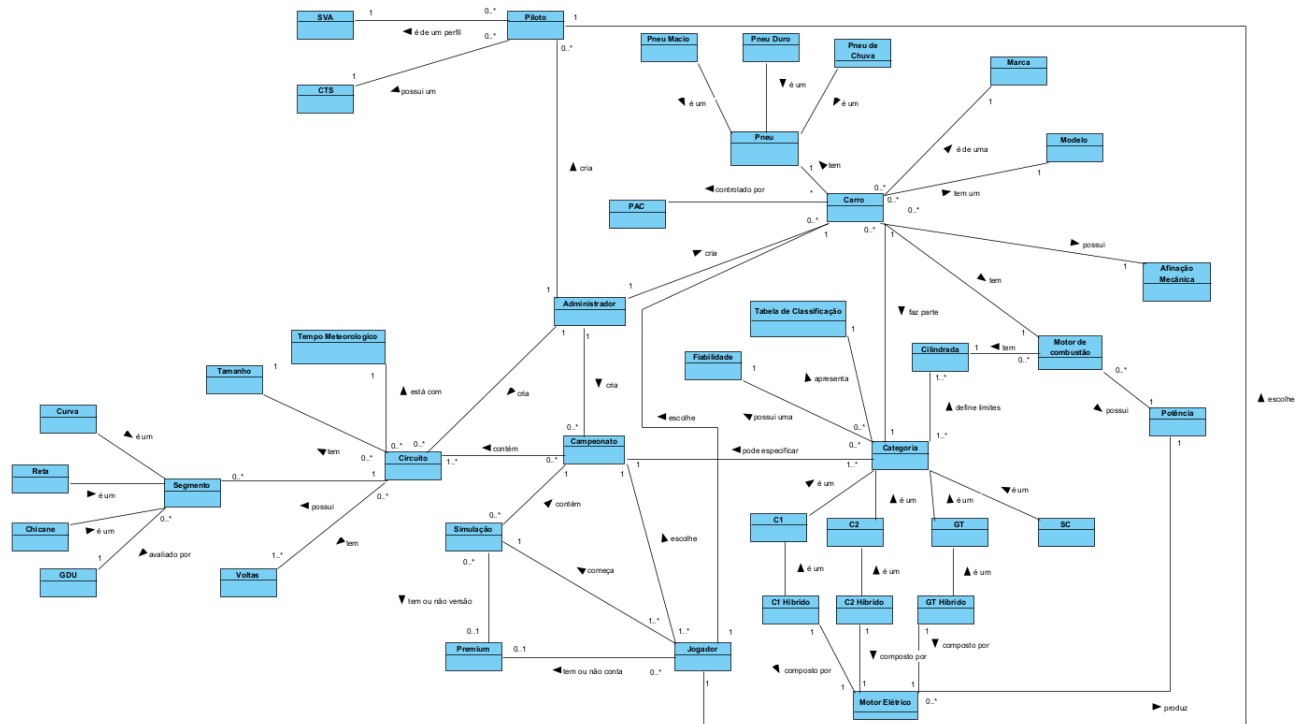


Figura 3.1: Modelo de domínio

### 3.1 Entidades presentes no Modelo

- Administrador, Piloto, SVA, CTS, Campeonato, Circuito, Voltas, Tempo Meteorológico, Tamanho, Voltas, Segmento(Curva, Reta, Chicane, GDU),

- Carro, Marca, Modelo, Cilindrada, Pneu(Macio, Duro, de Chuva), Motor de Combustão, Motor Elétrico, Potência, Fiabilidade, Ajuste Mecânica, Categoria(C1(Híbrido), C2(Híbrido), GT(Híbrido), SC),
- Jogador, Tabela de Classificação, Simulação, Premium.

## 3.2 Leitura do Modelo de Domínios

O **Administrador** pode criar:

- Um **Campeonato**, com a respetiva lista de **Circuitos** pretendidos;
- Um **Circuito**, com os seus componentes: **Voltas, Tempo Meteorológico, Tamanho, Voltas, Segmento (Curva, Reta, Chicane, GDU)**;
- Um **Carro**, com os seus componentes: **Marca, Modelo, Cilindrada, Motor de Combustão, Potência, Fiabilidade, PAC, Categoria(C1(Híbrido), C2(Híbrido), GT(Híbrido), SC)**, Motor Elétrico (só é preciso fornecer se o carro for híbrido);
- Um **Piloto**, com o seu **SVA** e **CTS**.

**Nota:** As entidades **Campeonato**, **Circuito** e **Piloto** precisam de um nome, portanto nós interpretamos que as entidades, neste caso, são o próprio nome.

O **Jogador**, para começar a jogar, precisa primeiro de escolher o **Campeonato** e de seguida o **Piloto** e o **Carro** pretendidos. Ainda antes de começar a jogar, o **Jogador** pode escolher mudar a **Ajuste Mecânica**, os **Pneus(Macio, Duro, de Chuva)** e o **PAC** do **Carro**. Depois dessa fase começa a **Simulação** do **Campeonato**, em que poderá ser **Premium** ou não, dependendo se o **Jogador** também o é. Ao fim de realização de cada **Circuito** o **Jogador** pode escolher mudar a **Ajuste Mecânica** do **Carro** outra vez. No fim da **Simulação** do **Campeonato**, os lugares de cada **Jogador** vão ser apresentados na **Tabela de classificação** da respetiva **Categoria** do **Carro**.

## 4 Use case

O modelo de Use Cases descreve as interações entre o sistema e o seu ambiente, fazendo parte deste os utilizadores e outros sistemas externos. Com este modelo podemos melhor guiar as funcionalidades requeridas do programa a desenvolver.

Na construção deste modelo analisámos os cenários no enunciado e identificamos os atores e as funcionalidades que estes requiriam.

**Nota:** Neste relatório não mostramos os cenários de *use cases* utilizados, mas eles estão presentes nas especificações dos use cases enviado em anexo.



## 4.1 Diagrama de use cases

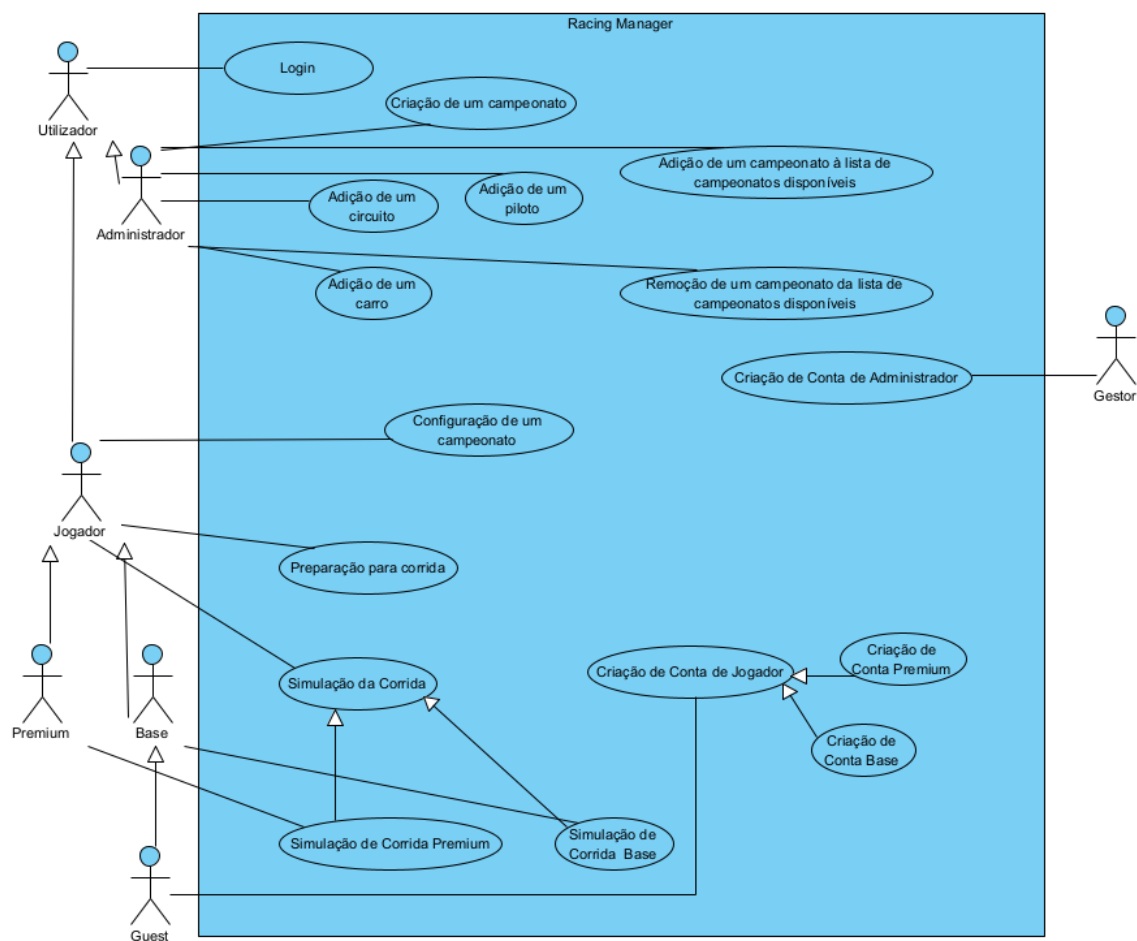


Figura 4.1: Diagrama dos *use cases*

## 4.2 Leitura do Diagrama de use cases

O ator **Utilizador** serve para mostrar que qualquer ator, tirando o **Gestor**, poderá efetuar o Login. Decidimos esta solução pois achamos que qualquer pessoa poderia querer trocar de conta a qualquer momento ou poderia ainda não ter efetuado o login assim que ligou o jogo.

O ator **Administrador** poderá criar campeonatos e adicionar circuitos, carros e pilotos, e também adicionar ou remover campeonatos na lista de campeonatos disponíveis, para o caso do **Administrador** achar que o campeonato deva ou não dar para jogar pelos **Jogadores**.

O ator **Jogador** poderá então configurar um campeonato (escolhendo o campeonato o carro e o piloto para poder começar a jogar), preparar-se para a corrida (mudando as afinações mecânicas, os pneus e o PAC do carro) e, depois de se preparar, começar a simulação da corrida, em que, dependendo se ele for um jogador **Base** ou **Premium**, terá direito à simulação de corrida base ou premium.

O ator **Guest** representa as pessoas que não têm conta ou que ainda não fizeram Login, podendo, assim, criar uma conta premium ou base, e tendo as mesmas funcionalidades que um **Jogador Base**.

Para se poder criar uma conta de administrador é preciso que o ator que tenha controlo dos registos das contas lhe adicione a conta, sendo esse ator o **Gestor**.

### 4.3 Atores e use cases

<b>Atores</b>	<b>Use Cases</b>
<b>Gestor</b>	Criação da conta de Administrador
<b>Utilizador</b>	Login
<b>Guest</b>	Criação da conta de Jogador Base
	Criação da conta de Jogador Premium
<b>Administrador</b>	Criação de um campeonato
	Adição de um campeonato à lista de campeonatos disponíveis
	Remoção de um campeonato da lista de campeonatos disponíveis
	Adição de um circuito
	Adição de um carro
	Adição de um piloto
<b>Jogador</b>	Configuração de um campeonato
	Preparação para a corrida
	Simulação da Corrida Base
	Simulação da Corrida Premium

Tabela 4.1: Atores e *use cases*

## 4.4 Especificações de use case

Use Case	Criação da conta de Administrador	
Ator	Gestor	
Pré-condição	True	
Pós-condição	Nova conta de Administrador adicionada ao sistema	
	<b>Ator</b>	<b>Sistema</b>
Fluxo Normal	1. Gestor fornece nome e palavra-passe	
		2. Sistema verifica disponibilidade do nome
		3. Sistema regista a conta como "administrador"
Fluxo alternativo (1) [Nome indisponível](passo 2)		1. Sistema informa que o nome não está disponível
		2. Regressa a 1

Tabela 4.2: Especificação da Criação conta de Administrador

Use Case	Login	
Ator	Guest (Administrador ou Jogador)	
Pré-condição	True	
Pós-condição	Utilizador autenticado (como "administrador" ou "jogador")	
	<b>Ator</b>	<b>Sistema</b>
Fluxo Normal	1. Utilizador fornece um nome e uma palavra passe	
		2. Sistema verifica validade dos dados
		3. Sistema verifica que o utilizador é um "jogador"
		4. Sistema autentica o login do utilizador como "jogador"
Fluxo alternativo (1) [Dados de login de administrador] (passo 3)		1. Sistema verifica que o utilizador é um "administrador"
		2. Sistema autentica o login do utilizador como "administrador"

Tabela 4.3: Especificação Login

Use Case	Criação da conta de Jogador Base	
Ator	Guest	
Pré-condição	True	
Pós-condição	Nova conta de Jogador Base adicionada ao jogo	
	<b>Ator</b>	<b>Sistema</b>
Fluxo Normal	1. Guest define um nome e uma palavra passe	
		2. Sistema verifica disponibilidade do nome
		3. Sistema regista a conta como "jogador base"
Fluxo alternativo (1) [Nome indisponível](passo 2)		1. Sistema informa que o nome não está disponível
		2. Regressa a 1

Tabela 4.4: Especificação Criação da conta de Jogador Base

Use Case	Criação da conta de Jogador Premium	
Ator	Guest	
Pré-condição	True	
Pós-condição	Nova conta de Jogador Premium adicionada ao jogo	
	<b>Ator</b>	<b>Sistema</b>
Fluxo Normal	1. Guest define um nome e uma palavra passe	
		2. Sistema verifica disponibilidade do nome
		3. Sistema regista a conta como "jogador premium"
Fluxo alternativo (1) [Nome indisponível](passo 2)		1. Sistema informa que o nome não está disponível
		2. Regressa a 1

Tabela 4.5: Especificação Criação da conta de Jogador Premium

<b>Use Case</b>	Criação de um campeonato	
<b>Ator</b>	Administrador	
<b>Pré-condição</b>	1. Administrador autenticado 2. Existência de circuitos	
<b>Pós-condição</b>	Novo campeonato adicionado ao jogo	
	<b>Ator</b>	<b>Sistema</b>
<b>Fluxo Normal</b>	1. Administrador fornece nome do campeonato	
		2. Sistema verifica disponibilidade do nome
		3. Sistema apresenta circuitos existentes
	4. Admin escolhe circuito(s) a adicionar	
		5. Sistema regista o campeonato
<b>Fluxo alternativo (1)</b> <b>[Nome indisponível](passo 2)</b>		1. Sistema informa que o nome não está disponível
		2. Regressa a 1

Tabela 4.6: Especificação da Criação de um campeonato

<b>Use Case</b>	Adição de um campeonato à lista de campeonatos disponíveis	
<b>Ator</b>	Administrador	
<b>Pré-condição</b>	1. Administrador autenticado 2. Existência de campeonatos indisponíveis	
<b>Pós-condição</b>	Novo campeonato disponível aos jogadores	
	<b>Ator</b>	<b>Sistema</b>
<b>Fluxo Normal</b>		1. Sistema apresenta campeonatos indisponíveis
	2. Administrador seleciona um campeonato a disponibilizar	
		3. Sistema atualiza campeonato como disponível

Tabela 4.7: Especificação da Adição de um campeonato à lista de campeonatos disponíveis

<b>Use Case</b>	Remoção de um campeonato da lista de campeonatos disponíveis	
<b>Ator</b>	Administrador	
<b>Pré-condição</b>	1. Administrador autenticado 2. Existência de campeonatos disponíveis	
<b>Pós-condição</b>	Novo campeonato indisponível aos jogadores	
	<b>Ator</b>	<b>Sistema</b>
<b>Fluxo Normal</b>		1. Sistema apresenta campeonatos disponíveis
	2. Administrador seleciona um campeonato a indisponibilizar	
		3. Sistema atualiza campeonato como indisponível

Tabela 4.8: Especificação da Remoção de um campeonato à lista de campeonatos disponíveis

<b>Use Case</b>	Adição de um circuito	
<b>Ator</b>	Administrador	
<b>Pré-condição</b>	Administrador autenticado	
<b>Pós-condição</b>	Novo circuito adicionado ao jogo	
	<b>Ator</b>	<b>Sistema</b>
<b>Fluxo Normal</b>	1. Administrador fornece nome, tamanho em km, nº de curvas, nº de chicanes e nº voltas	2. Sistema verifica disponibilidade do nome do circuito
		3. Sistema calcula nº de retas
		4. Sistema apresenta lista de curvas e retas (segmentos da pista)
	5. Administrador fornece GDU's para cada segmento	
		6. Sistema regista novo circuito
<b>Fluxo alternativo (1)</b> <b>[Nome indisponível](passo 2)</b>		1. Sistema informa que o nome não está disponível
		2. Regressa a 1

Tabela 4.9: Especificação da Adição de um circuito

Use Case	Adição de um carro	
Ator	Administrador	
Pré-condição	Administrador autenticado	
Pós-condição	Novo carro adicionado ao jogo	
	<b>Ator</b>	<b>Sistema</b>
Fluxo Normal	2. Administrador escolhe categoria, marca, modelo, cilindrada e potência	1. Sistema apresenta as categorias disponíveis
	4. Administrador indica que carro não é híbrido	3. Sistema verifica que o carro é C1 e necessita de fiabilidade (e pode ser híbrido)
	5. Administrador indica fiabilidade	
	7. Administrador indica PAC	6. Sistema verifica fiabilidade
		8. Sistema regista carro e este fica disponível para jogar
Fluxo alternativo (1) [Carro é SC](passo 3)		1. Sistema verifica que o carro é SC
Fluxo alternativo (2) [Carro é C2](passo 3)		2. Regressa a 5
Fluxo alternativo (3) [Carro é GT](passo 3)		1. Sistema verifica que o carro é C2
		2. Regressa a 4
		1. Sistema verifica que o carro é GT
		2. Regressa a 4
Fluxo alternativo (4) [Carro é C2 híbrido](passo 4)	1. Administrador indica que carro é híbrido	
	2. Administrador indica potência do motor elétrico	
		3. Regressa a 5
Fluxo alternativo (5)[Cilindrada não se enquadra na categoria](passo 2)	1. Sistema verifica que a cilindrada não se enquadra na categoria.	
		2. Regressa a 4

Tabela 4.10: Especificação da Adição de um carro

Use Case	Adição de um piloto	
Ator	Administrador	
Pré-condição	Administrador autenticado	
Pós-condição	Novo piloto adicionado ao jogo	
	<b>Ator</b>	<b>Sistema</b>
Fluxo Normal	1. Administrador indica o nome do piloto	2. Sistema verifica disponibilidade do nome do piloto
	3. Administrador indica os níveis de perícia nos critérios de CTS ("Chuva vs. Tempo Seco") e SVA ("Segurança vs Agressividade")	4. Sistema verifica que os dados de níveis de perícia estão válidos
		5. Sistema regista novo piloto
		1. Sistema informa que o nome não está disponível
Fluxo alternativo (1) [Nome indisponível](passo 2)		2. Regressa a 1
Fluxo alternativo (2) [Níveis de perícia inválidos](passo 4)		1. Sistema informa que os dados são inválidos
		2. Regressa a 3

Tabela 4.11: Especificação da Adição de um piloto

Use Case	Configuração de um campeonato	
Ator	Jogador	
Pré-condição	True	
Pós-condição	Jogador registado	
	<b>Ator</b>	<b>Sistema</b>
Fluxo Normal	1. Jogador seleciona campeonato	
	2. Jogador seleciona carro e piloto pretendido	
		3. Sistema verifica dados
		4. Sistema regista o jogador no campeonato
		5. Sistema regista novo piloto
Fluxo alternativo (1) [Adicionar outro jogador](passo 5)	1. Jogador escolhe adicionar outro jogador	
		2. Regressa a 2

Tabela 4.12: Especificação da Configuração de um campeonato

Use Case	Preparação para a corrida	
Ator	Jogador	
Pré-condição	Campeonato configurado	
Pós-condição	Jogador registado como pronto para corrida	
	<b>Ator</b>	<b>Sistema</b>
Fluxo Normal	1. Jogador escolhe fazer afinações ao seu carro	
	2. Jogador faz afinações	
	3. Jogador escolhe o tipo de pneu que pretende e se motor é híbrido ou não	
		4. Sistema regista jogador como pronto
Fluxo alternativo (1) [Não faz afinações](passo 1)	1. Jogador escolhe não fazer afinações ao seu carro	
		2. Regressa para 3
Fluxo alternativo (2) [Escolha de motor](passo 3):	1. Jogador não consegue escolher tipo de motor se o carro for de categoria SC	
		2. Regressa para 4

Tabela 4.13: Especificação da Preparação para a corrida

<b>Use Case</b>	Jogo simula a corrida	
<b>Ator</b>	Jogador	
<b>Pré-condição</b>	1. Corrida com todos os jogadores prontos 2. Jogador registrado no sistema como "jogador base" ou não ter efetuado login	
<b>Pós-condição</b>	Corrida simulada	
	<b>Ator</b>	<b>Sistema</b>
<b>Fluxo Normal</b>		1. Sistema inicia a simulação da corrida
		2. Após cada segmento (curva/reta/chicane) o sistema atualiza situações do mesmo
		3. Sistema indica posições após cada volta
		4. Sistema apresenta resultados no fim da corrida
		5. Sistema atribui pontos a cada jogador, por posições e categorias somando para corridas futuras no mesmo campeonato
		6. Sistema simula próximas corridas até serem feitas todas as existentes do campeonato
		7. Sistema verifica que jogador não está autenticado
	8. Jogador escolhe fazer login	
		9. Sistema contabiliza os pontos obtidos no ranking da sua conta
		10. Sistema mostra os rankings
<b>Fluxo alternativo (1)</b> <b>[Jogador já está autenticado](passo 7)</b>		1. Sistema verifica que jogador já está autenticado
<b>Fluxo alternativo (2)</b> <b>[Jogador não faz login](passo 8)</b>	1. Jogador não faz login	2. Regressa a 9
		2. Regressa a 10

Tabela 4.14: Especificação da Simulação da Corrida Base

<b>Use Case</b>	Simulação da Corrida Premium	
<b>Ator</b>	Jogador	
<b>Pré-condição</b>	1. Corrida com todos os jogadores prontos 2. Jogador registrado no sistema como "jogador premium"	
<b>Pós-condição</b>	Corrida simulada	
	<b>Ator</b>	<b>Sistema</b>
<b>Fluxo Normal</b>		1. Sistema inicia a simulação da corrida
		2. Sistema atualiza as situações do mesmo em tempo real
		3. Sistema indica posições em tempo real
		4. Sistema apresenta resultados no fim da corrida
		5. Sistema atribui pontos a cada jogador, por posições e categorias somando para corridas futuras no mesmo campeonato
		6. Sistema simula próximas corridas até serem feitas todas as existentes do campeonato
		7. Sistema contabiliza os pontos obtidos no ranking da sua conta
		8. Sistema mostra os rankings

Tabela 4.15: Especificação da Simulação da Corrida Premium

## 5 Conclusão

Concluindo a primeira fase deste trabalho dá-se então fim à segunda fase do ciclo de vida do desenvolvimento de sistemas, a análise.

Através da realização desta fase, com os respetivos Modelo de Domínio, Diagrama e Especificações de *Use Cases*, conseguimos ganhar uma melhor perceção deste trabalho.

Na realização do Modelo de Domínio identificamos as entidades do sistema e as suas relações, ajudando-nos, assim, a compreender melhor o domínio em questão.

Na realização do Diagrama de *Use Cases* identificamos os atores do sistema e as ações possíveis pelos mesmos, percebendo mais facilmente as funcionalidades que vamos implementar.

Na realização das especificações de *Use Cases* especificamos as ações do sistema, facilitando a compreensão das funcionalidades.

Esta fase também nos preparou para o começo da próxima fase, a concepção, onde vamos realizar uma arquitetura conceptual do sistema, capaz de suportar os requisitos identificados, e os modelos comportamentais necessários para descrever o comportamento pretendido para o sistema.



# Anexos

Tal como requerido nesta primeira entrega intermédia, em anexo encontram-se os respetivos diagramas, modelo de domínios e o diagrama de use cases e também as especificações do mesmo. Contudo em seguida, é possível observar uma visualização geral dos diagramas, que já foram especificados mais detalhadamente ao longo deste relatório.