

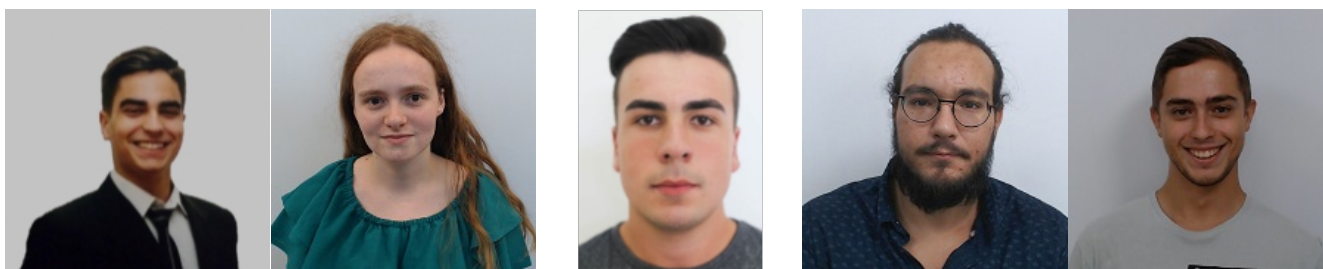


Universidade do Minho
Escola de Engenharia

Desenvolvimento de Sistemas de Software
Relatório de Projeto: FASE 1
2022/2023

Afonso Xavier Cardoso Marques 94940
Ana Filipa da Cunha Rebelo 90234
Délio Miguel Lopes Alves 94557
Pedro Miguel Duarte Araújo 70699
Ricardo Manuel Almeida Vieira De Castro 70492

20 de Outubro de 2022



Índice

| | | |
|----------|--|-----------|
| 1 | Introdução | 4 |
| 2 | Modelo de Domínio | 4 |
| 3 | Use Cases | 8 |
| 3.1 | Diagrama de Use Case | 8 |
| 4 | Especificação dos Use Cases | 9 |
| 4.1 | Use case: Criar conta de Administrador | 9 |
| 4.2 | Use case: Autenticar Administrador | 9 |
| 4.3 | Use case: Criar conta de Jogador | 10 |
| 4.4 | Use case: Autentica-se Jogador | 11 |
| 4.5 | Use case: Adicionar campeonato | 12 |
| 4.6 | Use case: Adicionar circuito | 12 |
| 4.7 | Use case: Adicionar carro | 13 |
| 4.8 | Use case: Adicionar piloto | 14 |
| 4.9 | Use case: Configurar campeonato | 14 |
| 4.10 | Use case: Configurar corridas | 15 |
| 4.11 | Use case: Simular corrida | 15 |
| 4.12 | Use case: Simular campeonato | 16 |
| 5 | Conclusão | 17 |

List of Figures

| | | |
|---|---|---|
| 1 | 1ª Versão do Modelo de Domínio | 5 |
| 2 | 2ª Versão do Modelo de Domínio | 6 |
| 3 | Versão final do Modelo de Domínio | 7 |
| 4 | Diagrama de Use Cases | 8 |

1 Introdução

O presente documento visa apresentar a primeira fase de desenvolvimentos do projeto proposto na Unidade Curricular de Desenvolvimento de Sistemas de Software.

O objetivo do projeto passa pela idealização, e em fases posteriores a respetiva implementação, de um sistemas capaz de simular campeonatos de corrida de veículos motorizados, nomeadamente automóveis.

Neste documento serão apresentadas as duas modelações iniciais relativas ao projeto acima mencionado: o **Modelo de Domínio** e o **Diagrama de Use Cases**.

O Modelo de Domínio é um modelo conceptual onde são apresentadas as entidades assim como as relações que estas têm entre si. Este modelo revela-se de extrema importância, uma vez que é através do mesmo que todas as modelações e implementações futuras serão idealizadas.

O Diagrama de Use Cases representa os diferentes atores que irão interagir com o sistema em questão. Cada ator terá associado a si um conjunto de cenários que surgem através das interação destes com o sistema. Estes cenários irão dar origem a diferentes Use Cases que subsequentemente serão explorados em detalhe nas Secções seguintes.

2 Modelo de Domínio

Como mencionado na Secção 1, o sistema proposto deve ser capaz de simular provas automobilísticas e garantir que o seu funcionamento cumpre com todas as especificações propostas.

Deste modo, surge a necessidade de criação de um Modelo de Domínio: uma representação conceptual, que, beneficiando-se da notação UML, tenciona demonstrar de forma simples e clara todas as interações que ocorrem no sistema a ser implementado. Após uma análise cuidada dos cenários de utilização apresentados no enunciado, foram identificadas as seguintes entidades:

- Jogador
- Administrador
- Jogo
- Campeonato
- Circuito
- Corrida
- Carro
- Motor
- Tipo de Pneu
- Categoria do carro
- Piloto

A primeira versão do Modelo de Domínio começa através da idealização do Jogo final. Partindo da análise da entidade Jogo, Figura 1, é possível observar que todas as relações representadas nesta fase inicial refletem as interações entre os atores e o sistema, tendo sido identificadas as seguintes relações:

- O Administrador, na sua capacidade de gestor do Jogo, tem o poder de adicionar um campeonato; um circuito, um carro e um piloto
- O Jogador pode por sua vez escolher: em que campeonato pretende correr e avaliar os circuitos que o compõem assim como piloto e o carro que pretende utilizar

Usando como exemplo o ator Administrador, é possível ver que a sua relação com a entidade Jogo é estabelecida através de uma relação de multiplicidade **1..*** para **1**, como apresentado na Figura 1, permitindo assim a existência de um ou mais Administradores que realizem a gestão do Jogo. A mesma lógica aplica-se para as restantes relações.

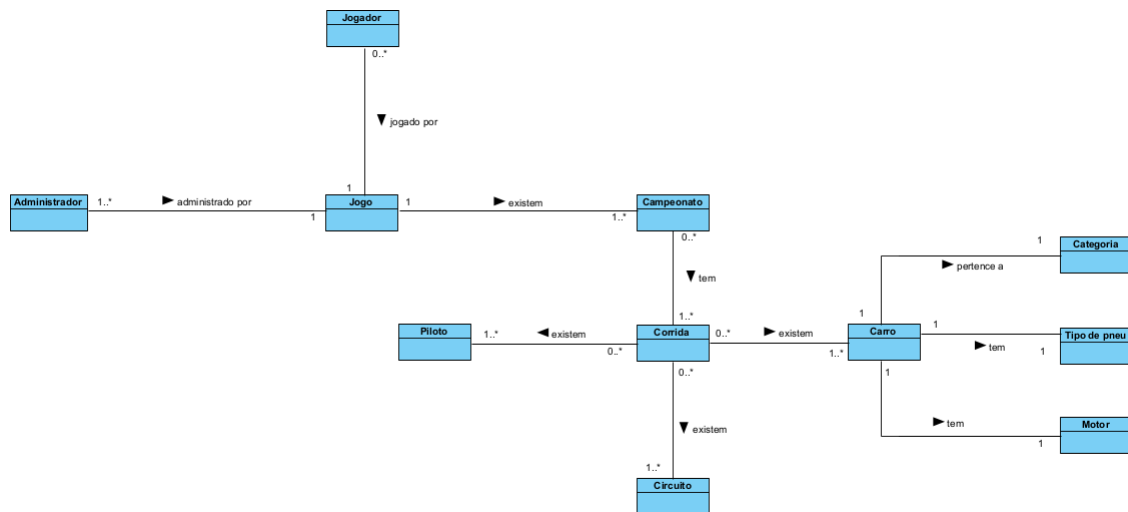


Figure 1: 1ª Versão do Modelo de Domínio

No entanto, esta primeira versão carece de detalhes e funcionalidades relevantes presentes na descrição dos sistema. Partindo da análise do enunciado é possível de deduzir deste modo a segunda versão do Modelo de Domínio, representado na Figura 2.

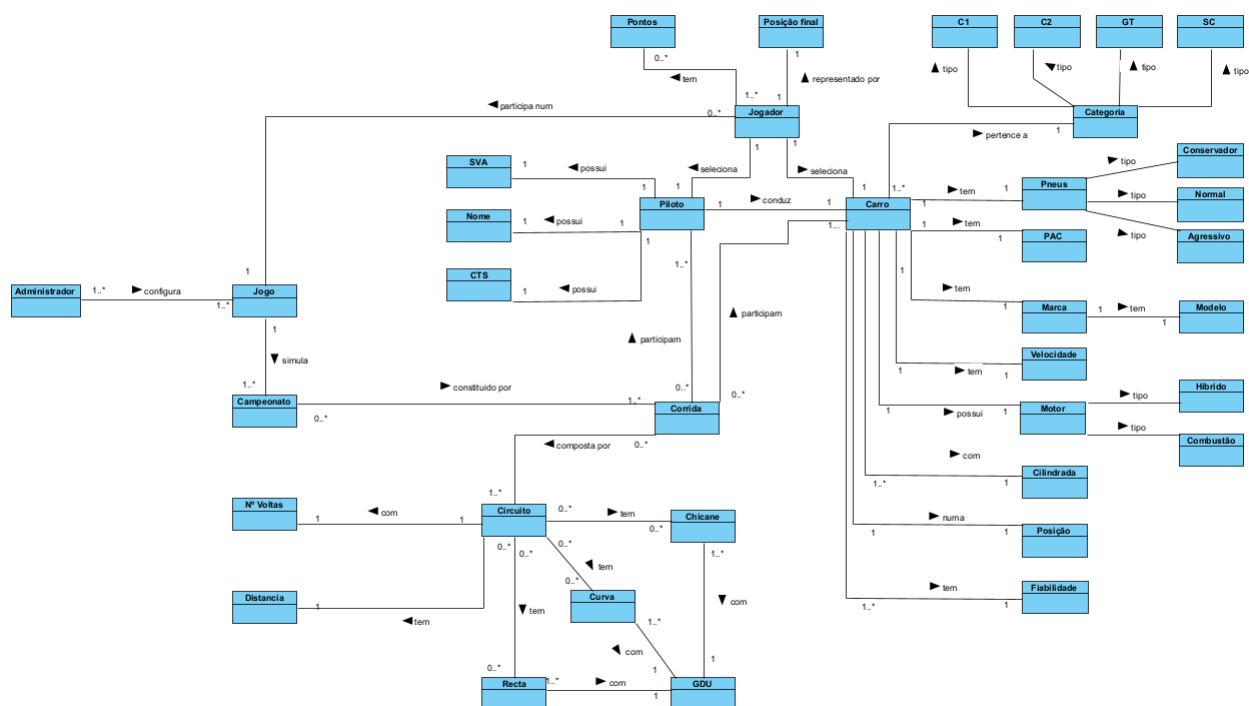


Figure 2: 2ª Versão do Modelo de Domínio

Como é possível observar na Figura 2, a complexidade do Modelo Domínio é consideravelmente maior; existindo um maior número de relações entre as diversas entidades. No entanto, estas não se revelaram suficientes de modo a que a representação dos sistema no modelo fosse o mais fiel possível da descrição fornecida pelo enunciado. Existe então a necessidade de implementar uma terceira, e última versão, do Modelo de Domínio apresentada na Figura 3.

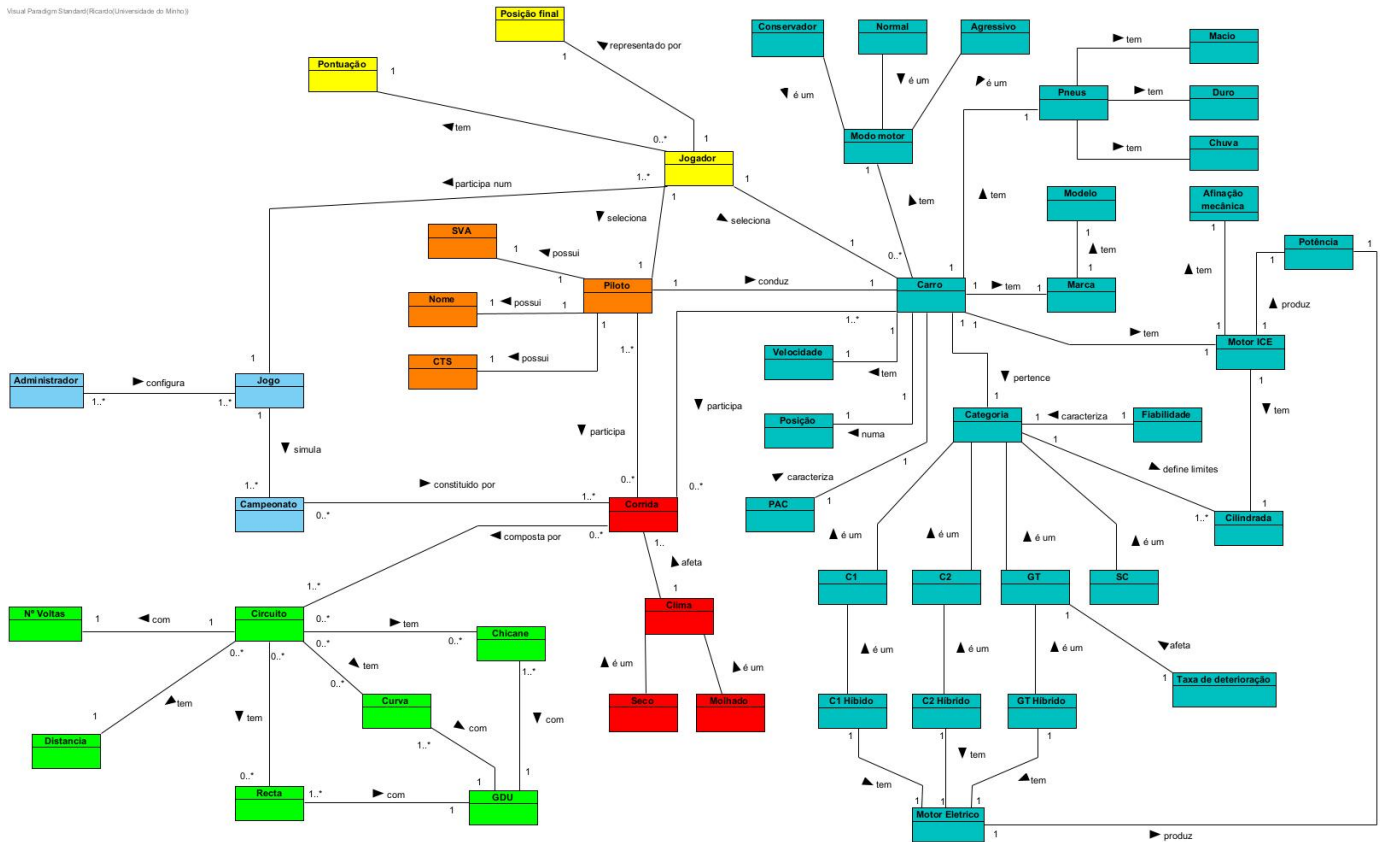


Figure 3: Versão final do Modelo de Domínio

A Figura 3 representa a versão final do Modelo de Domínio. Em comparação com as versões anteriormente apresentadas é possível observar que a complexidade aumentou consideravelmente, assim como as relações e dependências entre as diferentes entidades. Algumas das alterações realizadas, foram:

- A multiplicidade da entidade Jogador deixa de ser **0..*** para **1..*** dado que não faria sentido haver um *Jogo* sem jogadores;
- Foi criado *Modo Motor*, corrigindo deste modo a associação aos diferentes tipos de condução inicialmente atribuídos a *Pneu*;
- Tendo deste modo sido atribuído a *Pneu* as suas novas características;
- Foi adicionado o *Clima*;
- Tendo finalmente sido revistas e corrigidas todas as restantes multiplicidades.

Foi definido também um sistema de coloração para o Modelo, de modo a que este seja mais legível e compreensível. Tendo sido atribuídas as seguintes cores:

- Azul Escuro: Tudo o que implica o *Administrador*, o *Jogo* e o *Campeonato*;
- Azul Claro: Tudo o que é correlacionado com o *Carro*;
- Amarelo: Tudo o que é relativo ao *Jogador*;
- Laranja: Tudo o que está relacionado com o *Piloto*;
- Vermelho: Tudo o que está associado às *Corridas*;
- Verde: Tudo o que envolve os *Circuitos*.

3 Use Cases

Um Use Case é um descrição do modo como os atores interagem com o sistema com o intuito de concretizar um determinado cenário.

Deste modo, O Diagrama de Use Cases, Figura 4, representa a associação entre os atores e os cenários, Use Cases, a concretizar no decorrer da sua interação com o sistema.

Assim, a presente Secção 3 pretende apresentar o Diagrama em conjugação com pormenorização de cada Use Case.

3.1 Diagrama de Use Case

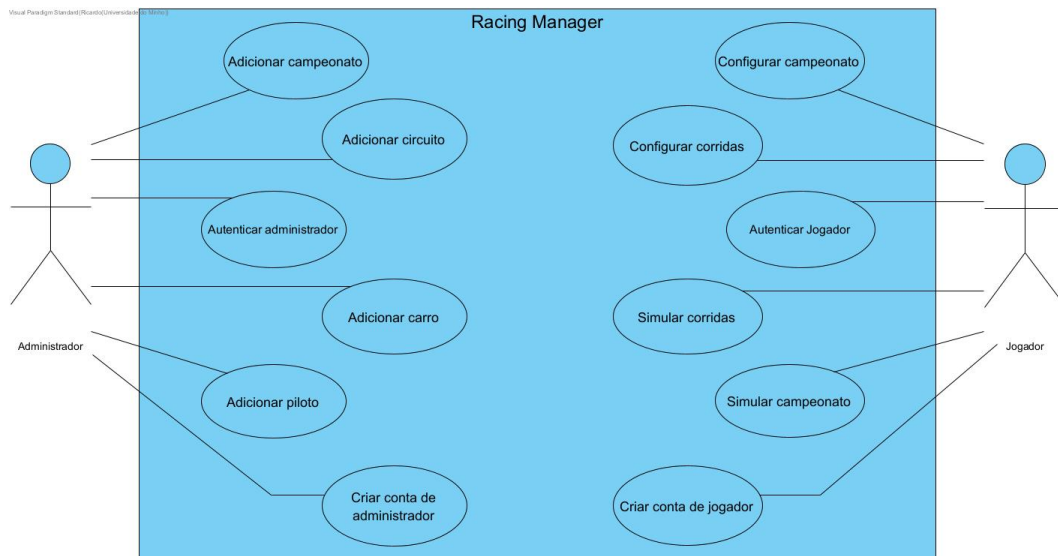


Figure 4: Diagrama de Use Cases

Como podemos observar pela Figura 4, os atores identificados foram:

- **Jogador:**

- Ator que representa as funcionalidades de jogabilidade do jogo;
- Pode consultar e escolher um campeonato; um circuito, um carro e um piloto para posteriormente jogar. Estes processos compõem o processo de configuração do campeonato;
- No final de cada corrida do campeonato o Jogador pode consultar os resultados obtidos;
- Tem também acesso a operações de criação e autenticação de conta.

- **Administrador:**

- Este ator representa a entidade com maior nível de manipulação do sistema;
- O Administrador pode adicionar novos campeonatos, circuitos, carros e pilotos.
- Tal como Jogador, tem acesso a processos de criação e autenticação de conta.

4 Especificação dos Use Cases

4.1 Use case: Criar conta de Administrador

Cenários: O José regista-se no Racing Manager como Administrador;

Ator: Administrador;

Pré-Condição: True;

Pós-Condição: o utilizador passa a ter uma conta de Administrador registada no sistema.

- **Fluxo normal**

- 1: O utilizador fornece um possível username e password ao sistema;
- 2: O sistema verifica se o username recebido já existe;
- 3: O sistema verifica se a password recebida já existe;
- 4: O sistema cria uma conta de Administrador associada ao utilizador.

- **Fluxo alternativo 1** [o username fornecido pelo utilizador já existe na base de dados] (passo 2)

- 2.1: O sistema informa que o username fornecido já está associado a outra conta;
- 2.2: O sistema pede ao utilizador para fornecer um username diferente;
- 2.3: Regressa ao passo 1.

4.2 Use case: Autenticar Administrador

Cenários: O José autentica-se como Administrador no Racing Manager;

Ator: Administrador;

Pré-Condição: True;

Pós-Condição: a autenticação é concluída e o utilizador pode interagir com o sistema.

- **Fluxo normal**
 - 1: O sistema pede ao utilizador o seu username e password;
 - 2: O utilizador fornece o seu username e password ao sistema;
 - 3: O sistema verifica se o utilizador existe na sua base de dados;
 - 4: O sistema verifica se a password fornecida corresponde à que tem armazenada para este utilizador;
 - 5: A autenticação é concluída e o utilizador tem acesso ao sistema como Administrador.

- **Fluxo alternativo 1** [o username não se encontra registado no sistema (antes da terceira tentativa)] (passo 3)
 - 3.1: O sistema indica que esse utilizador não existe na base de dados;
 - 3.2: Regressa ao passo 1.

- **Fluxo alternativo 2** [a password não coincide com a que tem armazenada para este utilizador (antes da terceira tentativa)] (passo 4)
 - 4.1: O sistema indica que essa password é incorreta;
 - 4.2: Regressa ao passo 1.

- **Fluxo de exceção 1** [o username não se encontra registado no sistema pela terceira vez] (passo 3)
 - 3.1: O sistema informa que esse utilizador não existe na sua base de dados;
 - 3.2: O processo de autenticação é cancelado.

- **Fluxo de exceção 2** [os dados fornecidos não existem na base de dados pela terceira vez] (passo 4)
 - 4.1: O sistema informa que os dados introduzidos pelo utilizador não existem na sua base de dados;
 - 4.2: O processo de autenticação é cancelado.

4.3 Use case: Criar conta de Jogador

Cenários: O Francisco regista-se no Racing Manager como jogador;

Ator: Jogador;

Pré-Condição: True;

Pós-Condição: o utilizador passa a ter uma conta de Jogador registada no sistema.

- **Fluxo normal**
 - 1: O utilizador fornece um possível username e password ao sistema;
 - 2: O sistema verifica se o username recebido já existe;
 - 3: O sistema verifica se a password recebida já existe;
 - 4: O sistema cria uma conta de Jogador associada ao utilizador.

- **Fluxo alternativo 1** [o username fornecido pelo utilizador já existe na base de dados] (passo 2)
 - 2.1: O sistema informa que o username fornecido já está associado a outra conta;
 - 2.2: O sistema pede ao utilizador para fornecer um username diferente;
 - 2.3: Regressa ao passo 1.

4.4 Use case: Autentica-se Jogador

Cenários: O Francisco autentica-se como jogador no Racing Manager;

Ator: Jogador;

Pré-Condição: True;

Pós-Condição: a autenticação é concluída e o utilizador pode interagir com o sistema.

- **Fluxo normal**
 - 1: O sistema pede ao utilizador o seu username e password;
 - 2: O utilizador fornece o seu username e password ao sistema;
 - 3: O sistema verifica se o utilizador existe na sua base de dados;
 - 4: O sistema verifica se a password fornecida corresponde à que tem armazenada para este utilizador;
 - 5: A autenticação é concluída e o utilizador tem acesso ao sistema como Jogador.
- **Fluxo alternativo 1** [o username não se encontra registado no sistema (antes da terceira tentativa)] (passo 3)
 - 3.1: O sistema indica que esse utilizador não existe na base de dados;
 - 3.2: Regressa ao passo 1.
- **Fluxo alternativo 2** [a password não coincide com a que tem armazenada para este utilizador (antes da terceira tentativa)] (passo 4)
 - 4.1: O sistema indica que essa password é incorreta;
 - 4.2: Regressa ao passo 1.
- **Fluxo de exceção 1** [o username não se encontra registado no sistema pela terceira vez] (passo 3)
 - 3.1: O sistema informa que esse utilizador não existe na sua base de dados;
 - 3.2: O processo de autenticação é cancelado.
- **Fluxo de exceção 2** [os dados fornecidos não existem na base de dados pela terceira vez] (passo 4)
 - 4.1: O sistema informa que os dados introduzidos pelo utilizador não existem na sua base de dados;
 - 4.2: O processo de autenticação é cancelado.

4.5 Use case: Adicionar campeonato

Descrição: Administrador adiciona um campeonato;

Cenários: Cenário 3 do enunciado;

Ator: Administrador;

Pré-Condição: O utilizador tem de estar autenticado como Administrador no sistema;

Pós-Condição: É adicionado um novo campeonato ao conjunto de campeonatos disponíveis.

- **Fluxo normal**

- 1: O Administrador fornece o nome para o novo campeonato;
- 2: O sistema verifica se o nome recebido é válido;
- 3: O Administrador selecciona dos circuitos que existem na base de dados quais vão fazer parte do campeonato;
- 4: O sistema adiciona o campeonato à lista.

- **Fluxo alternativo 1** [o nome recebido já foi atribuído a outro campeonato] (passo 2)

- 2.1: Sistema indica que esse nome já está a ser usado por outro campeonato;
- 2.2: O sistema pede ao Administrador para fornecer outro nome;
- 2.3: Regressa ao passo 1.

- **Fluxo de exceção 1** [administrador não adiciona um número suficiente de circuitos ao campeonato (pelo menos um)] (passo 3)

- 3.1: O sistema informa que não foram adicionados circuitos suficientes ao campeonato;
- 3.2: O sistema cancela o processo;

4.6 Use case: Adicionar circuito

Descrição: Administrador adiciona um circuito;

Cenários: Cenário 2 do enunciado;

Ator: Administrador;

Pré-Condição o utilizador tem de estar autenticado como Administrador no sistema;

Pós-Condição: é adicionado ao sistema um novo circuito.

- **Fluxo normal**

- 1: O Administrador indica o nome do novo circuito;
- 2: O sistema verifica se o nome recebido já está atribuído a outro circuito;
- 3: O Administrador indica a distância do circuito;
- 4: O Administrador indica o número de curvas presentes no circuito;
- 5: O Administrador indica o número de chicanes presentes no circuito;
- 6: O sistema calcula automaticamente o número de retas presentes no circuito;
- 7: O Administrador indica o GDU para cada uma das N curvas;

- 8: O Administrador indica o GDU para cada uma das N retas;
- 9: O sistema automaticamente atribui o GDU “difícil” para cada uma das N chicanes;
- 10: O Administrador indica o número de voltas que o circuito terá;
- 11: O sistema regista o circuito na sua base de dados.

- **Fluxo alternativo 1** [o nome fornecido pelo Administrador já está associado a outro circuito] (passo 2)
 - 2.1: O sistema informa o Administrador que já existe um circuito com aquele nome;
 - 2.2: O sistema pede ao Administrador para fornecer outro nome;
 - 2.3: Regressa ao passo 1.

4.7 Use case: Adicionar carro

Descrição: Administrador regista carro novo;

Cenários: Cenário 3 enunciado;

Ator: Administrador;

Pré-Condição: O utilizador tem de estar autenticado como Administrador no sistema;

Pós-Condição: É adicionado ao sistema um novo carro.

- **Fluxo normal**
 - 1: O Administrador seleciona uma determinada categoria de carro (C1, C2, GT, SC);
 - 2: O Administrador fornece a marca do carro;
 - 3: O Administrador fornece o modelo do carro;
 - 4: O Administrador fornece o valor da cilindrada do carro;
 - 5: O sistema verifica se o valor de cilindrada é adequado para a categoria selecionada no passo 1;
 - 6: O Administrador fornece o valor da potência do motor a combustão do carro;
 - 7: O Administrador fornece o valor do PAC do carro;
 - 8: O sistema regista o carro na sua base de dados.
- **Fluxo alternativo 1** [o valor de cilindrada fornecido pelo Administrador não se enquadra na categoria que selecionou] (passo 5)
 - 5.1: O sistema informa o Administrador que o valor de cilindrada não é válido;
 - 5.2: O sistema pede ao Administrador para fornecer outro valor;
 - 5.3: Regressa ao passo 4.
- **Fluxo alternativo 2** [o modelo fornecido pelo Administrador trata-se de um carro híbrido] (passo 6)
 - 6.1: O sistema permite ao Administrador fornecer um valor de potência para o motor elétrico do carro;
 - 6.2: Passa para o passo 7.

4.8 Use case: Adicionar piloto

Descrição: Administrador adiciona um piloto;

Cenários: Cenário 4 do enunciado;

Ator: Administrador;

Pré-Condição: o utilizador tem de estar autenticado como Administrador no sistema;

Pós-Condição: é adicionado ao sistema um novo piloto.

- **Fluxo normal**

- 1: O Administrador indica o nome do piloto;
- 2: O sistema verifica se o nome já está atribuído a outro piloto;
- 3: O Administrador fornece o valor do CTS do piloto;
- 4: O Administrador fornece o valor do SVA do piloto;
- 5: O sistema regista o piloto na sua base de dados.

- **Fluxo alternativo 1** [o nome fornecido pelo Administrador já está atribuído a outro piloto] (passo 2)

- 2.1: O sistema informa o Administrador que o nome fornecido já está atribuído a outro piloto;
- 2.2: O sistema pede ao Administrador para fornecer outro nome;
- 2.3: Regressa ao passo 1.

4.9 Use case: Configurar campeonato

Descrição: Jogador configura um campeonato;

Cenários: Cenário 5 do enunciado;

Ator: Jogador;

Pré-Condição: O campeonato já deve existir no sistema;

Pós-Condição: O campeonato fica configurado.

- **Fluxo normal**

- 1: O Jogador seleciona da lista uma determinada corrida;
- 2: O Jogador seleciona da lista um determinado carro;
- 3: O Jogador seleciona da lista um determinado piloto;
- 4: O sistema informa que o campeonato está configurado.

4.10 Use case: Configurar corridas

Descrição: Jogador configura a corrida;

Cenários: Cenário 5 do enunciado;

Ator: Jogador;

Pré-Condição: A corrida já deve existir no sistema;

Pós-Condição: A corrida fica configurada.

- **Fluxo normal**

- 1: O Jogador seleciona da lista um circuito;
- 2: O Jogador seleciona da lista uma condição meteorológica;
- 3: O Jogador seleciona o valor da afinação;
- 4: O Jogador seleciona da lista um determinado pneu;
- 5: O Jogador seleciona da lista um determinado modo motor;
- 6: O sistema informa que a corrida está configurada.

4.11 Use case: Simular corrida

Descrição: Jogador simula a corrida;

Cenários: Cenário 5 do enunciado;

Ator: Jogador;

Pré-Condição: A corrida e o carro tem de estar configurados;

Pós-Condição: A corrida é simulada.

- **Fluxo normal**

- 1: O sistema dá informações sobre as posições dos carros/pilotos/jogadores após cada volta;
- 2: O sistema dá informações sobre as ultrapassagens efetuadas;
- 3: O sistema dá informações sobre os despistes ocorridos;
- 4: O sistema dá informações sobre a ocorrência de avarias;
- 5: O sistema informa no final da corrida as posições dos jogadores;
- 6: O sistema informa no final da corrida as pontuações dos jogadores.

- **Fluxo Alternativo 1** [o Jogador tem versão base do jogo] (passo 3)

- 3.1: O simulador decide se cada carro consegue ultrapassar o carro à sua frente;
- 3.2: Regressa ao passo 3.

- **Fluxo Alternativo 2** [carro não consegue ultrapassar o carro à sua frente] (passo 3.1)

- 3.1.1: Carro sai da pista;
- 3.1.2: Regressa ao passo 3.2.

4.12 Use case: Simular campeonato

Descrição: Jogador simula um campeonato;

Cenários: Cenário 5 do enunciado;

Ator: Jogador;

Pré-Condição: O campeonato tem de estar configurado;

Pós-Condição: A pontuação é atribuída.

- **Fluxo normal**

- 1: O sistema soma as pontuações obtidas por cada jogador nas corridas;
- 2: O sistema verifica se todos os utilizadores fizeram login;
- 3: O sistema soma pontuação do campeonato com a pontuação global do jogador;
- 4: O sistema apresenta a posição de cada jogador no ranking do Racing Manager.

- **Fluxo alternativo 1** [caso tenha login efetuado] (passo 2)

- 2.1: O sistema atribui ao jogador pontos por ter efetuado o login;
- 2.2: Passa para o passo 3.

- **Fluxo alternativo 2** [caso não tenha login efetuado] (passo 2)

- 3.1: O sistema descarta a pontuação atribuída por login, excluindo-a do raking;
- 3.2: Passa para o passo 4.

5 Conclusão

O presente documento teve o intuito de apresentar e explicar o modo como o enunciado foi percebido, podendo deste modo ser afirmado que o objetivo primordial desta primeira fase foi atingido com sucesso.

Encontra-se neste trabalho apresentada uma solução robusta ao problema apresentado, uma vez que os diagramas construídos tem uma base sólida e bem fundamentada tendo esta sido apresentada nas Secções anteriores.

Deste modo, através de uma análise profunda ao enunciado disponibilizado foi possível, iterativamente, aprimorar os diagramas representados assim como a definição dos diversos Use Cases. Contempla-se então, que a base aqui estabelecida se encontra em concordância com as *guidelines* fornecidas. A solidez desta base irá permitir uma maior agilização na implementação futura desta solução.

Não obstante, é possível que sejam realizadas alterações futuras de modo a enriquecer o presente trabalho no decorrer das próximas fases.