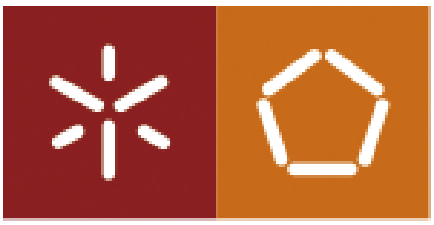

Universidade do Minho
Escola de Engenharia

Desenvolvimento de Sistemas de Software

Licenciatura em Engenharia Informática

F1 Manager

1ª Fase - Grupo 19

Afonso Bessa – 95225

Telmo Oliveira – 97018

Hugo Novais – 96267

Martim Ribeiro - 96113

https://github.com/HNovais/DSS

outubro, 2022

Índice

Índice de Figura

Capítulo 1: Introdução

No âmbito da Unidade Curricular de Desenvolvimento de Sistemas de Software foi-nos proposto a elaboração de um sistema capaz de simular campeonatos de automobilismo. Na sua base a aplicação final será semelhante à do **F1 Manager**, uma vez que tratar-se-á de uma aplicação de simulação de provas automobilísticas entre jogadores.

Nesta primeira fase, elaboramos um Modelo de Domínio, que retrata as diferentes entidades do nosso jogo, essenciais para o seu funcionamento. Além disso, apresentamos também um Diagrama de Use Cases, que demonstra as interações dos atores com o sistema e, por fim, uma descrição detalhada dos diferentes Use Cases.

Capítulo 2 - Modelo de Domínio

Com base nos requisitos apresentados pelos Cenários de Utilização elaboramos, usando a notação UML lecionada nas aulas, o Modelo de Domínio abaixo descrito, com o objetivo de representar as Entidades que compõem o sistema, bem como as Relações entre estas.

Capítulo 2.1 - Descrição do modelo

Numa primeira abordagem, entendemos focar-nos em cinco Entidades que consideramos fundamentais ao Sistema e não deixariam dúvidas de que seriam necessárias. Optamos por trabalhá-las separadamente e, lendo o enunciado com mais detalhe, adicionar tudo aquilo que seria pertinente o suficiente para ser considerado uma Entidade.

Estas Entidades são:

1. Campeonato,
2. Circuito,
3. Corrida,
4. Carro,
5. Piloto.

Sendo a maioria Entidades pouco complexas, a única que necessitou de mais atenção foi o Carro, levando até que fosse necessário mudar de abordagem algumas vezes.

Numa primeira leitura, criamos uma Entidade Categoria, que tinha uma relação com às várias Entidades C1, C2, GT e SC e, posteriormente, cada uma destas tinha uma relação com um ou mais motores (dependendo de quantos podia ter), como mostra a figura em baixo:

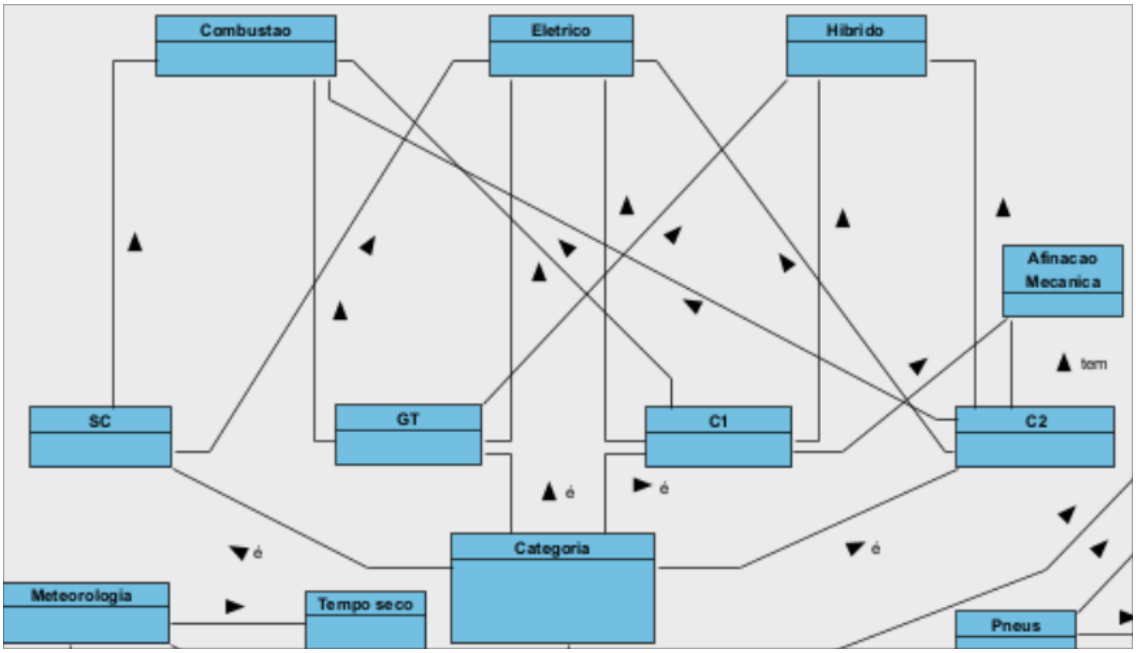

Figura 1 - Modelo de Domínio (Carro)

Posteriormente, este modelo foi abandonado por ser demasiado confuso, caótico e não ter sequer em conta a Entidade Potência. Acabamos por chegar a outra conclusão, como se pode ver na figura 1, onde passamos a ter o Motor de combustão como Entidade obrigatória de todas os Carros e as várias categorias passaram a estar mais bem organizadas.

Acabando de organizar todas as Entidades, decidimos então começar a pensar em como iríamos ligá-las. Sem grande dificuldade, achamos correto ligar tudo "seguido". Ou seja, um Campeonato é constituído por um ou mais Circuitos, num Circuito realizam uma ou mais Corridas, uma Corrida tem dois ou mais Carros e um Carro tem um Piloto. Houve, no entanto, algum debate sobre a ordem do Circuito e da Corrida. Ou seja, ponderamos em fazer o modelo de maneira que o Campeonato tivesse Corridas e essas fossem realizadas em Circuitos. Após alguma trocas de ideias por parte de todo o grupo de trabalho, decidimos manter como foi feito inicialmente.

Por último, faltava-nos abordar os Utilizadores e Administradores. Foi decidido que o Administrador poderia criar Campeonatos, Circuitos, Carros e Pilotos e o Jogadores poderia jogar Campeonatos.

O resto das Entidades e Relações são relativamente autoexplicativas.

Chegamos assim ao modelo final. Resumidamente, temos um administrador que pode criar campeonatos, circuitos, carros e pilotos e um jogador que pode jogar campeonatos. Cada campeonato tem um número limite de participantes e tem uma classificação. Cada circuito tem retas, curvas e pode ter chicanes. A todos corresponde um respetivo GDU. Cada circuito tem também uma distância. Cada corrida tem uma meteorologia, um número de voltas e uma grelha de partida. Cada carro tem uma marca, modelo, pneus, PAC e um motor de combustão. Cada carro pertence a uma categoria. Cada piloto tem um nível de perícia.

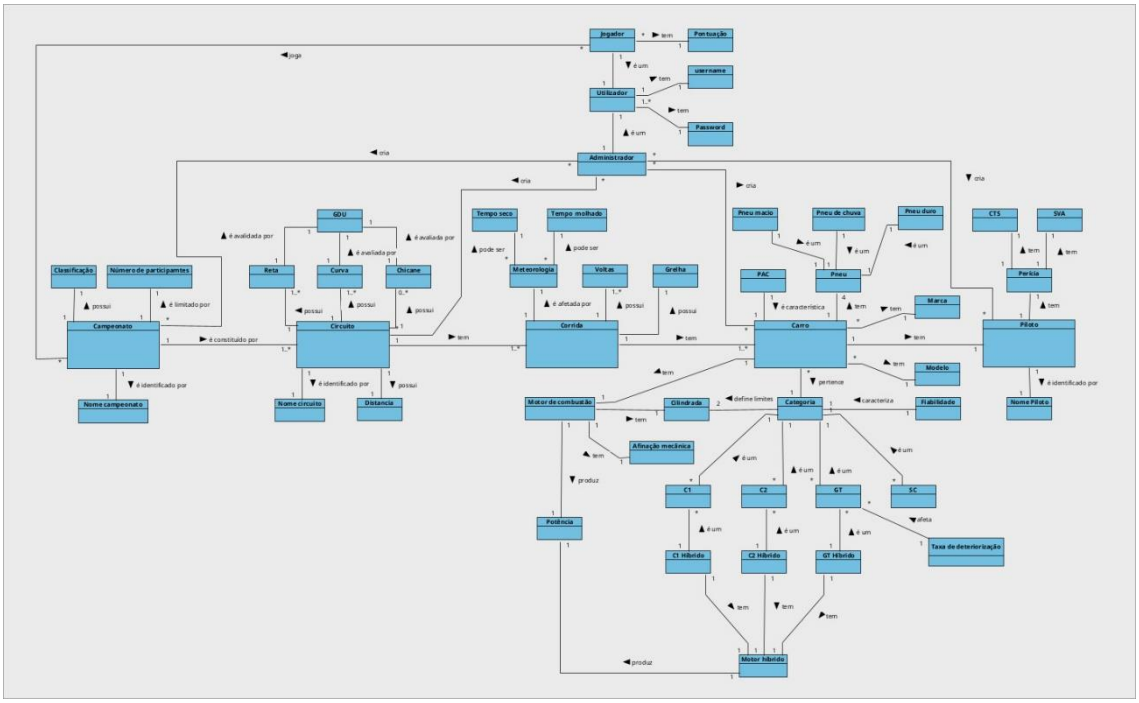

Figura 2 - Modelo de Domínio

Capítulo 3 - Diagrama de Use Cases

Foi também elaborado um Diagrama dos Use Cases com o intuito de demonstrar como os atores interagem com o sistema perante os diferentes cenários.

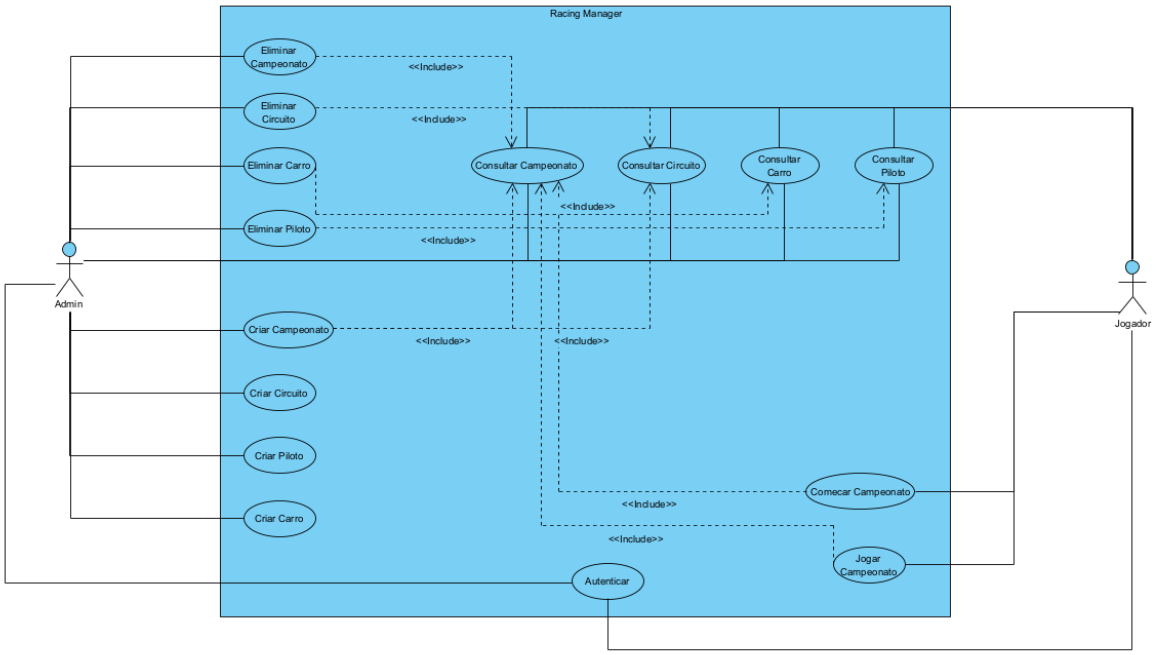

Figura 3 - Diagrama de Use Cases

Neste diagrama podemos também salientar a presença dos <includes> que são necessários tanto pelo Administrador como pelo Jogador, para evitar a repetição dos mesmos fluxos.

Pelo administrador, uma vez que sempre que pretende criar algo tem de consultar se o mesmo já existe recorrendo a um Use Case de Consulta ou caso deseje Eliminar algo necessita de Consultar a lista dos que existem, recorrendo também a um Use Case do mesmo tipo.

Pelo jogador quando se pretende Jogar Campeonato ou Começar Campeonato consultando a lista dos campeonatos disponíveis.

Capítulo 3.1 - Identificação dos Atores

Ao analisar o nosso Modelo de Domínio podemos identificar dois atores:

- **Administrador:** Responsável pela manutenção do jogo tendo a possibilidade de criar Campeonatos, Circuitos, Carros e Pilotos, bem como eliminar os mesmos caso assim pretenda.
- **Jogador:** Trata-se do normal utilizador do jogo com a possibilidade de jogar os diferentes Campeonatos e para isso escolher o seu Carro e Piloto pretendido.

Capítulo 3.2: Descrição dos Use Cases

Use Cases: Admin

Os *Use Cases* do tipo Criar e Eliminar só se encontram disponíveis para utilizadores autenticados como Administradores. Estes possibilitam alterações nos "dados" do jogo.

1. Criar Campeonato

Use Case: Criar Campeonato.

Descrição: Administrador cria um campeonato .

Cenários: O José faz login no jogo como administrador e cria um campeonato e adiciona-o à lista de campeonatos disponíveis. Começa por lhe dar o nome e, de seguida, escolhe um ou vários circuitos da lista de circuitos disponíveis. Depois de consultar a lista de campeonatos actualmente disponíveis para jogar, decide adicionar o campeonato criado à mesma, pelo que este fica imediatamente disponível para ser jogado.

Pré-Condição: Ator está autenticado.

Pós-Condição: Sistema fica com mais um campeonato disponível para jogar.

Fluxo Normal:

1. Ator escolhe nome do campeonato.
2. Sistema aprova nome.
3. <include> Consultar Circuitos
4. Ator escolhe circuitos.
5. <include> Consultar Campeonatos.
5. Ator adiciona o campeonato criado à lista.
6. Sistema cria campeonato.

Fluxo Alternativo (1) [Nome já existe] (Passo 1):

1.1 Sistema avisa que o nome fornecido já existe.

1.2 Sistema retorna a dar a possibilidade de escolher um nome.

1.3 Regressa ao passo 1.

2. Eliminar Campeonato

Use Case: Eliminar Campeonato.

Descrição: Administrador elimina um campeonato.

Cenários: O José faz login no jogo como admnistrador e acede à lista de campeonatos disponíveis. Seleciona um campeonato e elimina-o. O campeonato fica imediatamente indisponível para ser jogado.

Pré-Condição: Ator está autenticado.

Pós-Condição: Sistema fica com menos um campeonato disponível para jogar.

Fluxo Normal:

1. <include> Consultar Campeonatos.
2. Ator seleciona um campeonato para eliminar.
3. Sistema pergunta se o Ator realmente deseja eliminar o campeonato.
4. Ator responde que sim.
5. Sistema elimina campeonato.

Fluxo de Exceção (1) [Ator responde que não] (Passo 4):

4.1 Sistema verifica que a resposta do Ator foi não

4.2 Sistema termina processo.

3. Criar Circuito

Use Case: Criar Circuito.

Descrição: Administrador cria um Circuito.

Cenários: O José faz login no jogo como administrador e opta por adicionar um novo circuito. Indica o nome do novo circuito a adicionar. De seguida, indica que o mesmo tem 2Km, 9 curvas e 1 chicane. Com essa informação, o sistema calcula que o circuito tem 10 rectas e apresenta a lista de curvas e rectas de modo a que o José indique o grau de dificuldade de ultrapassagem (GDU) em cada uma. Finalmente, regista o circuito, indicando que cada corrida deverá ter 10 voltas. O circuito passa a estar disponível para integrar campeonatos.

Pré-Condição: Ator está autenticado.

Pós-Condição: Sistema fica com mais um circuito disponível nos campeonatos.

Fluxo Normal:

1. Ator escolhe nome do circuito.

2. Sistema aprova nome.

3. Ator escolhe quilómetros, números de curvas e chicanes.

4. Sistema apresenta lista de curvas e retas.

5. Ator indica o GDU em cada reta, curva e chicane e número de voltas em cada corrida.

6. Sistema cria circuito.

Fluxo Alternativo (1) [Nome já existe] (Passo 2):

2.1 Sistema avisa que o nome fornecido já existe.

2.2 Sistema retorna a dar a possibilidade de escolher um nome.

2.3 Regressa ao passo 1.

4. Eliminar Circuito

Use Case: Eliminar Circuito.

Descrição: Administrador elimina um circuito.

Cenários: O José faz login no jogo como admnistrador e acede à lista de circuitos disponíveis. Seleciona um circuitos e elimina-o. O circuito fica imediatamente indisponível.

Pré-Condição: Ator está autenticado.

Pós-Condição: Sistema fica com menos um circuito disponível para jogar e todos os campeonatos que o possuíam deixam de o ter.

Fluxo Normal:

1. <include> Consultar Circuito.
2. Ator seleciona um circuito para eliminar.
3. Sistema pergunta se o Ator realmente deseja eliminar o circuito.
4. Ator responde que sim.
5. Sistema elimina circuito.

Fluxo de Exceção (1) [Ator responde que não] (Passo 4):

4.1 Sistema verifica que a resposta do Ator foi não

4.2 Sistema termina processo.

5. Criar Carro

Use Case: Criar Carro.

Descrição: Administrador cria um Carro.

Cenários: O José faz login no jogo como administrador e opta por adicionar um novo carro. Começa por consultar as categorias disponíveis nesta versão do jogo. O José opta por uma categoria e, de seguida, indica a marca e modelo do carro. O José fornece os valores da cilindrada e da potência do motor (ou motores caso tenha eletrico). Finalmente, escolhe o perfil aerodinâmico do carro (PAC). Termina o registo do carro, que fica disponível para ser utilizado em jogos.

Pré-Condição: Ator está autenticado.

Pós-Condição: Sistema fica com mais um carro disponível para jogar.

Fluxo normal:

1. Sistema apresenta categorias disponíveis.
2. Ator escolhe categoria C2.
3. Sistema verifica que categoria é C2.
4. Ator indica marca, modelo, cilindrada e potência do motor de combustão.
5. Ator indica que carro não é híbrido.
6. Ator indica PAC.
7. sistema cria carro.

Fluxo Alternativo (1): [Carro é C1] (passo 2)

1.1 Ator escolhe categoria C1.

1.2 Sistema verifica que categoria é C2.

1.3 Ator indica marca, modelo, cilindrada e potência do motor de combustão.

1.4 Ator indica que carro não é híbrido.

1.5 Ator indica PAC.

1.6 Sistema cria carro.

Fluxo Alternativo (2): [Carro é GT] (passo 2)

2.1 Ator escolhe categoria GT.

2.2 Sistema verifica que categoria é GT.

2.3 Ator indica marca, modelo, cilindrada e potência do motor de combustão.

2.4 Ator indica que carro não é híbrido.

2.5 Ator indica PAC.

2.6 Sistema cria carro.

Fluxo Alternativo (3): [Carro é SC] (passo 2)

3.1 Ator escolhe categoria SC.

3.2 Sistema verifica que categoria é SC.

3.3 Ator indica marca, modelo, cilindrada e potência do motor de combustão.

3.4 Ator indica PAC.

3.5 Sistema cria carro.

Fluxo Alternativo (4) [Carro é híbrido] (passo 5)

4.1 Ator indica que carro é híbrido.

4.2 Ator indica potência do motor elétrico.

4.3 Ator indica PAC.

4.4 Sistema cria carro.

6. Eliminar Carro

Use Case: Eliminar Carro.

Descrição: Administrador elimina um carro.

Cenários: O José faz login no jogo como admnistrador e acede à lista de carros disponíveis. Seleciona um carro e elimina-o. O carro fica imediatamente indisponível.

Pré-Condição: Ator está autenticado .

Pós-Condição: Sistema fica com menos um carro disponível para jogar.

Fluxo Normal:

1. <include> Consultar Carro.
2. Ator seleciona um carro para eliminar.
3. Sistema pergunta se o Ator realmente deseja eliminar o carro.
4. Ator responde que sim.
5. Sistema elimina carro.

Fluxo de Exceção (1) [Ator responde que não] (Passo 4):

4.1 Sistema verifica que a resposta do Ator foi não

4.2 Sistema termina processo.

7. Criar Piloto

Use Case: Criar Piloto.

Descrição: Administrador cria um piloto.

Cenário: O José faz login no jogo como administrador e decide adicionar um novo piloto. Começa por indicar que o nome e, de seguida, os seus níveis de perícia. No critério "Chuva vs. Tempo Seco" (CTS), indicou um valor de 0.6, indicando um ligeiro melhor desempenho em tempo seco. No critério "Segurança vs. Agressividade" (SVA), indicou um valor de 0.4, indicando que o piloto tende a arrisca pouco. Finalmente, regista o piloto, que fica disponível.

Pré-Condição: Ator está autenticado.

Pós-Condição: Sistema fica com mais um piloto fica disponível.

Fluxo Normal:

1. Ator escolhe nome do campeonato.
2. Sistema aprova nome e que necessita dos níveis de perícia.
3. Ator indica níveis de perícia.
4. Ator indica CTS.
5. Sistema verifica que o valor está entre 0 e 1.
6. Ator indica SVA.
7. Sistema verifica que o valor está entre 0 e 1.
8. Sistema cria piloto.

Fluxo Alternativo (1) [Nome já existe] (Passo 2):

2.1 Sistema avisa que o nome fornecido já existe.

2.2 Sistema retorna a dar a possibilidade de escolher um nome.

2.3 Regressa ao passo 3.

Fluxo de Exceção (2) [CTS fora do intervalo 0 a 1] (passo 5)

5.1 Sistema verifica que CTS se encontra fora do intervalo de 0 a 1.

5.2 Sistema termina processo.

Fluxo de Exceção (3) [SVA fora do intervalo 0 a 1] (passo 7)

7.1 Sistema verifica que CTS se encontra fora do intervalo de 0 a 1.

7.2 Sistema termina processo.

8. Eliminar Piloto

Use Case: Eliminar Piloto.

Descrição: Administrador elimina um piloto.

Cenários: O José faz login no jogo como admnistrador e acede à lista de pilotos disponíveis. Seleciona um piloto e elimina-o. O carro fica imediatamente indisponível.

Pré-Condição: Ator está autenticado.

Pós-Condição: Sistema fica com menos um piloto disponível para jogar.

Fluxo Normal:

1. <include> Consultar Piloto.
2. Ator seleciona um piloto para eliminar.
3. Sistema pergunta se o Ator realmente deseja eliminar o piloto.
4. Ator responde que sim.
5. Sistema elimina piloto.

Fluxo de Exceção (1) [Ator responde que não] (Passo 4):

4.1 Sistema verifica que a resposta do Ator foi não

4.2 Sistema termina processo.

Use Cases: Utilizador

Os *Use Cases* do tipo Consultar e Autenticar encontram-se disponíveis para todos os Utilizadores e permitem, tal como o nome indica, consultar listas de entidades do jogo e realizar a autenticação no jogo, respetivamente.

1. Consultar Campeonatos

Use Case: Consultar Campeonatos.

Descrição: Jogador consulta os campeonatos disponíveis.

Cenário: José decide consultar os campeonatos que no momento da sua pesquisa se encontram disponíveis para jogar.

Pré-condição: True.

Pós-condição: Jogador visualizou os campeonatos.

Fluxo Normal:

1. Sistema devolve lista com os campeonatos disponíveis.

2. Consultar Circuito

Use Case: Consultar Circuito.

Descrição: Jogador consulta os circuitos disponíveis.

Cenário: José decide consultar os circuitos que no momento da sua pesquisa se encontram disponíveis para jogar.

Pré-condição: True.

Pós-condição: Jogador visualizou os circuitos.

Fluxo Normal:

1. Sistema devolve lista com os circuitos disponíveis.

3. Consultar Carro

Use Case: Consultar Carro.

Descrição: Jogador consulta os carros disponíveis.

Cenário: José decide consultar os carros que no momento da sua pesquisa se encontram disponíveis para jogar.

Pré-condição: True.

Pós-condição: Jogador visualizou os carros.

Fluxo Normal:

1. Sistema devolve lista com os carros disponíveis.

4. Consultar Piloto

Use Case: Consultar Piloto.

Descrição: Jogador consulta os pilotos disponíveis.

Cenário: José decide consultar os pilotos que no momento da sua pesquisa se encontram disponíveis para jogar.

Pré-condição: True.

Pós-condição: Jogador visualizou os pilotos.

Fluxo Normal:

1. Sistema devolve lista com os Pilotos disponíveis.

5. Autenticar

Use Case: Autenticar.

Descrição: Utilizador autentica-se no sistema.

Cenário: No final do campeonato as posições foram: Francisco, Sara, Daniela e Manuel. Os 12 pontos do primeiro lugar no campeonato são somados à pontuação global do Francisco e com isso ele sobe ao segundo lugar do *ranking* do *Racing Manager*. A Daniela autentica-se para que os 10 pontos do segundo lugar sejam contabilizados no seu total. A Sara e o Manuel não têm conta (ou não fazem login), pelo que os pontos deste campeonato não são contabilizados no *ranking*.

Pré-Condição: True.

Pós-Condição: Ator fica autenticado.

Fluxo Normal:

1. Sistema pede *Username* e *Password.*
2. Ator insere os dados de jogador.
3. Sistema valida dados e dá ocorrência do login.
4. Ator fica autenticado como jogador.

Fluxo Alternativo (1) [Ator autentica-se como administrador] (Passo 2):

2.1 Ator insere dados de administrador.

2.2 Sistema valida dados e dá ocorrência do login.

2.3 Ator fica autenticado como administrador.

Fluxo de Exceção (2) [Ator insere dados errados] (Passo 3):

3.1 Sistema não valida dados.

3.2 Sistema termina o processo.

Use Cases: Jogador

Os *Use Cases* do tipo Começar Campeonato e Jogar Campeonato só se encontram disponíveis para utilizadores autenticados como Jogador.

1. Começar Campeonato

Use Case: Começar Campeonato.

Descrição: Jogador começa novo campeonato.

Cenário: O José e três amigos resolver jogar um campeonato de *Racing Manager*. O José escolhe um campeonato e um piloto. Após inscrever-se, cada um dos amigos escolhe também o seu carro e piloto.

Pré-Condição: Ator que começa campeonato está autenticado.

Pós-Condição: Sistema inicia um campeonato e os jogadores estão registados

Fluxo Normal:

1. <include> Consulta Campeonatos.
2. Ator escolhe um campeonatos.
3. Sistema pede número e nome dos jogadores.
4. Ator fornece número e nome dos jogadores.
5. Sistema pede piloto e carro dos jogadores.
6. Jogadores escolhem piloto e carro.
7. Sistema exibe as escolhas feitas e pede confirmação para iniciar.
8. Jogador responde que sim e novo campeonato é iniciado.
9. Sistema inicia um campeonato e os jogadores estão registados.

Fluxo Alternativo (1) [Ator responde que não] (Passo 9):

9.1 Sistema verifica que a resposta do Ator foi não

9.1 Sistema termina processo.

2. Jogar Campeonato

Use Case: Jogar Campeonato.

Descrição: Jogador continua campeonato.

Cenário: José consulta a lista de campeonatos disponíveis e escolhe um campeonato para jogar. As condições da primeira corrida e a situação meteorológica são apresentadas. Cada um dos jogadores decide se pretende alterar a afinação do carro, tendo em consideração que, por se tratar de um campeonato com três corridas, apenas poderão fazer duas afinações ao longo do mesmo. Após considerar as características do circuito, do carro e do piloto, o Francisco decide alterar a afinação
e aumenta a *downforce* de 0.5 para 0.7. Finalmente, todos os jogadores devem escolher os pneus e modo do motor a usar na corrida**.**

Pré-Condição: True.

Pós-Condição: Campeonato acaba uma corrida situada num circuito e avança para a próxima.

Fluxo Normal:

1. <include> Consulta Campeonatos.
2. Ator escolhe um campeonato.
3. Sistema indica o circuito e a meteorologia.
4. Sistema pergunta se pretende alterar as afinações dos carros.
5. Ator indica que sim.
6. Sistema indica o número máximo de afinações possíveis (2/3 do número de corridas no campeonato).
7. Ator efetua as afinações.
8. Sistema pergunta se pretende alterar o *downforce.*
9. Ator altera o *downforce*.
10. Sistema pede o tipo de pneus e o modo do motor de cada carro.
11. Jogador seleciona os pneus e modo do motor pretendidos.
12. Sistema simula corrida e apresenta os resultados.

Fluxo Alternativo (1) [Ator não pretende alterar as afinações] (Passo 5):

5.1 Ator indica que não pretende alterar as afinações

5.2 Vai para o passo 8.

Fluxo Alternativo (2) [Ator não pretende alterar o *downforce*] (Passo 9):

9.1 Ator indica que não pretende alterar as afinações

9.2 Vai para o passo 10.

Fluxo de Exceção (3) [Ator efetua um número de afinações acima ao permitido] (Passo 7):

7.1 Sistema regista um número de afinações acima do permitido

7.2 Sistema termina processo

Fluxo de Exceção (2) [*Downforce* fora do intervalo 0 a 1] (passo 9)

9.1 Sistema verifica que CTS se encontra fora do intervalo de 0 a 1.

9.2 Sistema termina processo.

Capítulo 4: Análise Crítica

Numa prespetiva crítica, acreditamos que a análise exaustiva do enunciado e o constante acompanhamento das aulas práticas da UC foram fulcrais no desenvolvimento e conclusão desta primeira fase do projeto.

Estamos bastante satisfeitos com o resultado obtido e achamos que a nossa solução retrata bem a ideia proposta e que nos será de grande ajuda nas próximas etapas do projeto.

Capítulo 5: Conclusão

A elaboração desta 1ª fase do projeto será sem dúvida alguma muito importante e indispensável no desenvolvimento da aplicação final. Tanto o Modelo de Domínio como o Diagrama dos Use Cases serão fundamentais para guiar o projeto daqui para a frente, como também nos darão uma ideia de que classes serão necessárias, fornecendo-nos assim uma visão geral de todo o projeto permitindo evitar lapsos futuros.

Já os Use Cases, que retratam as interações entre Sistema e os Atores, serão fundamentais no desenrolar dos métodos que possibilitarão o funcionamento correto do nosso programa.