



Universidade do Minho  
Escola de Engenharia  
Licenciatura em Engenharia informática

## Unidade Curricular de Desenvolvimento de Sistemas de Software

Ano Letivo de 2022/2023

Grupo 31

Inês Ferreira (A97372)



José Ferreira (A97642)



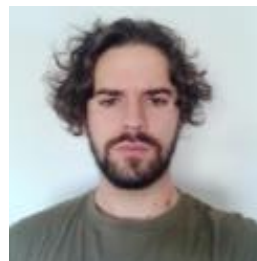
Joana Branco (A96584)



Marta Sá (A97158)



Robert Szabo (A91682)



**URL do repositório:** <https://github.com/venicexbish/DSS>

Braga, 20 de outubro de 2022

## Índice

1. Introdução	3
2. Objetivos	4
3. Descrição	5
4. Modelo de Domínio	6
5. Modelo de Use Case	7
6. Anexos	9
6.1. Use Case	
10	
6.1.1 Criar Conta	10
6.1.2 Fazer <i>Login</i>	10
6.1.3 Criar Campeonato	11
6.1.4 Criar Circuito	11
6.1.5 Criar Carro	12
6.1.6 Criar Piloto	13
6.1.7 Configurar Campeonato	13
6.1.8 Configurar Corrida	14
6.1.9 Simular Corrida	15
6.1.10 Consultar Resultado Final	16

# 1. Introdução

Este relatório foi desenvolvido no âmbito da Unidade Curricular de Desenvolvimento de Sistemas de Software onde foi-nos proposta a criação de um sistema, semelhante ao *F1 Manager*, capaz de simular campeonatos de automobilismo. Assim, a aplicação tem como objetivo criar um ambiente de jogo onde os seus participantes podem competir entre si.

De forma a garantir o sucesso da aplicação, deve ser traçado um plano. Numa fase inicial, é feito o levantamento dos requisitos apoiados pelos cenários fornecidos no enunciado. De seguida, os requisitos passam por uma análise e, posteriormente, é criado um modelo com base neles.

Para a conceção e desenvolvimento da aplicação são criados modelos em UML. A ferramenta usada para a especificação dos Modelo de Domínio e Modelo de Use Cases é o *Visual Paradigm*.

## **2. Objetivos**

A primeira fase do projeto tem como objetivo efetuar a análise de requisitos a partir de uma modelação do sistema baseada num Modelo de Domínio e num Modelo de Use Case.

Com o primeiro modelo referido, pode-se construir um diagrama caracterizado pelas entidades relevantes do sistema e pelas associações entre si.

Já o Modelo de Use Cases tem como objetivo a criação de um diagrama em que seja possível ter uma representação esquemática da interação entre os atores e o sistema, com as funcionalidades propostas para o sistema.

### **3. Descrição**

No enunciado do trabalho é nos proposto criarmos um sistema que permita a criação de campeonatos automobilísticos, com vários componentes. O sistema que vai ser construído deverá assegurar o processo desde a criação de corridas, carros, circuitos, utilizadores, etc, até a termos toda constituição de campeonatos e podermos realizar vários desses mesmos.

Deu-se início ao projeto fazendo uma análise elaborada do sistema em questão e aos requisitos que precisamos de seguir. De uma maneira a construir um sistema vigoroso, aplicamos técnicas de modelação com os cenários que nos foram fornecidos.

Durante o desenvolvimento da aplicação na modelação do sistema, geramos Modelo de Domínio e Modelo de Use Cases. O modelo do domínio vai representar as diferentes entidades integradas no Sistema e as relações entre essas. Já o Modelo de Use Case representa as associações entre os atores e o sistema (e entidades fora do sistema), os use cases são ações atômicas que usam serviços e várias funcionalidades do Sistema.

## 4. Modelo de Domínio

Após análise do enunciado proposto e, tendo em conta os requisitos do sistema, deu-se início ao desenvolvimento do sistema a nível do domínio. Tendo em conta os cenários de utilização apresentados considerou-se que as entidades relevantes do sistema são:

- Campeonato
- Circuito
- Corrida
- Jogador
- *Guest*
- Administrador
- Piloto
- Carro

Para além disto, também foram reconhecidas algumas associações entre as diversas entidades combinando com a multiplicidade em causa. Com o propósito de reduzir o nível de complexidade do sistema foram adicionadas mais informações a este modelo de domínio no que toca às especificações do tipo de carro, tipo de piloto e características do circuito.

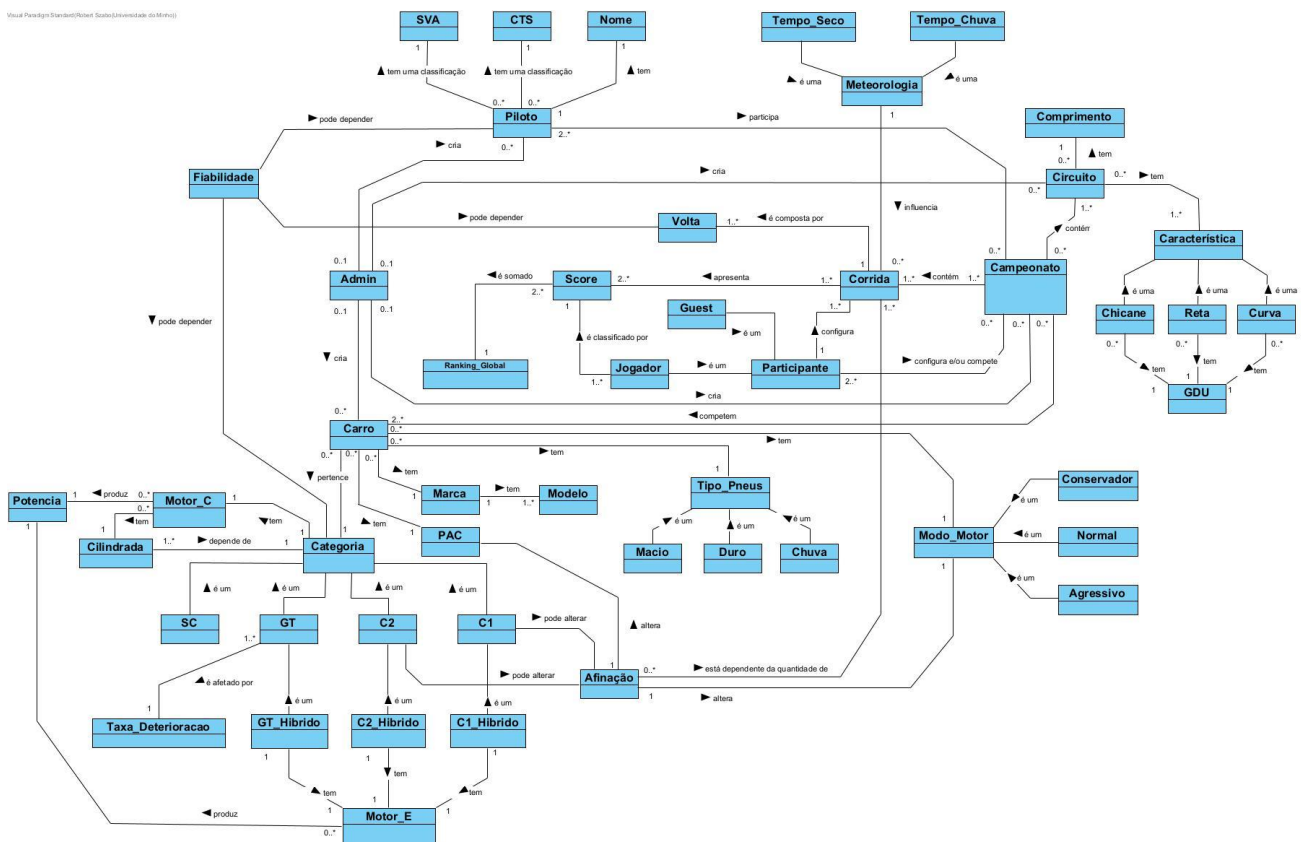


Figura 1 – Modelo de Domínio

## 5. Modelo de Use Case

Com base no Modelo de Domínio, torna-se mais simples a construção do Modelo de Use Case. Tendo em conta, novamente, os cenários propostos procedeu-se à especificação de cada Use Case necessário para a implementação do sistema, ou seja, cada ação possível por parte de um utilizador.

Como primeiro passo podemos definir os atores:

- Administrador
- Jogador
- *Guest*

São estas as únicas figuras capazes de interagir com o sistema podendo ser generalizadas por um utilizador. De modo a reduzir o nível de complexidade, agrupou-se os atores jogador e *guest* como participantes, pois terão o mesmo tipo de atividade, à exceção da criação de conta.

Também podemos especificar os seguintes Use Cases:

- Criar campeonato
- Criar circuito
- Criar carro
- Criar piloto
- Simular corridas
- Configurar campeonato
- Configurar corridas
- Criar conta
- Fazer login
- Consultar resultado final

Numa primeira fase, foram identificados os Use Case associados a cada um dos cenários apresentados. De seguida, estes mesmos cenários foram analisados a um nível mais profundo e percebeu-se a necessidade da criação de mais Use Case de modo a seguir com os objetivos propostos. A especificação de Use Case pode ser encontrada nos Anexos.

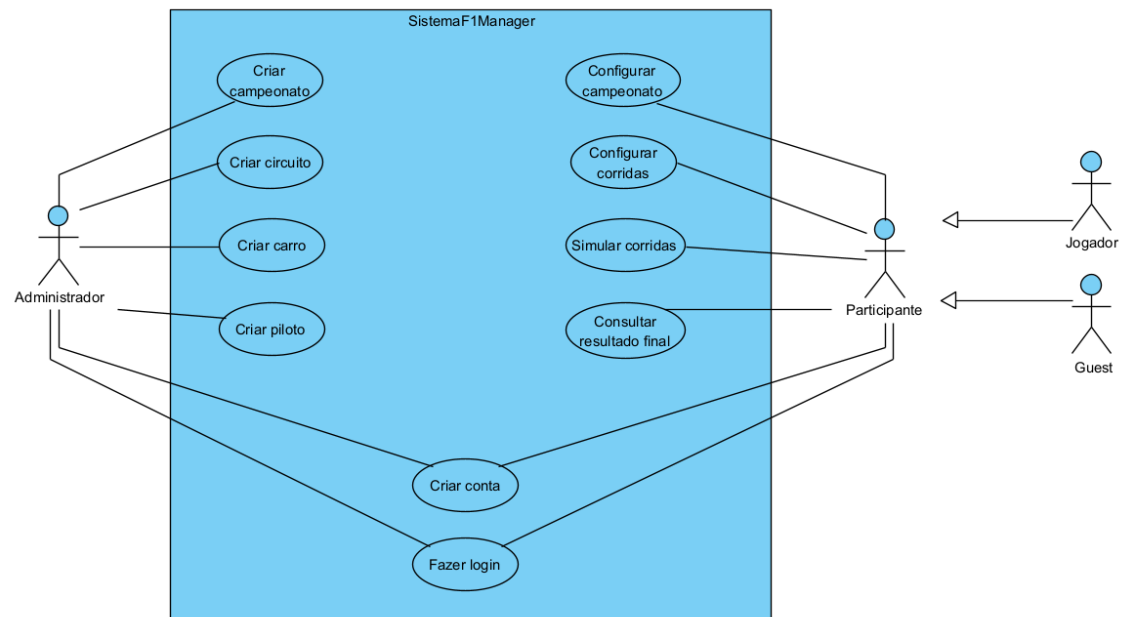


Figura 2 – Modelo de Use Case



## 6. Conclusão

Ao seguir a metodologia apresentada nas aulas compreendemos melhor o funcionamento do sistema que nos foi descrito no enunciado. Acreditamos ter conseguido um bom modelo de domínio, pelo que, certamente, qualquer relação entre entidades no mesmo será facilmente interpretada.

Tentamos clarificar, também, as funcionalidades que o sistema suporta para os diferentes tipos de utilizadores e os comportamentos possíveis no uso do sistema. Concordamos que a análise de requisitos do sistema é uma das etapas mais importantes no desenvolvimento de sistemas de software, uma vez que é esta que define o esqueleto do problema em questão.

## 7. Anexos

### 7.1. Use Case

#### 7.1.1 Criar Conta

**Descrição:** O utilizador cria uma conta.

**Cenários:** O Francisco cria uma conta de jogador. O José cria uma conta de administrador.

**Pré-condição:** True

**Pós-condição:** O sistema regista a conta.

**Fluxo Normal:**

1. O sistema pede um nome de utilizador;
2. O ator introduz um nome de utilizador;
3. O sistema pede uma palavra-passe;
4. O ator introduz uma palavra-passe;
5. O ator escolhe o modo jogador;
6. O sistema autentica a conta.
7. O sistema regista a conta.

**Fluxo Alternativo (1) [o ator escolhe o modo de administrador] (passo 5):**

- 5.1. O ator cria uma conta como administrador;
- 5.2. Regressa a 6.

**Fluxo de Exceção (2) [o sistema não valida a conta] (passo 6):**

- 6.1. O sistema informa que as credenciais não são válidas;
- 6.2. O sistema termina o processo.

#### 7.1.2 Fazer Login

**Descrição:** Utilizador faz *login*.

**Cenários:** O José faz *login* como jogador. O Francisco faz *login* como administrador.

**Pré-condição:** True

**Pós-condição:** É feito login numa conta de administrador ou jogador.

**Fluxo Normal:**

1. Sistema pede um nome de utilizador;
2. Ator introduz um nome de utilizador;

3. Sistema pede uma palavra-passe;
4. Ator introduz uma palavra-passe;
5. Sistema valida o nome de utilizador e a palavra-passe.
6. Sistema indica o tipo de conta e confirma a mesma.

**Fluxo de Exceção (1) [as credenciais não são válidas] (passo 5):**

- 5.1. O sistema indica que as credenciais não são válidas;
- 5.2. O sistema termina o processo.

### 7.1.3 Criar Campeonato

**Descrição:** O administrador cria um campeonato.

**Cenários:** O José faz login no jogo como administrador e adiciona um campeonato à lista de campeonatos disponíveis. Começa por lhe dar o nome “CampeUMnato”. De seguida, escolhe os circuitos “Gualtar Campus”, “Azurém Campus” e “Circuito Sameiro-Bom Jesus” da lista de circuitos disponíveis. Depois de consultar a lista de campeonatos atualmente disponíveis para jogar, decide adicionar o “CampeUMnato” à mesma, pelo que fica imediatamente disponível para ser jogado.

**Pré-condição:** O *login* como administrador é autenticado.

**Pós-condição:** O campeonato é adicionado à lista de campeonatos.

**Fluxo Normal:**

1. O administrador fornece um nome para o campeonato;
2. O administrador escolhe os circuitos da lista de circuitos;
3. O sistema adiciona o campeonato à lista de campeonatos.

### 7.1.4 Criar Circuito

**Descrição:** O administrador cria um circuito.

**Cenários:** O José faz login no jogo como administrador e opta por adicionar um novo circuito. Indica como nome do novo circuito a adicionar “Gualtar Campus”. De seguida, indica que o mesmo tem 2Km, 9 curvas e 1 chicane. Com essa informação, o sistema calcula que o circuito tem 10 retas e apresenta a lista de curvas e retas de modo que o José indique o grau de dificuldade de ultrapassagem (GDU) em cada uma (as chicanes têm sempre um GDU de difícil). Para as retas 1 e 6, e curvas 2 e 3, indica um GDU de possível. Para as retas 4, 5, 7 e 8, e curvas 4, 5, 7 e 8, indica um GDU de impossível. Para os restantes, indica um GDU de difícil. Finalmente, regista o circuito, indicando que cada corrida deverá ter 10 voltas. O circuito passa a estar disponível para integrar campeonatos.

**Pré-condição:** O *login* como administrador é autenticado.

**Pós-condição:** O circuito é adicionado à lista de circuitos.

**Fluxo Normal:**

1. O administrador atribui um nome ao circuito;
2. O administrador indica o valor da distância e o número de curvas e de chicanes;
3. O sistema calcula o número de retas;
4. O sistema apresenta a lista de curvas e retas;
5. O administrador atribui um Grau de Dificuldade de Ultrapassagem a cada elemento da lista;
6. O administrador indica o número de voltas da corrida;
7. O sistema adiciona o circuito à lista de circuitos.

## 7.1.5 Criar Carro

**Descrição:** O administrador cria um carro.

**Cenários:** O José cria um carro híbrido da categoria C1.

**Pré-condição:** O *login* como administrador é autenticado.

**Pós-condição:** O carro é adicionado à lista de carros.

**Fluxo Normal:**

1. Administrador escolhe categoria, marca, modelo, cilindrada e potência do motor de combustão;
2. Sistema verifica que é um carro C1 e indica que necessita de uma fiabilidade;
3. Ator indica fiabilidade (C1 é o único que pede fiabilidade);
4. Sistema verifica que a fiabilidade é aproximadamente 95%;
5. Sistema pergunta se o carro é híbrido;
6. Ator indica que carro não é híbrido;
7. O administrador escolhe o Perfil Aerodinâmico do Carro;
8. O sistema regista o carro.

**Fluxo Alternativo (1) [o carro é C2] (passo 2):**

- 2.1. O sistema verifica que o carro da categoria C2;
- 2.2. Regressa a 5.

**Fluxo Alternativo (2) [o carro é GT] (passo 2):**

- 2.1. O sistema verifica que o carro é da categoria GT;
- 2.2. Sistema pede a taxa de degradação;
- 2.3. Ator indica a taxa de degradação;

2.4. Regressa a 5.

**Fluxo Alternativo (3) [o carro é SC] (passo 2):**

- 2.1. O sistema verifica que o carro é da categoria SC;
- 2.2. Regressa a 7.

**Fluxo Alternativo (4) [o ator diz que o carro é híbrido] (passo 6):**

- 6.1. Sistema pede a potência do motor elétrico;
- 6.2. Ator define a potência do motor elétrico;
- 6.3. Regressa a 7.

**Fluxo de Exceção (5) [fiabilidade não é aproximadamente 95%] (passo 4):**

- 4.1. O sistema verifica se a viabilidade é demasiado distante de 95%;
- 4.2. O sistema termina o processo;

## 7.1.6 Criar Piloto

**Descrição:** O administrador cria um piloto.

**Cenários:** O José faz login no jogo como administrador e decide adicionar um novo piloto. Começa por indicar que o nome será “Battery Voltar” 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 arriscar pouco (logo terá alguma maior dificuldade em ultrapassar, mas menor probabilidade de se despistar). Finalmente, regista o piloto, que fica disponível.

**Pré-condição:** O *login* como administrador é autenticado.

**Pós-condição:** O piloto fica disponível.

**Fluxo Normal:**

1. O administrador indica o nome do piloto;
2. O administrador indica um valor para o critério “Chuva VS Tempo Seco” (CTS);
3. O administrador indica um valor para o critério “Segurança VS Agressividade” (SVA);
4. O sistema regista o piloto, que fica disponível.

## 7.1.7 Configurar Campeonato

**Descrição:** Num determinado campeonato, cada participante escolhe um carro e um piloto.

**Cenários:** O Francisco e três amigos resolveram jogar um campeonato de *Racing Manager*. O Francisco faz *login* como jogador, escolhe um campeonato e avalia os circuitos que o compõem. Como a maioria são circuitos rápidos, decide participar com um Ferrari 488 GTE (um carro da categoria C2). Escolheu como piloto Battery Voltas, por considerar ser um piloto equilibrado em termos de desempenho. Após inscrever-se, cada um dos amigos escolhe também o seu carro e piloto.

**Pré-condição:** Há carros e pilotos disponíveis nas respectivas listas e, pelo menos, dois participantes.

**Pós-condição:** O campeonato é assumido pelo sistema.

**Fluxo Normal:**

1. O sistema apresenta a lista de campeonatos;
2. O participante seleciona um campeonato;
3. O sistema apresenta as especificações do campeonato selecionado;
4. O sistema apresenta a lista de carros;
5. Cada participante escolhe um carro;
6. O sistema apresenta a lista de pilotos;
7. Cada participante escolhe um piloto;
8. O sistema regista os participantes.

### 7.1.8 Configurar Corrida

**Descrição:** Cada participante faz as preparações finais do carro.

**Cenários:** Quando todos estão registados, dá-se início ao campeonato. As condições da primeira corrida são apresentadas: o circuito é o “Gualtar Campus” e a situação meteorológica é de tempo seco (a outra possibilidade seria chuva). 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 (possível por se tratar de um C2) e aumenta a downforce de 0.5 (valor neutro) para 0.7. Ao aumentar a downforce, sacrifica alguma velocidade para ter maior estabilidade em curva. Deste modo, troca alguma capacidade de ultrapassar em reta por capacidade de ultrapassar em curva, compensando a menor propensão para o risco do piloto. Finalmente, todos os jogadores devem escolher os pneus e modo do motor a usar na corrida. Dos três tipos de pneu disponíveis, neste momento, no jogo (macio, duro e chuva) o Francisco escolhe pneus macios, o que permite ter melhor desempenho no início da corrida, à custa do desempenho no final. Dos três modos de funcionamento do motor (conservador, normal ou agressivo), o Francisco escolhe o agressivo, aumentando o desempenho do carro à custa de maior probabilidade de o motor ter uma avaria.

**Pré-condição:** Todos os participantes estão registados.

**Pós-condição:** A corrida é configurada de acordo com as escolhas de cada participante.

**Fluxo Normal:**

1. O circuito e a situação meteorológica são apresentados;
2. O sistema pergunta se quer fazer alguma afinação;
3. O participante decide não afinar o carro;
4. Cada participante escolhe o tipo de pneus;

**Fluxo Alternativo (1) [o participante decide afinar o carro] (passo 3):**

- 3.1. O sistema verifica se é possível fazer uma afinação;
- 3.2. O participante define o novo valor de *downforce*;
- 3.3. Cada participante escolhe o modo do motor.
- 3.4. Regressa a 4.

**Fluxo Alternativo (2) [não é possível fazer uma afinação] (passo 3.1):**

- 3.1.1. O sistema informa que não é possível fazer uma afinação;
- 3.1.2. Regressa a 4.

## 7.1.9 Simular Corrida

**Descrição:** Cada corrida de um determinado campeonato é simulada.

**Cenários:** Quando todos têm os carros prontos, a corrida começa. A partir desse momento, o *Racing Manager* simula a corrida. A simulação tem em conta as características do circuito, dos carros e dos pilotos e vai indicando, para cada curva, reta e chicane de cada volta, eventuais ultrapassagens, despistes e avarias. No final de cada volta são indicadas as posições dos carros/pilotos/jogadores. Uma vez que o Francisco tem a versão base do jogo, os cálculos são feitos em função das posições relativas dos carros. Em cada ponto relevante do circuito (reta/curva/chicane), o simulador decide se cada carro consegue, ou não, ultrapassar o carro que se encontra à sua frente, e no final da volta é indicada a ordem dos carros.

A Sara optou por um carro da categoria C1, híbrido, com *downforce* mínima, modo de motor agressivo, pneus macios e um piloto com SVA alto. Assim, durante a primeira volta consegue fazer uma ultrapassagem na curva 74. No entanto, na sétima volta, ao tentar uma ultrapassagem na chicane, acaba por sair de pista e ficar na última posição. Os pneus já não estavam em bom estado e acabou por não conseguir ultrapassar ninguém até ao final da corrida. Terminou em terceiro lugar pois o Manuel despistou-se na curva 1 durante a última volta. Nesta corrida nenhum carro sofreu uma avaria.

**Pré-condição:** Cada corrida do campeonato está configurada.

**Pós-condição:** O campeonato é disputado e as pontuações do mesmo são atualizadas.

**Fluxo Normal:**

1. O sistema confirma que pelo menos um jogador tem a versão base do jogo;
2. O sistema calcula para cada curva, reta e chicane eventuais ultrapassagens, despistes e avarias a partir da posição;
3. O sistema apresenta o resultado da volta;
4. O sistema verifica se há mais voltas;
5. O sistema apresenta o resultado da corrida, por categoria;
6. O sistema atualiza e apresenta os pontos da classificação do campeonato de cada participante;
7. O sistema verifica se há mais corridas a serem disputadas;
8. O campeonato termina.

**Fluxo Alternativo (1) [todos os jogadores têm a versão *premium* do jogo] (passo 1):**

- 1.1. O sistema calcula para cada curva, reta e chicane eventuais ultrapassagens, despistes e avarias a partir das diferenças de tempo entre os carros;
- 1.2. Regressa a 3.

**Fluxo Alternativo (2) [um participante despista-se ou tem uma avaria] (passo 2):**

- 2.1. O sistema indica que o participante teve um despiste ou avaria;
- 2.2. O participante é “retirado” da corrida;
- 2.3. O sistema atribui uma classificação “nula” ao participante;
- 2.4. Regressa a 3.

**Fluxo Alternativo (3) [nº de voltas realizadas é < nº total de voltas] (passo 4):**

- 4.1. Regressa a 1.

**Fluxo Alternativo (4) [nº de corridas realizadas é < nº total de corridas] (passo 7):**

- 7.1. Regressa a 1.

## **7.1.10 Consultar Resultado Final**

**Descrição:** O sistema atualiza e apresenta a pontuação global do *ranking* do *Racing Manager*.

**Cenários:** 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 faz login 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:** Concluir um campeonato.



**Pós-condição:** A classificação global do *ranking* do *Racing Manager* é apresentada.

**Fluxo Normal:**

1. O sistema pergunta a cada participante se quer fazer o *login*;
2. O participante decide fazer o *login*.
3. O sistema soma os pontos do campeonato de cada participante à classificação global do *Racing Manager*.
4. O sistema atualiza as posições da classificação do *Racing Manager*.
5. O sistema apresenta a classificação global do *ranking* do *Racing Manager*.

**Fluxo Alternativo (1) [o participante não faz *login*] (passo 2):**

- 2.1. O sistema informa que a pontuação feita durante o campeonato não será considerada na classificação global do *Racing Manager*;
- 2.2. Regressa a 3.