

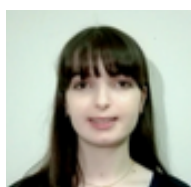


UNIVERSIDADE DO MINHO

DESENVOLVIMENTO DE SISTEMAS DE SOFTWARE LICENCIATURA EM ENGENHARIA INFORMÁTICA

ENTREGA INTERMÉDIA 1

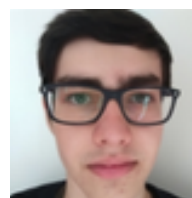
GRUPO 30



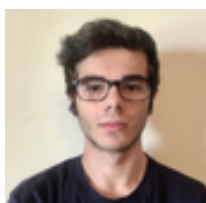
Ana Rita Poças
(a97284)



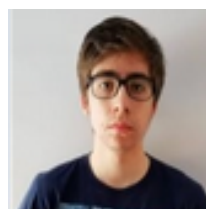
João Pedro Braga
(a97368)



Miguel Pinto
(a96106)



Orlando Palmeira
(a97755)



Pedro Miguel Martins
(a97613)

URL do repositório:

<https://github.com/orlandopalmeira/Trabalho-DSS-2022-2023>

ÍNDICE

1.	Introdução	1
2.	Modelo de Domínio	2
3.	Modelo de Use Case	4
3.1.	Criar conta	5
3.2.	Fazer Login de acesso	6
3.3.	Criar Campeonato	7
3.4.	Criar Pilotos	8
3.5.	Criar Circuitos	9
3.6.	Criar carro	10
3.7.	Configurar Campeonato	12
3.8.	Jogar Corrida	13
4.	Conclusão e Análise dos Resultados Obtidos	14

Introdução

Este relatório foi elaborado no âmbito da unidade curricular de Desenvolvimento de Sistemas de Software no qual nos foi proposta a elaboração de um sistema que permita simular campeonatos de automobilismo, algo similar a um F1 Manager.

Nesta primeira fase, é nos pedido a modelação do problema de modo a documentar as decisões tomadas, bem como ajudar a descrever os componentes do problema, facilitando a compreensão do nosso sistema.

Foram desenvolvidos dois modelos, o **Modelo de Domínio** que descreve as entidades do problema e os seus relacionamentos sendo uma visão estática e representativa do problema, e o **Modelo de Use Case** que representa as funcionalidades do sistema em que estão descritos os fluxos de interação com o sistema.

Modelos de Domínio

Um modelo de Domínio bem concebido deve representar o problema proposto, sem apresentar soluções para a sua resolução.

Como tal, o modelo desenvolvido pelo nosso grupo (figura 1) tem como objetivo representar o problema que nos foi apresentado, englobando as devidas entidades nele envolvidas, bem como os relacionamentos que existem entre si.

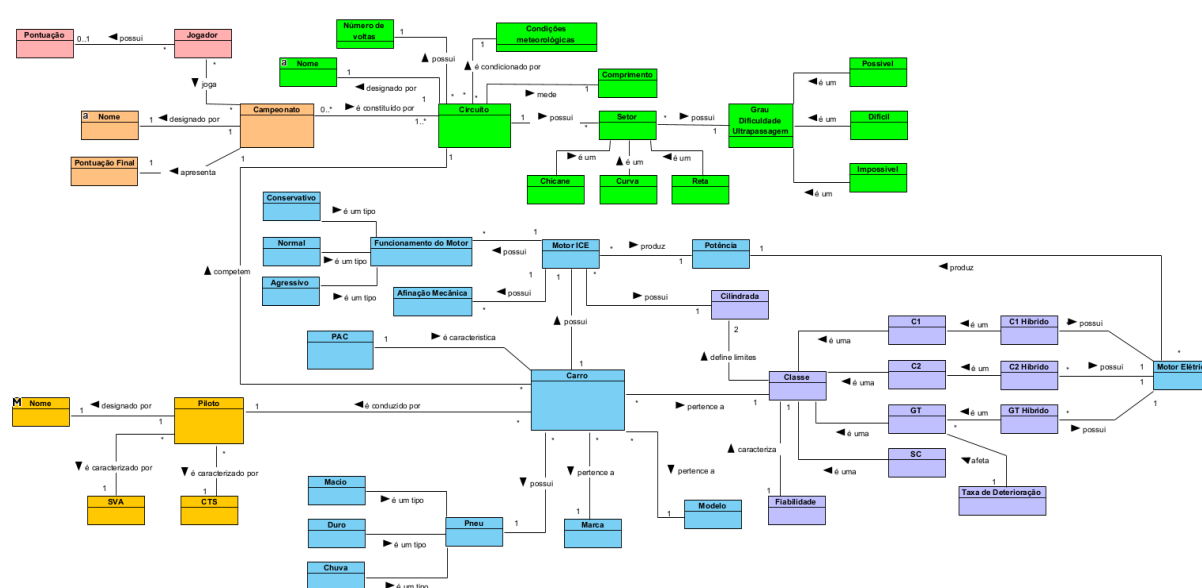


Figura 1 - Modelo de Domínio do nosso projeto

Através da leitura do enunciado identificamos as seguintes entidades principais:

- **Campeonato**
- **Circuito**
- **Carro**
- **Piloto**
- **Classe**

Categorizamos as diferentes secções com cores distintas, de maneira a ser mais perceptível os relacionamentos que envolvem as entidades principais:

- Um **Campeonato** é designado por um **nome** e é jogado por um **jogador**.
- Um **Circuito** é designado por um **nome**, possui um **comprimento**, contém vários **setores** (reta, curva, chicane), possui um **número de voltas** e é condicionado por uma **condição meteorológica**.

- Um **Carro** pertence a uma certa **classe**, **modelo** e a uma **marca**, possui um determinado tipo de **pneus**, possui um **motor ICE**, é caracterizado por uma **“PAC”** é conduzido por um **piloto** e compete num **circuito**.
- Um **Piloto** é designado por um **nome**, um critério **“SVA”** (Segurança vs Agressividade) e um critério **“CTS”** (Chuva vs Tempo Seco).
- Uma **Classe** é uma entidade que contém vários tipos (**C1, C2, GT, SC**), delimita uma **cilindrada** e é caracterizada por uma **fiabilidade**.

Modelo de Use Case

Enquanto que o modelo de domínio tem como objetivo representar o problema proposto, o modelo de Use Case tem como finalidade definir a solução do problema do sistema a desenvolver.

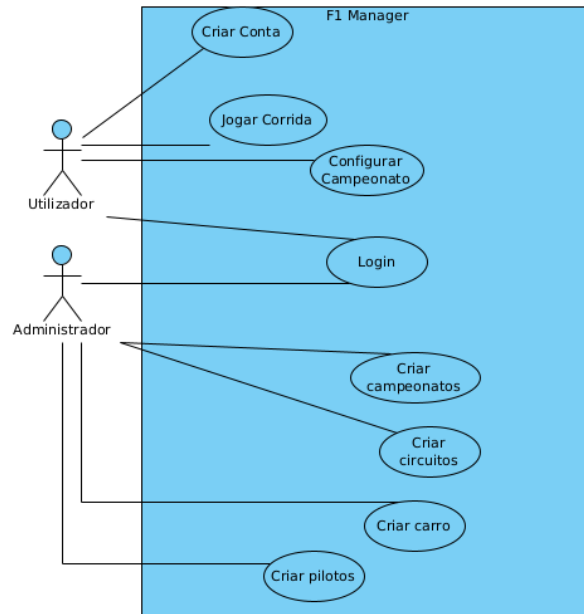


Figura 2 - Diagrama de Use Case

1. Criar Conta

Use case: Criar conta

Descrição: Adição de uma nova conta no sistema

Cenários: O João pretende jogar, para tal deve criar uma conta

Pré-condição: True

Pós-condição: Uma nova conta é adicionada ao sistema

Fluxo normal (1):

1. O sistema solicita um username.
2. O João fornece o username.
3. O sistema verifica que o username não existe.
4. O sistema solicita uma password.
5. O utilizador fornece uma password.
6. O sistema cria uma nova conta com este username e password.

Fluxo de exceção (2) [O username já existe] (passo 3):

- 3.1 O sistema informa que o username não se encontra disponível.
- 3.2 A conta não é adicionada ao sistema.

2. Login

Use case: Login

Descrição: Entrada de um utilizador no sistema.

Cenários: O Guilherme entra como jogador. O Orlando entra como administrador.

Pré-condição: True

Pós-condição: O utilizador consegue acesso ao sistema.

Fluxo normal (1):

1. O sistema solicita o username e password.
2. O utilizador apresenta o seu username e password.
3. O sistema verifica que é um utilizador válido (que existe).
4. O sistema verifica que a password está correta.
5. O sistema concede acesso ao utilizador.

Fluxo de exceção (2) [Utilizador inválido (inexistente)] (passo 3):

- 3.1. O sistema indica que o utilizador fornecido não existe.
- 3.2. O sistema não garante acesso ao utilizador.

Fluxo de exceção (3) [Password errada] (passo 4):

- 4.1. O sistema indica que a password fornecida não é válida.
- 4.2. O sistema não garante acesso ao utilizador.

3. Criar Campeonato

Use case: Criar campeonatos

Descrição: Um administrador adiciona um campeonato ao sistema.

Cenários: A Maria faz login como administradora e adiciona o campeonato “WSeries” à lista de campeonatos disponíveis.

Pré-condição: O utilizador está autenticado e é administrador e haver circuitos disponíveis.

Pós-condição: O campeonato é adicionado ao sistema.

Fluxo normal (1):

1. O sistema pede o nome do novo campeonato.
2. O utilizador fornece o nome.
3. O sistema verifica que o nome é válido (não existe no sistema).
4. O sistema apresenta os circuitos disponíveis e solicita quais é que o utilizador pretende adicionar ao campeonato.
5. O utilizador indica os circuitos pretendidos.
6. O campeonato é adicionado ao sistema.

Fluxo de exceção (2) [O nome do campeonato é inválido] (passo 3):

- 3.1. O sistema indica que o nome do campeonato é inválido.
- 3.2. O campeonato não é adicionado ao sistema.

4. Criar Pilotos

Use case: Criar pilotos

Descrição: O administrador adiciona um novo piloto ao sistema.

Cenários: A Rosa faz login como administradora e decide adicionar um novo piloto ao sistema.

Pré-condição: O utilizador está autenticado e é administrador.

Pós-condição: O piloto é adicionado ao sistema.

Fluxo normal (1):

1. O sistema pede o nome do novo piloto.
2. O utilizador fornece o nome.
3. O sistema verifica que o nome é válido (não existe no sistema).
4. O sistema solicita o valor do critério "Chuva vs. Tempo Seco" (CTS).
5. O utilizador indica o valor do CTS.
6. O sistema solicita o valor do critério "Segurança vs. Agressividade" (SVA).
7. O utilizador indica o valor da SVA.
8. O piloto fica registado no sistema.

Fluxo de exceção (2) [O nome do piloto é inválido] (passo 3):

- 3.1. O sistema indica que o nome do piloto é inválido.
- 3.2. O piloto não é adicionado ao sistema.

5. Criar Circuitos

Use case: Criar circuitos

Descrição: Um administrador adiciona um circuito novo ao sistema.

Cenários: O Henrique faz login como administrador e opta por adicionar um novo circuito.

Pré-condição: O utilizador está autenticado e é administrador.

Pós-condição: O circuito é adicionado ao sistema.

Fluxo normal (1):

1. O sistema pede o nome do novo circuito
2. O utilizador fornece o nome do novo circuito.
3. O sistema verifica que o nome fornecido é válido (não é repetido).
4. O sistema solicita informações do circuito (comprimento, retas, curvas e chicanes).
5. O utilizador fornece essas mesmas informações.
6. O sistema solicita os graus de dificuldade de cada reta e curva.
7. O utilizador fornece essas mesmas informações.
8. O sistema solicita o número de voltas que terão de ser dadas no circuito.
9. O utilizador fornece o número de voltas.
10. O sistema guarda o novo circuito.

Fluxo de exceção (2) [Nome do circuito inválido (repetido)] (passo 3):

- 3.1. O sistema indica que o nome do circuito é inválido.
- 3.2. O circuito não é adicionado ao sistema.

6. Criar carro

Use case: Criar carro

Descrição: Um administrador adiciona um novo carro ao sistema

Cenários: O Gabriel pretende adicionar um novo carro da Classe 1, da marca Ferrari 125 F1.

Pré-condição: O utilizador está autenticado e é administrador.

Pós-condição: Um novo carro é adicionado ao sistema

Fluxo normal (1):

1. O sistema apresenta as Classes disponíveis.
2. O utilizador seleciona a classe, marca, modelo, cilindrada e potência que pretende.
3. O sistema verifica que a cilindrada corresponde à classe do veículo.
4. O sistema verifica que o carro é C1 e necessita de fiabilidade (e pode ser híbrido).
5. O utilizador indica fiabilidade.
6. O sistema verifica que a fiabilidade é aproximadamente 95%.
7. O utilizador indica que o carro não é híbrido.
8. O utilizador indica PAC.
9. O sistema verifica que este carro não existe no sistema.
10. O sistema guarda o novo carro.

Fluxo alternativo (2) [carro é SC] (passo 4):

- 4.1. Sistema verifica que carro é SC.
- 4.2. Regressa a 7.

Fluxo alternativo (3) [carro é C2] (passo 4):

- 4.1. Sistema verifica que o carro é C2.
- 4.2. Regressa a 6.

Fluxo alternativo (4) [carro é GT] (passo 4):

- 4.2. O utilizador indica a taxa de deterioração.
- 4.3. Regressa a 6.

Fluxo alternativo (5) [carro é híbrido] (passo 7):

- 7.1. Utilizador indica que o carro é híbrido e indica potência do motor elétrico.
- 7.2. Regressa a 7.

Fluxo de exceção (6) [O carro já existe no sistema] (passo 9):

- 9.1. O sistema indica que o carro já se encontra no sistema, pelo que não pode ser criado.
- 9.2. O carro não é adicionado ao sistema.

Fluxo de exceção (7) [A cilindrada não corresponde ao intervalo da Classe selecionada] (passo 3):

- 3.1. O sistema indica que a cilindrada não corresponde ao intervalo da Classe que selecionou.
- 3.2. O carro não é adicionado ao sistema.

Fluxo de exceção (8) [O Sistema verifica que a fiabilidade não é aproximadamente 95%] (passo 6):

- 6.1. O sistema indica que a fiabilidade não corresponde ao intervalo permitido.
- 6.2. O carro não é guardado no sistema.

7. Configurar Campeonato

Use Case: Configurar campeonato

Descrição: O jogador vai escolher que campeonato quer jogar, bem como o piloto e o seu carro.

Cenários: O Francisco escolhe um campeonato e avalia os circuitos que o compõem. Decide participar com um Ferrari 488 GTE, escolhendo como piloto Battery Voltas.

Pré-condição: O jogador fez login e não é administrador.

Pós-condição: O campeonato encontra-se pronto para jogar de acordo com as especificações do jogador.

Fluxo normal (1):

1. O sistema apresenta lista de campeonatos disponíveis para jogar.
2. O jogador escolhe o campeonato que pretende jogar.
3. O sistema solicita o número de jogadores para o jogo (indicando que no mínimo devem ser 2).
4. O jogador seleciona o número.
5. O sistema pergunta ao jogador se quer fazer login ou se quer concorrer anonimamente.
6. O sistema apresenta a lista de carros disponíveis.
7. O jogador seleciona o carro com que pretende jogar.
8. O sistema apresenta a lista de pilotos disponíveis.
9. O jogador seleciona o piloto com que pretende jogar.
10. O sistema adiciona este jogador.
11. Caso haja mais jogadores, o sistema regressa ao ponto 5.
12. O sistema regista todos os jogadores que vão participar.

Fluxo de exceção (2) [O número de jogadores não corresponde ao mínimo para haver um jogo] (passo 4):

- 4.1. O sistema indica que o número não corresponde ao mínimo.
- 4.2. O campeonato não fica pronto para jogar.

8. Jogar Corrida

Use Case: Jogar Corrida

Descrição: O jogador configura o seu carro para jogar a corrida.

Cenário: O Francisco entra numa corrida, decide alterar a afinação e aumenta a downforce de 0.5 para 0.7 e encontra-se pronto para correr.

Pré-Condição: Todos os jogadores estão registados no campeonato.

Pós-Condição: A corrida termina e são apresentados os resultados.

Fluxo Normal (1):

1. O sistema apresenta as condições da primeira corrida.
2. O sistema pergunta aos jogadores com carros da classe C1 e C2 se pretendem alterar a afinação do seu carro, caso o limite ainda não tenha sido excedido.
3. Cada um dos jogadores seleciona as afinações que pretende.
4. O sistema solicita a cada um dos jogadores dos os pneus desejados.
5. Os jogadores selecionam os pneus que pretendem.
6. O sistema solicita a cada um dos jogadores das classes C1 e C2 o modo do motor que pretendem usar na corrida.
7. Os jogadores selecionam o modo do motor desejado.
8. Os carros encontram-se prontos para a corrida.
9. Estando os carros prontos, a corrida começa e o sistema simula a corrida.
10. No final da corrida, são apresentados os resultados.

Fluxo Alternativo (2) [O número de afinações possíveis ultrapassou o limite] (passo 2):

- 2.1. O sistema indica ao jogador específico que não é possível alterar a afinação do seu carro, uma vez que excedeu o limite.
- 2.2. Regressa a 4.

Conclusão e Análise dos Resultados Obtidos

Uma vez elaborado o Modelo de Domínio e o Modelo de Use Case, damos por concluída a fase 1 do projeto que nos foi proposto.

Em primeiro lugar, o Modelo de Domínio que realizamos vai ao encontro com a interpretação que fizemos, abrangendo todos os cenários que contemplamos. Através dele somos capazes de obter uma representação visual das entidades associadas ao problema que nos foi apresentado.

Com o nosso Modelo de Use Case, descrevemos as funcionalidades que o nosso sistema irá suportar, representando os requisitos funcionais que levantamos a partir do enunciado fornecido.

Em suma, através desta esquematização do nosso problema, as etapas futuras de desenvolvimento do sistema poderão ser realizadas de uma maneira mais metódica e organizada, otimizando assim a sua qualidade e futura interpretação.