







Universidade do Minho
Escola de Engenharia
Licenciatura em Engenharia informática

Unidade Curricular de Desenvolvimento de Sistemas Software

Ano Letivo de 2022/2023

Trabalho Prático – Entrega Intermédia I

<p>A96547 <i>Rodrigo José Teixeira Freitas</i></p>		<p>A95076 <i>Pedro Marcelo Bogas Oliveira</i></p>	
<p>A92974 <i>José dos Santos Mendes</i></p>		<p>A93186 <i>Eduardo Fernando Cruz Henriques</i></p>	

Índice

Índice	1
Introdução	2
Modelo de Domínio	2
Modelo de Use Case	3
1. Use Case 1	4
2. Use Case 2	4
3. Use Case 3	5
4. Use Case 4	6
5. Use Case 5	7
6. Use Case 6	9
7. Use Case 7	9
Conclusão	10

Uma vez que os carros são definidos pelas suas características (por exemplo: marca, modelo e perfil aerodinâmico), decidimos reparti-los por categorias sendo possível especificar a sua classe bem como a existência de um motor elétrico e a dedução da sua cilindrada.

Sendo que cada utilizador da aplicação participa com um piloto, estes foram também descritos sendo caracterizados pelo seu nome, agressividade e a sua performance durante uma corrida.

Consequentemente, e tratando-se de um jogo de corridas é obrigatório a existência de circuitos destacando-se o nome, número de voltas, retas, curvas, chicanes e o grau de dificuldade dos mesmos. Posteriormente à criação dos circuitos, a implementação campeonatos é composta pelo seu nome, bem como, uma lista de circuitos.

Por fim, foi tido em conta a simulação da corrida onde é descrita a meteorologia da corrida e encontra-se ligada às classes mais importantes (piloto, campeonato, circuito e carro), uma vez que necessita destas informações para que seja executada com sucesso.

Modelo de Use Case

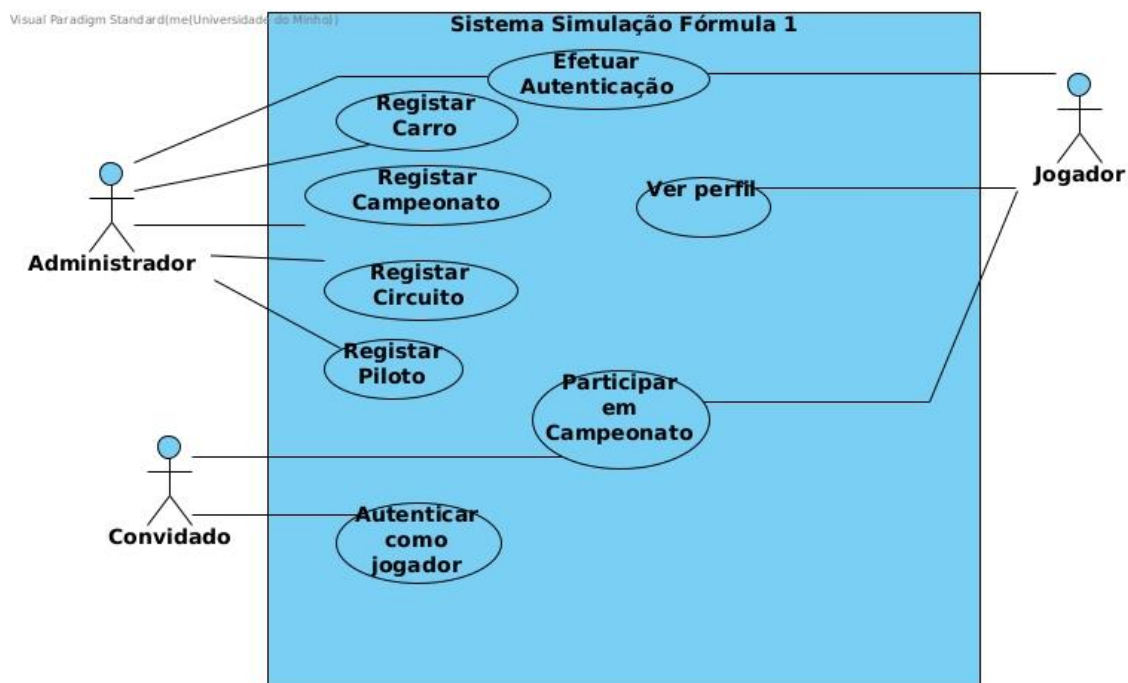


Ilustração 2 - Modelo Use Case

Os Use Case's abaixo representam os cenários descritos no enunciado, bem como, alguns mais que consideramos ser relevantes tendo em conta a nossa análise de requisitos do projeto.

1. Use Case 1

Use Case: Criação de Campeonato.

Descrição: O Ator (Administrador) cria um campeonato que ficará guardado no sistema para uso futuro.

Cenário: Cenário 1 (Enunciado).

Pré-Condição: Administrador está autenticado.

Pós-Condição: Campeonato é criado com sucesso e armazenado no sistema.

Fluxo normal:

1. Administrador escreve o nome do campeonato.
2. Sistema consulta a lista de campeonatos disponíveis para verificar se o nome está disponível.
3. Administrador seleciona pelo menos dois circuitos da lista de circuitos disponíveis para uso no campeonato.
4. Administrador escolhe acabar a criação do campeonato e adicioná-lo à lista.
5. Sistema guarda o campeonato criado pelo Administrador na lista de campeonatos jogáveis.
6. Sistema volta ao menu inicial.

Fluxo alternativo (1): [Campeonato criado já existe] (passo 2 ou 2.2)

- 2.1 (ou 2.2.1) Administrador escreve outro nome para o campeonato.
- 2.2 (ou 2.2.2) Sistema verifica se o novo nome está disponível.
- 2.3 (ou 2.2.3) Regressar para 3.

Fluxo de exceção (2): [administrador cancela criação do campeonato] (passo 2.1)

- 2.1.1 Sistema confirma anulação da criação do campeonato e descarta informação sobre este.
- 2.1.2 Sistema volta ao menu inicial.

Fluxo de exceção (3): [Administrador cancela criação do campeonato] (passo 4)

- 4.1 Sistema confirma anulação da criação do campeonato e descarta informação sobre este.
- 4.2 Sistema volta ao menu inicial.

2. Use Case 2

Use Case: Criação de um circuito.

Descrição: O Ator (Administrador) cria um circuito que ficará guardado no sistema para uso futuro.

Cenário: Cenário 2 (Enunciado).

Pré-condição: Administrador está autenticado.

Pós-condição: Circuito é criado com sucesso e armazenado no sistema.

Fluxo Normal:

1. Administrador fornece o nome do circuito, distância da pista, número de curvas e de chicanes
2. Sistema calcula o número de retas do circuito e apresenta-as ao ator
3. Administrador fornece ao sistema o grau de dificuldade de ultrapassagem de cada seguimento do circuito
4. Administrador indica o número de voltas a ser executado no circuito
5. Sistema cria o circuito
6. Sistema volta para o menu inicial

Fluxo de exceção (1) [circuito já existe no sistema] (passo 1)

- 1.1. Sistema informa que o circuito já existe
- 1.2. Sistema cancela a criação do circuito
- 1.3. Sistema volta para o menu inicial

Fluxo alternativo (2) [ator não fornece os graus de dificuldade de ultrapassagem] (passo 3)

- 3.1. Sistema atribui valores aleatórios a cada um dos seguimentos
- 3.4. Regressa a 4

3. Use Case 3

Use Case: Criação de um carro.

Descrição: O Ator (Administrador) cria um carro que ficará guardado no sistema para uso futuro.

Cenário: Cenário 3 (Enunciado).

Pré-condição: Administrador está autenticado.

Pós-condição: Carro é criado com sucesso e armazenado no sistema.

Fluxo normal:

1. Administrador indica classe, marca, modelo, cilindrada e potência do carro a criar;
2. Sistema verifica que é um carro C1, logo pode ser híbrido;
3. Sistema verifica que a fiabilidade é aproximadamente 95%;
4. Administrador indica que carro não é híbrido;
5. Administrador indica PAC;
6. Sistema regista carro;

Fluxo de exceção (1) [carro já existe] (passo 1)

- 1.1 Sistema informa que o carro já existe;
- 1.2 Sistema volta para o menu inicial

Fluxo alternativo (2) [carro é SC] (passo2)

- 2.1 Sistema verifica que é um carro SC;
- 2.2 Sistema calcula a fiabilidade e esta depende do piloto e da cilindrada;
- 2.3 Regressa para o ponto 5;

Fluxo alternativo (3) [carro é C2] (passo 2)

- 2.1. Sistema verifica que é um carro C2, logo pode ser híbrido;
- 2.2 Sistema calcula a fiabilidade, que rondará os 80%;
- 2.3 Regressa para o ponto 4;

Fluxo alternativo (4) [carro é GT] (passo 2)

- 2.1 Sistema verifica que é um carro GT, logo pode ser híbrido;
- 2.2 Sistema calcula a fiabilidade e esta diminui com o número de voltas;
- 2.3 Regressa para o ponto 4;

Fluxo alternativo (5) [carro é híbrido] (passo 4)

- 4.1 Administrador indica que é híbrido e indica a potencia do motor elétrico
- 4.2 Regressa para o ponto 6

4. Use Case 4

Use Case: Criação de um piloto.

Descrição: O Ator (Administrador) cria um piloto que ficará guardado no sistema para uso futuro.

Cenário: Cenário 4 (Enunciado).

Pré-condição: Administrador está autenticado.

Pós-condição: Piloto é criado com sucesso e armazenado no sistema.

Fluxo normal:

1. Administrador indica o nome do piloto.
2. Sistema verifica se já existe algum piloto com esse nome.
3. Administrador indica os seus níveis de perícia.
4. Sistema verifica se os valores inseridos são válidos.
5. Administrador escolhe acabar a criação do piloto e adicioná-lo á lista.
6. Sistema cria o piloto.
7. Sistema volta para o menu inicial.

Fluxo de alternativo (1) [Sistema indica que o nome já existe] (passo 2)

- 2.1. Sistema informa que o nome inserido já existe.
- 2.2. Sistema regressa ao ponto 1.

Fluxo alternativo (2) [Os valores inseridos são inválidos] (passo 4)

- 4.1 Sistema informa que os valores inseridos são inválidos.
- 4.2 Regressa para o ponto 3.

Fluxo de Exceção (3): [Administrador cancela criação do piloto] (passo 2.1 ou 5)

- 2.1.1 (ou 5.1) Sistema confirma anulação da criação do piloto e descarta informação sobre este.
- 2.2.2 (ou 5.2) Sistema volta ao menu inicial.

5. Use Case 5

Use case: Jogar um campeonato

Descrição: O ator (jogador) joga um campeonato

Cenário: Cenário 4 (Enunciado).

Pré-condição: Jogador está autenticado.

Pós-condição: O campeonato termina e as pontuações globais são atualizadas

Fluxo normal:

1. Jogador escolhe um campeonato da lista.
2. Jogador espera que os seus amigos se inscrevam no campeonato.
3. Jogadores escolhem um carro da lista.

4. Jogadores escolhem um piloto da lista.
5. Jogador inicia o campeonato.
6. O sistema guarda as informações de todos os jogadores do campeonato.
7. O sistema indica o nome do circuito.
8. O sistema indica as condições da primeira corrida.
9. Os participantes decidem fazer alterações ao carro.
10. O sistema atualiza os valores dos atributos do(s) carro(s).
11. O sistema inicia a simulação das corridas
12. O sistema indica as atualizações e os eventos de cada volta até ao fim da corrida.
13. O sistema anuncia os resultados de cada corrida.
14. O sistema anuncia os resultados do campeonato e guarda as pontuações dos Jogadores.

Fluxo alternativo (1) [Jogador escolhe jogar sozinho] (passo 2)

- 2.1 Regressa para o ponto 3.

Fluxo de Exceção (2) [Jogador cancela criação do campeonato] (passo 5)

- 5.1 Sistema confirma anulação da criação do campeonato e descarta informação sobre este.
- 5.2 Sistema volta ao menu inicial.

Fluxo alternativo (3) [Jogadores decidem não alterar os pré-requisitos do carro] (passo 9)

- 9.1 O sistema atribui os valores padrão.
- 9.2 Regressa para o ponto 11.

Fluxo alternativo (4) [Pelo menos um dos participantes tem a versão Premium] (passo 12)

- 12.1 A simulação é executada com critérios diferentes e indica a diferença de tempo entre os carros.
- 12.2. Regressa para o ponto 13.

Fluxo alternativo (5) [Convidado guarda os resultados na sua conta depois do campeonato terminar] (passo 14)

- 14.1 O Convidado efetua a autenticação e passa a ser jogador.
- 14.2 O sistema guarda os resultados no perfil do Jogador.
- 14.3 O sistema regressa ao menu inicial.

Fluxo de Exceção (6)[Convidado não dá login como Jogador depois do campeonato terminar](passo 14.1)

14.1.1 Os resultados do campeonato são descartados pelo sistema.

14.2.2 O sistema regressa ao menu inicial.

6. Use Case 6

Use Case: Efetuar login.

Descrição: O ator efetua login.

Cenário: Cenário 6 à “João efetua login.”

Pré-condição: Sistema deve estar iniciado.

Pós condição: O ator fica autenticado.

Fluxo Normal:

1. O ator fornece as credenciais ao sistema.
2. O sistema verifica e efetua a autenticação do ator.
3. O sistema vai para o menu inicial.

Fluxo de exceção (1): [os dados de autenticação não são válidos] (passo 1)

- 1.1 O sistema cancela o processo de autenticação.
- 1.2 Volta para o menu de autenticação.

7. Use Case 7

Use Case: Criar um usuário.

Descrição: O ator cria um usuário.

Cenários: Cenário 7 à “Manuel ainda não é usuário, logo cria um.”

Pré-condição: Sistema deve estar iniciado.

Pós condição: O sistema registra as credenciais do usuário.

Fluxo Normal:

1. O ator fornece um username e uma password.
2. O sistema cria o usuário.

Fluxo alternativo (1): [O username já existe] (passo 1)

- 1.1 O sistema indica que o username já foi utilizado.
- 1.2 Regressa para o ponto 1.

Fluxo alternativo (2): [Password é inválida] (passo 1)

1.1 O sistema indica que a password não é forte o suficiente.

1.2 Regressa para o ponto 1.

Conclusão

Com a elaboração dos modelos acima referidos, pensamos ter abordado todos os pontos chave para o desenvolvimento essencial da aplicação, tendo sempre em atenção as opiniões dos elementos do grupo e as dicas dadas pelos professores da disciplina.

Graças aos modelos realizados nesta fase, temos uma visão positiva e esclarecida face à próxima fase do projeto que se aproxima.