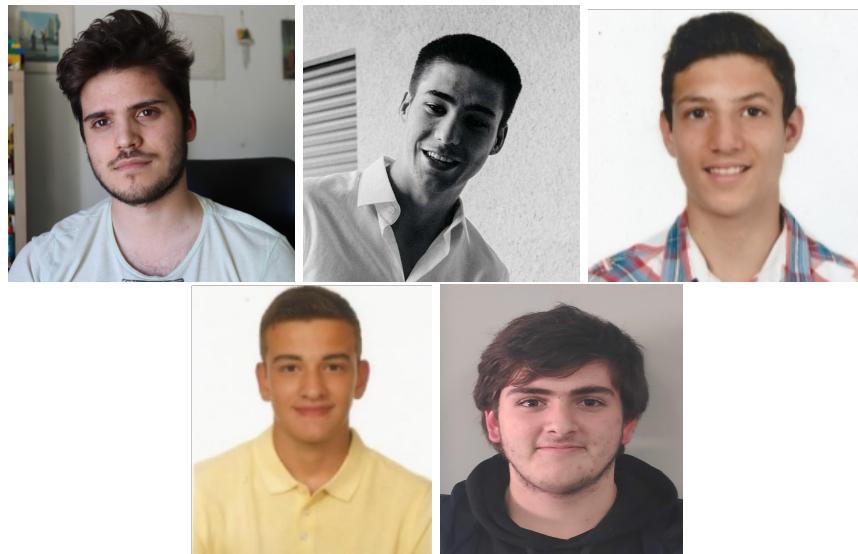




UNIVERSIDADE DO MINHO
Departamento de Informática

Entrega Intermédia parte 1 - Desenvolvimento de Sistemas de Software



Realizado por:

Andre Mota - A81384 | Simão Barroso - A96834 | Miguel Senra - A97496

Gonçalo Braga - A97541 | Rodrigo Pereira - A96561
GRUPO 21

<https://github.com/goncalobraga27/DSS-Grupo21.git>

20 de outubro de 2022

Conteúdo

1	Introdução	2
2	Objetivos da Primeira Fase	2
3	Descrição do enunciado proposto	2
3.1	Criação de pilotos	3
3.2	Criação de circuitos	3
3.3	Criação de carros	3
3.4	Criação de campeonatos	3
3.5	Configuração de campeonatos	4
3.6	Configuração de corrida	4
3.7	Simulação de uma corrida	4
4	Modelo de Domínio	4
4.1	Descrição do modelo de domínio	4
4.2	Modelo de domínio	5
5	Use cases	5
5.1	Requisitos	5
5.2	Os atores	6
5.3	Modelo de Use Cases	6
5.4	Tabelas de Use Cases	6
5.4.1	Registo de Utilizador	7
5.4.2	Fazer Login no Jogo	7
5.4.3	Criação de Campeonato	8
5.4.4	Criação de Piloto	8
5.4.5	Criação de Carro	9
5.4.6	Criação de Circuito	9
5.4.7	Configurar Campeonato	10
5.4.8	Configurar Corrida	10
5.4.9	Simular Corrida	11
6	Análise crítica e Resultados obtidos	12

1 Introdução

A unidade curricular (UC) de Desenvolvimento de Sistemas de Software pretende introduzir-nos às técnicas de planeamento de software usadas no mercado. Por esta razão, todos os anos é proposto um projeto para pôr em prática estes conhecimentos.

O projeto deste ano pede-nos que façamos um jogo de carros ao estilo do F1 Manager, e para tal pretende que façamos um sistema que tenha utilizadores e que faça simulações de corridas.

Na primeira parte deste trabalho, à qual este documento se refere, pretende-se mostrar modelo de domínio, diagrama e tabelas de use-cases referentes ao projeto. Para isso, ir-se-á recorrer sempre à linguagem UML, tal como lecionada nesta UC. Para além disso iremos abordar os principais objetivos, uma descrição do enunciado e as nossas considerações finais acerca desta fase.

2 Objetivos da Primeira Fase

Nós considerámos que existem vários objetivos com esta fase com várias vantagens para nós, alunos, entre as quais temos:

- a aplicação dos conhecimentos de Modelos de Domínio obtidos nas aulas teóricas;
- a aplicação dos conhecimentos de Casos de Uso obtidos nas aulas teóricas;
- perceber as dificuldades de construir estes modelos e consequentemente ultrapassá-las. (Não são modelos únicos, existem diferentes pontos de vista);
- valorizar a tomada de decisão em grupo e fundamentá-la convenientemente;
- dividir um problema em problemas mais simples de forma a obter uma solução mais completa.

3 Descrição do enunciado proposto

Este ano, na Unidade de Curricular de DSS, foi-nos pedido que fizessemos como trabalho prático uma aplicação similar ao F1Manager, jogo de simulação de corridas de carros. A aplicação a criar é na sua essência um jogo onde os utilizadores registados competem em provas automobilísticas que o software vai simular de forma a obter uma classificação final.

Sendo assim, tem de haver no jogo uma maneira de criarmos utilizadores, sejam eles administradores ou jogadores normais que vão poder participar nas corridas.

Além disso, como sabemos, num jogo inspirado em corridas automóveis, é necessário haver entidades básicas como pilotos, circuitos e carros. Assim, o administrador do jogo tem de ser capaz de adicionar novos elementos deste tipo.

Apesar de não acontecer no F1Manager já existente, no nosso jogo tem de ser possível criar campeonatos de forma a podermos jogar com outros jogadores.

Desta forma surge naturalmente, a configuração onde os jogadores são conectados para o mesmo campeonato e para as várias corridas, e depois, a respetiva simulação que existe de forma a atingir o objetivo do jogo, encontrar um vencedor.

3.1 Criação de pilotos

No jogo, cada piloto criado tem de possuir um nome, que serve como "identificador" do mesmo.

Além disso, como sabemos, cada piloto tem suas características de condução, por isso as mesmas estão implícitas no nível de perícia do piloto. O nível de perícia inclui CTS (condução em tempo seco vs chuva) e SVA(segurança vs agressividade). Estes parâmetros do nível de perícia vão ser abordados no Modelo de Negócio ou seja, na próxima fase, e influenciam muito o comportamento dos carros na hora da simulação.

3.2 Criação de circuitos

De modo a poder realizar corridas, é obrigatório ter circuitos. Com isto cada circuito tem de possuir um nome, que serve como "identificador" do mesmo, pelo que este deve ser único.

Um circuito é formado por um conjunto encadeado de retas, curvas e chicanes. Por este motivo, cada uma destas secções da pista tem um grau da dificuldade de ultrapassagem associado, naquele exato sítio, que representa a dificuldade dos pilotos para ultrapassar naquele local.

Com isto que foi dito anteriormente, conseguimos exemplificar uma volta de um circuito.

Consequentemente, de forma a facilitar a compreensão, iremos exemplificar uma volta de um circuito de forma prática.

Como uma corrida de fórmula 1 não é obviamente constituída apenas por uma volta, tem de ser indicado também o número de voltas que o circuito vai possuir. Esta é uma propriedade do circuito visto que, segundo o enunciado, cada corrida num mesmo circuito tem sempre o mesmo número de voltas.

3.3 Criação de carros

No nosso jogo, o jogador vai poder escolher um carro entre as diversas categorias existentes. As categorias existentes são: C1, C2, Grande Turismo(GT) e Stock Cars(SC).

Cada uma destas categorias tem como unidade motriz obrigatória um motor de combustão. Assim para este motor tem de ser indicada a sua respetiva cilindrada e a potência que o mesmo produz.

Como hoje em dia, os carros mais modernos possuem motores elétricos associados ao motor de combustão, os conhecidos híbridos, esta realidade foi retratada no nosso jogo, sendo que as categorias C1,C2,GT podem possuir dois motores(elétrico+combustão). Se for acrescentado um motor elétrico ao carro, tem de ser indicada a potência gerada pelo mesmo.

Os carros não são peças indestrutíveis pelo que existe uma fiabilidade dos mesmos, que se vai alterar por categoria de carro escolhida. Na categoria de carros GT, existe também um fator de degradação onde a sua fiabilidade vai diminuindo ao longo do decorrer de uma corrida. Caso a fiabilidade de um carro chegue a 0, este carro fica incapacitado de concluir a corrida, pelo que perde a corrida imediatamente.

3.4 Criação de campeonatos

Tal como a maior parte dos jogos atualmente, um dos objetivos deste jogo é promover um ambiente hostil multiplayer. Sendo assim, é atribuído ao campeonato criado um nome, que serve como "identificador" de cada campeonato, tal como por exemplo, associando à realidade, existe o Grande Prémio de Fórmula 1. Cada campeonato é constituído por diversos circuitos, onde se vão realizar as várias provas (corridas), escolhidos pelo administrador do jogo.

Com isto, quando um jogador decide jogar um campeonato, apenas necessita de ver na lista quais os campeonatos disponíveis para serem jogados, e escolher um.

3.5 Configuração de campeonatos

No jogo, cada jogador só pode escolher um piloto e um carro para utilizar em todo o campeonato. Por este motivo, antes do campeonato começar, cada jogador tem de fazer as suas escolhas para esse mesmo campeonato. Os jogadores fazem, portanto, um registo com a informação do carro e do piloto e existe uma sincronização entre todos os jogadores de forma a estarem todos prontos para iniciar o campeonato.

3.6 Configuração de corrida

Um campeonato é constituído por várias corridas onde cada corrida ocorre num circuito. Como cada corrida é diferentes das demais, seja em termos de características da pista, de localização, temporais, ou ambos, o tempo metereológico poderá ser diferente do da corrida transata, o que promove algum desafio no jogo, em termos de gestão, como também aleatoriedade nos resultados. De forma a corresponder a estes desafios, os utilizadores deverão alterar a afinação do carro antes de cada corrida. No entanto, existe um limite tendo em consideração que o número de afinações que os utilizadores podem realizar é no máximo 2/3 do número de corridas do campeonato.

Depois de terminados estes procedimentos, é que é possível ser simulada uma corrida.

3.7 Simulação de uma corrida

Depois de todos os processos de criação de entidades do jogo e de configuração de algumas delas, surge a parte mais interessante do jogo, a simulação da corrida.

A simulação da corrida tem em conta o circuito, os carros e os pilotos e vai indicando, para cada curva, recta e chicane de cada volta, eventuais ultrapassagens, despistes e avarias. Assim, conseguimos, determinar ao fim de cada volta as posições dos carros/pilotos/jogadores.

O simulador tem um papel ativo, ao longo da corrida, para cada carro, e para cada posição em pista do mesmo se é ou não possível realizar a ultrapassagem ao seu adversário direto.

Quando a corrida termina, é apresentado aos jogadores a sua posição final na corrida, bem como as posições dos seus adversários na corrida.

4 Modelo de Domínio

Face ao enunciado proposto, foi-nos pedido que elaborássemos um modelo de domínio capaz de representar este projeto. Procedemos à escolha de entidades, respetivas relações e multiplicidade destas conforme descrito no próximo ponto, e procedeu-se à elaboração dum modelo de domínio em UML.

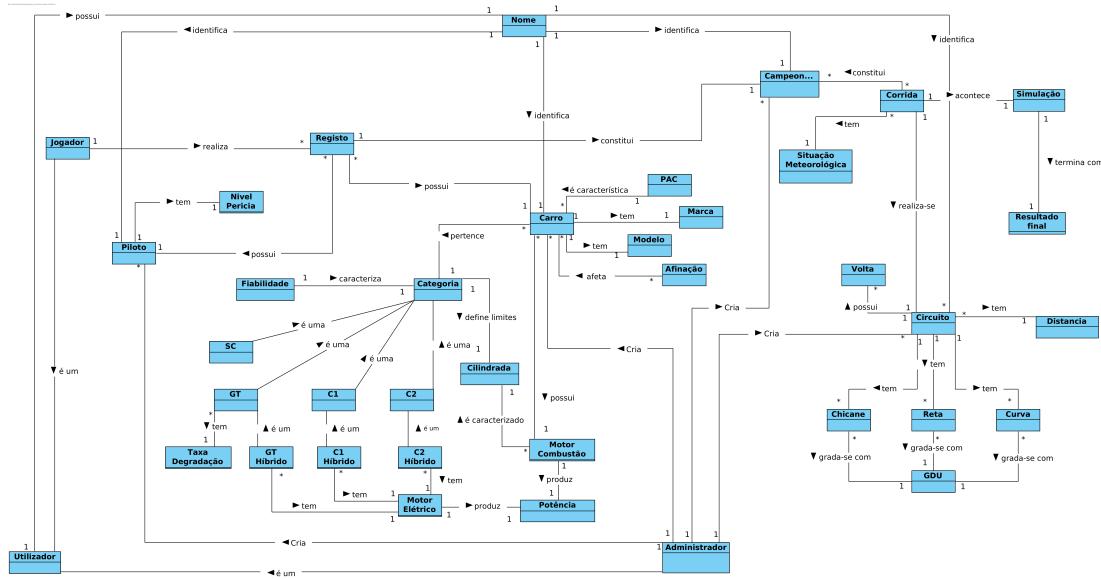
4.1 Descrição do modelo de domínio

A nossa abordagem inicial para este modelo consistiu na procura de todos os substantivos que pudessem corresponder a algo que fizesse parte do jogo, pelo que não foi difícil encontrar as entidades *principais*. Com estas entidades encontradas, foi muito mais fácil encontrar aspectos secundários e relações mais pormenorizadas destas, que acabamos também por incluir no nosso modelo de domínio.

Esta última abordagem pode, no entanto, ter incorrido num escrutínio demasiado aprofundado que levou a que tivéssemos candidatos a entidades que não seriam entidades na nossa última versão do modelo. Isto trouxe-nos algumas dúvidas que foram resolvidas ao longo das aulas práticas. Uma segunda análise mais cuidadosa, com este ponto de vista em mente, levou

ao presente modelo, que díriamos ter 4 conjuntos distintos de entidades que se relacionam de forma mais direta com os seus conjuntos mais pormenorizados de entidades.

4.2 Modelo de domínio



5 Use cases

A elaboração deste trabalho requeria também uma forma de representar diferentes *use cases* do sistema. Como estes se relacionam de forma direta com os cenários de utilização, a grande maioria destes foi automaticamente selecionado a partir das situações descritas nestes, como evidenciado mais abaixo.

Com isto, procedeu-se à elaboração dum diagrama de *use cases* e também de tabelas que representam os principais *use cases* referidos nesses cenários.

5.1 Requisitos

Após uma análise do enunciado, através dos vários cenários, conseguimos recolher os seguintes 9 requisitos funcionais:

- Registo de Utilizador
- Fazer Login no Jogo
- Criação de Piloto
- Criação de Carro
- Criação de Circuito
- Criação de Campeonato
- Configurar Campeonato

- Configurar Corrida
- Simular Corrida

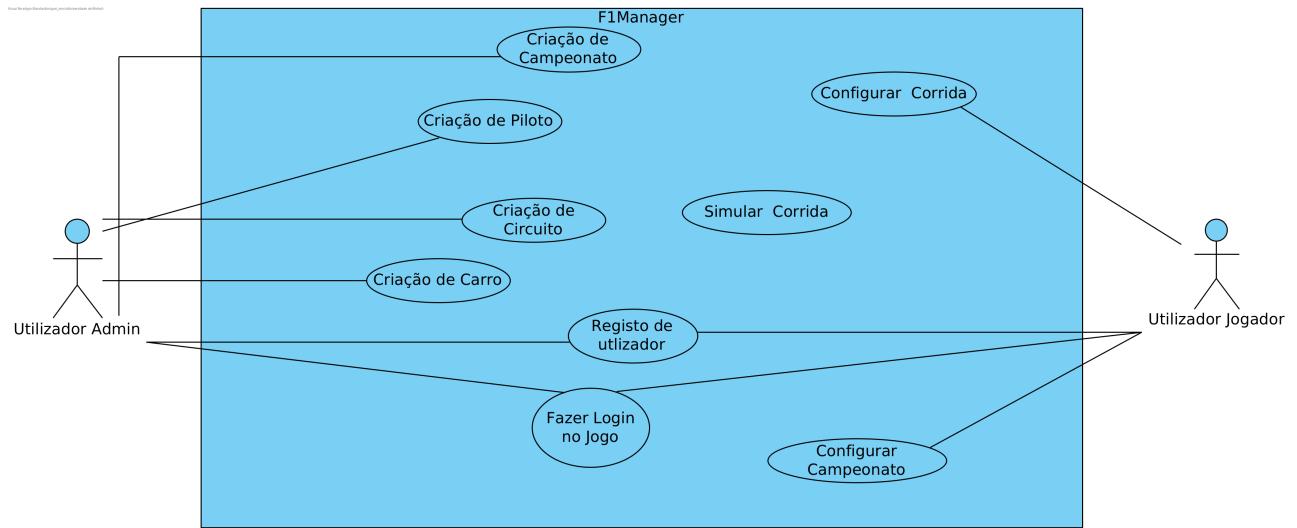
5.2 Os atores

Durante a análise do enunciado, pudemos constatar que existem 2 atores, o utilizador administrador e o utilizador jogador. A nossa análise mais aprofundada acabou por confirmar a nossa primeira ideia, pelo que se identificam os seguintes atores:

- **Utilizador Administrador:** é o ator responsável pela criação de todos os componentes de interação que todos os outros atores que não sejam administradores possam ter (representado pelo José).
- **Utilizador Jogador:** é o tipo de ator que vai na realidade utilizar o sistema, ou seja, jogar o jogo (representado pelo Francisco, pela Sara, pela Daniela e pelo Manuel).

5.3 Modelo de Use Cases

O modelo de Use Cases do nosso projeto é o seguinte



5.4 Tabelas de Use Cases

Através dos cenários dados e algumas das explicações do funcionamento esperado do sistema, concluímos que os 9 use cases abaixo descritos seriam os principais use cases que este sistema iria ter.

5.4.1 Registo de Utilizador

Use Case		
Use Case	Registo de Utilizador	
Cenário	Todos os cenários do enunciado	
Autor	Utilizador Administrador e Utilizador Jogador	
Descrição	Um utilizador regista-se no jogo	
Pré-Condição	True	
Pós-Condição	É criado um jogador , que pode administrar o jogo, ou jogá-lo	
Fluxo Normal	Autor	Sistema
	1. Sistema pede um nome de utilizador	
	2. Sistema verifica o nome de utilizador	
	3. Sistema pede uma password	
	4. Sistema verifica a password	
	5. Utilizador escolhe ser administrador do jogo	
	6. Sistema verifica o modo de jogador	
	7. Sistema cria o jogador	
Fluxo Alternativo (1): Criação de jogador "Normal"	5.1 Utilizador escolhe ser jogador "normal"do jogo	Regressa a 6
Fluxo de Exceção (2): Password inválida (Passo 4)		4.1 Sistema não obtém uma password válida
		4.2 Sistema cancela criação de jogador
		4.3 Sistema termina processo
Fluxo de Exceção (3): Nome de utilizador inválido (Passo 2)		2.1 Sistema não obtém um nome de utilizador válido
		2.2 Sistema cancela a criação de jogador
		2.3 Sistema termina processo
Fluxo de Exceção (4): Não é indicado o tipo de jogador (Passo 6)		6.1 Sistema que deteta que o tipo de jogador não é válido
		6.2 Sistema cancela criação de jogador
		6.3 Sistema termina o processo

5.4.2 Fazer Login no Jogo

Use Case		
Use Case	Fazer login no jogo	
Cenário	Todos os cenários do enunciado	
Autor	Utilizador Administrador e Utilizador Jogador	
Descrição	Um utilizador faz login no jogo	
Pré-Condição	O utilizador já está registado no jogo	
Pós-Condição	O utilizador está pronto a utilizar o sistema	
Fluxo Normal	Autor	Sistema
	1. O Ator insere o nome de utilizador	
	2. O Ator insere a password	
		3. Sistema verifica que o utilizador está registado
		4. Sistema verifica se o nome de utilizador inserido corresponde á password inserida
		5. Sistema informa que o utilizador está autenticado
Fluxo de Exceção (1): Utilizador não está registado (Passo 3)		3.1 Sistema informa que o nome de utilizador inserido não corresponde a um utilizador válido
		3.2 Sistema termina processo
Fluxo de Exceção (2): O utilizador não consegue fazer login (Passo 4)		4.1 Sistema informa que o nome de utilizador ou password estão erradas
		4.2 Sistema termina processo

5.4.3 Criação de Campeonato

Use Case	
Use Case	Criação de Campeonato
Cenário	1 - Campeonatos
Ator	Utilizador Administrador
Descrição	Um jogador administrador cria um campeonato
Pré-Condição	É autenticado no jogo um utilizador administrador
Pós-Condição	É criado um campeonato, que pode ser jogado
Ator	Sistema
Fluxo Normal	1. o A ator faz login no jogo como administrador
	2. O sistema verifica a autenticação do utilizador
	3. Ator dá um nome ao campeonato criado
	4. O sistema verifica se o nome do campeonato já existe
	5. O ator adiciona circuitos ao campeonato criado
	6. Sistema verifica se existe os circuitos pedidos para serem inseridos no campeonato
	7. O ator consulta a lista de campeonatos criados
	8. O ator adiciona o campeonato à lista de campeonatos disponíveis para serem jogados imediatamente
Fluxo de Exceção (1) : Utilizador Inválido (Passo 2)	2.1 Sistema informa que este utilizador é inválido
Fluxo de Exceção (2): Campeonato já existe (Passo 4)	2.2 Sistema termina processo
Fluxo de Exceção (3): Circuito não existe (Passo 6)	4.1 Sistema informa que o nome do campeonato já existe
	4.2 Sistema termina processo
	6.1 Sistema informática que o nome de um ou mais circuitos não existe
	6.2 Termina processo

5.4.4 Criação de Piloto

Use Case	
Use Case	Criação de Piloto
Cenário	4 - Pilotos
Ator	Utilizador Administrador
Descrição	O utilizador administrador cria pilotos
Pré-Condição	É registado no jogo um utilizador administrador
Pós-Condição	É criado um piloto, que pode competir em campeonatos
Ator	Sistema
Fluxo Normal	1. o A ator faz login no jogo como administrador
	2. O sistema verifica a autenticação do utilizador
	3. Ator decide inserir um novo piloto
	4. O ator indica o nome do novo piloto a criar
	5. O ator indica o CTS do novo piloto a criar
	6. O ator indica o SVA do novo piloto a criar
	7. O sistema verifica se existe um piloto criado com esses atributos
	8. O sistema adiciona o piloto à lista de pilotos disponíveis para jogar
Fluxo de Exceção (1): Ator não pode criar pilotos (Passo 1)	9. O sistema informa que o piloto está disponível para jogar
Fluxo de Exceção (2): O piloto que se está a tentar criar já existe(Passo 7)	2.1 Um ator faz login no jogo sem ser como administrador
	2.2 Sistema termina processo
	7.1 Sistema deteta que existe um piloto criado exatamente igual
	7.2 Sistema bloqueia a criação do piloto
	7.3 Sistema termina o processo

5.4.5 Criação de Carro

Use Case		
Use Case	Criação de Carro	
Cenário	3 - Carros	
Ator	Utilizador Administrador	
Descrição	Utilizador administrador cria carros	
Pré-Condição	É registado no jogo um utilizador administrador	
Pós-Condição	É criado um carro, que pode ser utilizado em campeonatos	
Fluxo Normal		
O carro a criar, se a categoria o permitir, não é híbrido		Ator
1. O ator faz login no jogo como administrador		Sistema
3. O ator decide inserir um novo carro		
4. O ator escolhe qual categoria do carro a criar		
5. O ator indica a marca do carro a criar		
6. O ator indica o modelo do carro a criar		
7. O ator indica a cilindrada e a potência do motor a combustão que tem o carro a criar		
8. O ator com base na categoria escolhida para o carro, decide que não é híbrido		
9. O ator indica qual o valor do perfil aerodinâmico do carro		
10. O sistema verifica se este carro já se encontra criado		
11. O sistema adiciona o carro à lista de carros disponíveis para jogar		
12. O sistema informa que o carro está disponível para jogar		
Fluxo Alternativo (1):		
O carro a criar, se a categoria o permitir, é híbrido (passo 8)		
8.1 O ator com base na categoria escolhida para o carro, decide que o carro é híbrido		
8.2 O sistema verifica se a categoria do carro permite motor híbrido		
8.3 O sistema informa que a categoria do carro permite motor híbrido		
8.4 O ator indica a potência adicional do motor elétrico		
8.5 Regressa a 9		
Fluxo Alternativo (2):		
O carro a criar, devido à categoria, não pode ser híbrido (passo 8.2)		
Fluxo de Exceção (3):		
Ator não pode criar carros (Passo 1)		
Fluxo de Exceção (4):		
O carro que se está a tentar criar já existe (passo 10)		

5.4.6 Criação de Circuito

Use Case		
Use Case	Criação de circuito	
Cenário	2 - circuitos	
Ator	Utilizador Administrador	
Descrição	Utilizador administrador faz a criação de um circuito	
Pré-Condição	É registado no jogo um utilizador administrador	
Pós-Condição	É criado um circuito, que pode ser inserido em campeonatos	
Fluxo Normal		
Fluxo Normal		
1. o A ator faz login no jogo como administrador		Ator
2. O sistema verifica a autenticação do utilizador		Sistema
3. Ator decide inserir um novo circuito		
4. Ator dá um nome ao circuito criado		
5. Sistema verifica se o nome do circuito já existe		
6. O ator indica a distância que o circuito possui		
7. O ator indica o número de curvas que o circuito possui		
8. O ator indica o número de chicane que o circuito possui		
9. O sistema calcula o número de retas que o circuito possui		
10. O sistema informa o número de retas existente no circuito		
11. O sistema apresenta a lista de curvas e retas existentes no circuito		
12. O ator indica o GDU em cada curva e reta		
13. O circuito é registado no sistema		
14.O ator indica o número de voltas que cada corrida possui		
15. O sistema adiciona o circuito à lista de circuitos disponíveis para jogar		
16. O sistema informa os circuitos disponíveis para jogar		
17. O circuito passa a estar disponível para integrar campeonatos		
Fluxo de Exceção (1): Utilizador inválido (Passo 2)		
2.1 Sistema informa que este utilizador é inválido		
2.2 Sistema termina processo		
Fluxo de Exceção (2): Circuito já existe (Passo 5)		
5.1 Sistema informa que o nome do circuito já existe		
5.2 Sistema termina processo		

5.4.7 Configurar Campeonato

Use Case			
Use Case	Configurar Campeonato		
Cenário	5 - Configurar campeonato		
Autor	Utilizador jogador, Utilizador Administrador		
Descrição	Configuração do campeonato para ser jogado		
Pré-Condição	Atores querem jogar o campeonato		
Pós-Condição	O campeonato que foi configurado já é possível ser jogado		
		Autor	
1. o A ator faz login no jogo		2. O sistema verifica a autenticação do utilizador	
3. Ator decide jogar um campeonato		4. Sistema verifica se existem campeonatos criados	
5. Ator escolhe o carro a utilizar no campeonato		8. Sistema informa que o campeonato está disponível para ser jogado	
6. Ator escolhe o piloto a utilizar no campeonato		2.1 Sistema não consegue autenticar o ator	
7. Ator espera que outros atores configurem o seu campeonato		2.2 Sistema termina processo	
Fluxo Normal			
Fluxo de Exceção (1): Utilizador não é validado (Passo 1)			
Fluxo de Exceção (2): Não existem campeonatos que possam ser jogados (Passo 4)			
		4.1 Sistema informa que o nome do circuito já existe	
		4.2 Sistema bloqueia a configuração de campeonatos	
		4.3 Sistema termina o processo	

5.4.8 Configurar Corrida

Use Case		
Use Case	Configurar Corrida	
Cenário	5 - Simular Corrida	
Autor	Jogador	
Descrição	Configuração das corridas no campeonato	
Pré-Condição	Os atores estão disponíveis para começar a corrida, e estão inscritos no campeonato	
Pós-Condição	Os atores têm o setup da corrida pronto a iniciar	
		Autor
1. O ator regista-se no campeonato		2. O sistema verifica se o campeonato existe
		3. O sistema apresenta o nome do circuito
		4. O sistema apresenta a situação meteorológica para a corrida
A afinação do carro não é alterada		
5. O ator decide não alterar a afinação do carro		
6. O ator escolhe quais os pneus a utilizar na corrida		
7. O ator escolhe qual o modo de motor a utilizar na corrida		
		8. O sistema informa que a corrida pode ser iniciada
Fluxo Normal		
Fluxo Alternativo (1) : A afinação do carro é alterada (Passo 5)	5.1 O ator decide alterar a afinação do carro	
	5.2 O ator altera o PAC do carro	
	7. Regressa a 6	
Fluxo de Exceção (2) : (Passo 2) Tentativa de registo no campeonato quando este não existe		
	2.1 Sistema verifica que o campeonato não existe	
	2.2 Sistema informa que o campeonato não existe	
	2.3 Sistema termina o processo	

5.4.9 Simular Corrida

Use Case	
Use Case	Simular Corrida
Cenário	5 - Simular Corrida
Ator	Jogador
Descrição	Simulação de corridas
Pré-Condição	Todos os atores que vão participar na corrida estão prontos
Pós-Condição	São apresentados os resultados da corrida
Atores	
Sistema	
Fluxo Normal	1. Os atores possuem todo o "race setup" pronto
	2. O sistema verifica se todos os atores têm o "race setup" pronto
	3. O sistema simula a corrida
	4. A corrida termina
	5. O sistema apresenta os resultados da corrida
	6. O sistema acrescenta os pontos desta corrida às estatísticas do campeonato
	7. O sistema apresenta as estatísticas do campeonato
Fluxo de Exceção (1) : Caso a corrida não termine (Passo 3)	3.1 O sistema deteta que a corrida não terminou
	3.2 O sistema informa que a corrida não terminou
	3.3 O sistema termina o processo

6 Análise crítica e Resultados obtidos

Por forma a entender mais sobre modelação de software, analisamos o enunciado com exigência com o objetivo de obtermos a melhor modelação possível. Finda esta fase do trabalho, achamos ter cumprido todos os objetivos pelo enunciado e por nós propostos. A fase de elaboração do modelo de domínio ajudou-nos a perceber o cerne do problema, ou seja os pontos principais. Desta forma conseguimos clarificar pormenores duma futura implementação deste projeto.

Por fim, conseguimos criar o modelo de domínio do nosso projeto, pensando e, acima de tudo, discutindo várias opções entre os diversos elementos do grupo, nas entidades e nos relacionamentos entre elas. Tal como apresentado no enunciado e justificado durante este relatório, tentamos colocar aquelas entidades que são mais relevantes e que apenas representam o problema e não a solução. Existiram portanto, entidades que não se encontram no modelo de domínio visto não terem grande relevância para representar o problema. De igual forma, criamos o modelo de Use Cases, que reflete as opções principais que o jogo irá disponibilizar aos diferentes utilizadores.

Dito isto, achamos essa solução bastante representativa do problema proposto, e que esta será um bom guia para as próximas fases, mantendo alguma flexibilidade para a implementação, conforme a situação o necessite.