



# **Desenvolvimento de Sistemas de Software**

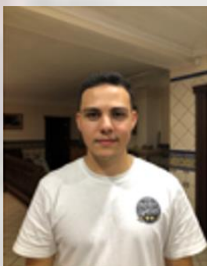
## **- Trabalho Prático -**

### **- Jogo Formula 1 Manager (Fase 1) -**

**Universidade do Minho**

**Licenciatura em Engenharia Informática**

**Trabalho realizado pelo grupo 14 composto por:**



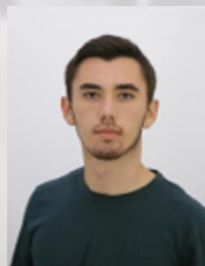
- A95442 -

André Oliveira Gonçalves



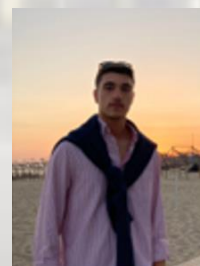
- A95460 -

Diogo Luís Almeida Costa



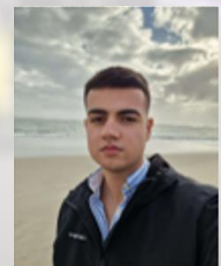
- A95323 -

Henrique Ribeiro Fernandes



- A96726 -

Ivo Miguel Alves Ribeiro



- A95641 -

João Pedro Moreira Brito

Ano Letivo 2022/2023

Submetido a: 20/10/2022



# Índice

<b>Índice</b>	<b>2</b>
<b>Introdução</b>	<b>3</b>
<b>Objetivos desta fase</b>	<b>3</b>
<b>Descrição do trabalho</b>	<b>4</b>
<b>Modelo de Domínio</b>	<b>4</b>
<b>Modelo de Use Case</b>	<b>6</b>
<b>Especificação dos use cases:</b>	<b>8</b>
• Relativamente ao Administrador:	8
UC - Criar Campeonato	8
UC - Eliminar Campeonato	9
UC- Criar Circuitos	9
UC - Eliminar Circuito	10
UC - Adicionar Piloto	10
UC - Eliminar Piloto	11
UC- Criar carro	12
UC - Eliminar Carro	14
• Relativamente ao Jogador:	14
UC - Escolher Campeonato	14
UC - Escolher Piloto	15
UC- Escolher Carro	15
UC - Escolher pneu	16
UC - Escolheu tipo de motor	17
UC - Jogar Corrida	17
• Ambos dispõem das seguintes funções:	18
UC- Registar	18
UC - Fazer login	19
<b>Conclusão</b>	<b>20</b>



## Introdução

No âmbito da cadeira de Desenvolvimento de Sistemas de Software, pretende-se a implementação de um sistema que permita simular campeonatos de automobilismo, sendo a génese desta aplicação similar ao de uma já existente, denominada “F1 Manager”. A aplicação a desenvolver deve ser na sua essência um jogo, na qual os utilizadores registados nela podem competir em provas automobilísticas que o software vai simular.

Para facilitar o levantamento de requisitos, apresentaram-se um conjunto de cenários de utilização do jogo, os quais devem ser suportados pela aplicação a desenvolver.

## Objetivos desta fase

Os objetivos pretendidos nesta fase são tanto a organização das ideias como a dos planos de execução a seguir por parte da nossa equipa de trabalho para podermos colocar em prática o enunciado e de facto desenvolvermos a aplicação em si, da maneira mais eficaz e correta possível.

Uma vez organizadas e planeadas podemos pôr em prática as nossas ideias e passar à próxima fase do trabalho e começar a conceber a aplicação.

Para isso desenvolvemos:

- Um Modelo de Domínio com as entidades relevantes;
- Um Modelo de Use Case (diagramas mais especificações do Use Case) com as funcionalidades propostas para o sistema;



## Descrição do trabalho

### Modelo de Domínio

Primeiramente começou-se por desenvolver o Modelo de Domínio, identificando as entidades relevantes, a partir dos cenários apresentados no enunciado.

Através do **cenário 1**, reparou-se que umas das entidades necessárias criar seria logo a entidade administrador, a entidade jogo, a entidade campeonato e a entidade circuitos. A entidade jogo associa-se com a entidade campeonato, pois o jogo é constituído por um conjunto de campeonatos, a entidade circuitos associa-se com a entidade campeonatos, pois um campeonato é formado por circuitos.

De seguida no **cenário 2**, decidiu-se adicionar as entidades curvas, km, retas, número de voltas, chicane e nome do circuito, estando estas todas associadas ao circuito. Depois, adicionou-se a entidade GDU que define a dificuldade de ultrapassagem, estando esta associada às entidades: curvas, retas e chicane.

Já no **cenário 3**, criou-se a entidade carro, sendo que surgiu uma certa divergência pois o grupo tinha em mente criar o menor número de entidades possíveis, mas numa pequena discussão com um docente da disciplina percebeu-se que o intuito dos cenários era adicionarmos mais entidades. Sendo assim, acrescentamos as entidades modelo, marca e categoria, esta última podendo ser C1, C2, GT, SC, que são todas as categorias existentes no jogo para carros. As categorias GT, C1, C2, podem ser híbridos, pelo que se criou as entidades GT híbrido, C1 híbrido, C2 híbrido associadas às entidades das categorias respetivas e como estas possuem um motor elétrico, formou-se e associou-se elas a uma entidade chamada motor elétrico. Posteriormente, criou-se a entidade fiabilidade que está relacionada com a entidade categoria, e com a entidade carro. A entidade potência que se associa à entidade cilindrada e à entidade motor elétrico está associada à entidade carro. A cilindrada é definida pela entidade motor a combustão, e esta possui uma entidade referida como afinação mecânica. A fiabilidade dos GT é afetada pelo número de voltas realizadas, pelo que, elaborou-se a entidade desgaste que interliga estas duas entidades.



De seguida no **cenário 4**, abordou-se a entidade piloto que é identificada pela entidade nome do piloto e caracterizada por dois critérios, que se definiu como entidades, “Chuva vs. Tempo Seco” (CTS) e “Segurança vs. Agressividade” (SVA). Estas duas entidades definem a entidade habilidade do piloto, que irá a definir a entidade capacidade de ultrapassagem (referida no cenário 1).

Por último, analisando o **cenário 5**, reparou-se que este seria o cenário que nos iria dar mais informações sobre a parte essencial da aplicação, a parte de jogar, começou-se por formar a entidade jogador, que se associa à entidade jogo. O jogador será responsável por escolher o campeonato, alterar a entidade afinação do carro (PAC) (que depende da entidade campeonato e da entidade carro), escolher a entidade modo do motor (que tem como modos: conservador, agressivo, normal, sendo estas todas entidades associadas à entidade modo do motor). O jogador pode ainda escolher a entidade piloto, alterar a entidade pneu (associada à entidade categoria do pneu, que pode ser soft, hard, wet, sendo estas entidades associadas à categoria do pneu, como é lógico) e ainda escolher a entidade carro (abordada no cenário 3). De seguida, notou-se que seria necessário adicionar uma entidade denominada condição climática, que iria estar associada à entidade corrida, esta seria relacionada com a entidade seco e com a entidade chuva, que são as duas possibilidades para as condições climáticas. Por fim, o grupo decidiu abordar a versão premium, que deu algumas dúvidas como a englobar no modelo de domínio, acabou-se por decidir adicionar a mesma a partir da entidade jogador premium, que se relaciona com a entidade jogador.





## Modelo de Use Case

Depois de realizado o Modelo de Domínio, começou-se a fazer o diagrama do Modelo de Use Case.

No nosso sistema temos presentes dois atores, um deles um jogador e o outro um administrador. Para este modelo tanto o ator jogador como o administrador têm a são capazes de tanto de fazer login como de se registar no jogo.

Para este caso o Jogador tem de ser capaz de escolher campeonato; escolher piloto; escolher pneu; escolher carro; escolher modo do motor e jogar a corrida. Por outro lado, o Administrador é capaz de criar o campeonato; criar circuitos (definindo todos os seus atributos específicos como número de curvas, número de chicanes, etc); criar pilotos; criar carro assim como as ações inversas a estas como eliminar campeonato; eliminar circuito; eliminar piloto e eliminar carro.

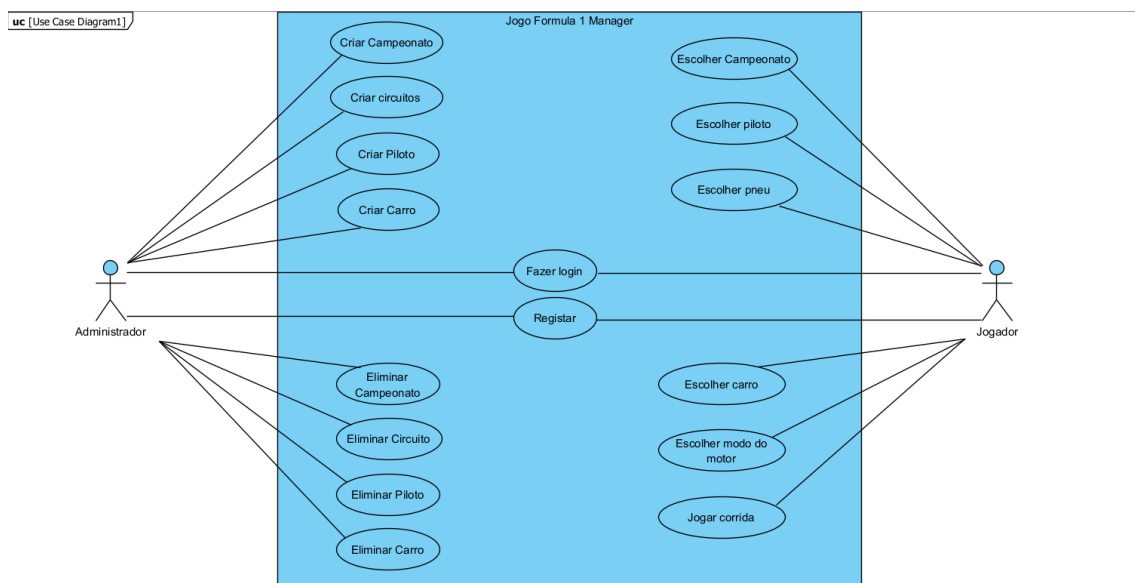


Figura 2 – Use Case Diagram



## Especificação dos use cases:

- Relativamente ao Administrador:

### UC - Criar Campeonato

Cenário: Cenário 1

Pré-condição - Administrador autenticado

Pós-condição - Sistema fica com mais um campeonato pronto para jogar

Fluxo Normal:

- 1- Administrador indica o nome
- 2- Sistema verifica o nome e pede os circuitos que o campeonato terá
- 3- Administrador indica os circuitos
- 4- Sistema verifica os circuitos
- 5 - Sistema regista o campeonato

Fluxo Exceção 1[nome repetido] [passo2]

- 2.1 Sistema verifica que o nome já é usado
- 2.2 Sistema cancela a criação do campeonato

Fluxo Exceção 2[não há circuitos] [passo 4]:

- 4.1 - Sistema não recebe nenhum circuito
- 4.2 - Sistema cancela a criação do campeonato





## UC - Eliminar Campeonato

Cenário: Cenário 1

Pré-condição - Existe pelo menos um campeonato criado

Pós-condição - O Campeonato é eliminado do Sistema

Fluxo Normal:

- 1- O Administrador indica o nome
- 2- Sistema verifica se o Campeonato existe
- 3- Sistema elimina o respetivo campeonato

Fluxo Exceção 1[Campeonato não existe] [passo 2]:

- 2.1 - Sistema verifica que o nome do campeonato é inválido
- 2.2 - Nenhum campeonato é eliminado

## UC- Criar Circuitos

Cenário: Cenário 2

Pré-Condição: Administrador autenticado

Pós Condição: Sistema regista mais um circuito operacional

Fluxo Normal:

- 1-Administrador indica o nome do circuito
- 2-Sistema verifica o nome e pede o tamanho, o número de curvas e o número de chicanes do percurso
- 3- Administrador indica o tamanho, o número de curvas e o número de chicanes do percurso
- 4- Sistema verifica o número de retas e pede o GDU de cada reta, curva e chicane
- 5- Administrador indica o GDU de cada uma delas
- 6- Sistema regista e pede o número de voltas que a corrida terá
- 7- Administrador indica o número de voltas
- 8 - Sistema regista o circuito



Fluxo Exceção 1[nome repetido] [passo2]

2.1 Sistema verifica que o nome já é usado

2.2 Sistema cancela a criação do circuito

### UC - Eliminar Circuito

Cenário: Cenário 2

Pré-condição: Administrador autenticado

Pós-Condição: Sistema apaga o registo do circuito

Fluxo Normal:

1- Administrador indica o nome do circuito que quer apagar

2- Sistema verifica o nome

3- Sistema elimina o circuito

Fluxo Exceção1[nome não existe] [passo 2]

2.1 Sistema verifica que nome não existe

2.2 Sistema cancela a eliminação do circuito

### UC - Adicionar Piloto

Cenário: Cenário 4

Pré-condição: Administrador autenticado

Pós-Condição: Sistema regista mais um piloto pronto para ser usado

Fluxo Normal:

1- Administrador indica o nome do piloto

2-Sistema verifica o nome e pede o CTS e o SVA do piloto

3-Administrador indica o CTS e o SVA do piloto

4- Sistema regista o piloto



Fluxo Exceção 1[nome repetido] [passo2]

2.1 Sistema verifica que o nome já é usado

2.2 Sistema cancela a criação do piloto

### UC - Eliminar Piloto

Cenário: Cenário 4

Pré-Condição: Existe pelo menos um piloto criado

Pós-condição: O piloto é eliminado do sistema

Fluxo Normal:

1- Administrador indica o nome do piloto

2- Sistema verifica o nome do piloto

3- Sistema elimina o piloto

Fluxo Exceção 1[Piloto inexistente] [passo 2]:

2.1 - Sistema verifica que o piloto não existe

2.2 - Sistema cancela o processo de eliminação



## UC- Criar carro

Cenário: Cenário 3

Pré-condição- Administrado autenticado

Pós-condição - Sistema fica com mais um carro disponível para jogar

Fluxo Normal:

- 1- Sistema apresenta categorias disponíveis
- 2- Administrador escolhe categoria, marca, modelo, cilindrada e potencia
- 3 - Sistema verifica que carro é C1 e que necessita de fiabilidade (pode ser híbrido)
- 4- Administrador indica a fiabilidade
- 5 - Sistema verifica que fiabilidade é aproximadamente 95%
- 6 - Administrador indica que é não híbrido
- 7 - Administrador indica PAC
- 8 - Sistema regista carro

Fluxo alternativo 1 [carro é SC] [passo3]

- 3.1 Sistema verifica que carro é SC
- 3.2 regressa a 7

Fluxo Exceção 2 [fiabilidade muito distante de 95%] [passo 5]

- 5.1 Sistema verifica que fiabilidade é demasiado distante de 95%
- 5.2 Sistema termina processo

Fluxo alternativo 3 [carro é híbrido] [passo 6]

- 6.1 Administrador indica que é híbrido e a potencia do motor elétrico
- 6.2 regressa a 7



#### Fluxo alternativo 4 [Carro é G2] [passo3]

3.1 Sistema verifica que carro é C2 e que necessita de fiabilidade (pode ser híbrido)

3.2 Administrador indica fiabilidade

3.3 Sistema verifica que fiabilidade é aproximadamente 80%

3.4 regressa 6

#### Fluxo Exceção 5 [fiabilidade muito distante de 80%] [passo 3.3]

3.3.1 Sistema verifica que fiabilidade é demasiado distante de 80%

3.3.2 Sistema termina processo

#### Fluxo alternativo 6 [Carro é GT] [passo3]

3.1 Sistema verifica que carro é GT e que necessita de fiabilidade (pode ser híbrido)

3.2 Administrador indica fiabilidade

3.3 regressa 6



## UC - Eliminar Carro

Cenário: Cenário 3

Pré-condição: Existe pelo menos um carro criado

Pós-condição: O carro é eliminado pelo sistema

Fluxo Normal:

- 1- Administrador indica a categoria, marca, modelo, cilindrada e potencia
- 2- Sistema verifica o carro
- 3- Sistema elimina o carro

Fluxo Exceção 1[carro inexistente] [passo 2]:

- 2.1- Sistema verifica que carro não existe
- 2.2- Sistema cancela o processo de eliminar carro

- Relativamente ao Jogador:

## UC - Escolher Campeonato

Cenário: Cenário 1

Pré-condição: Jogador autenticado

Pós-Condição: Sistema regista jogador dentro do campeonato escolhido

Fluxo Normal:

- 1- Sistema apresenta todos os campeonatos existentes
- 2- Jogador seleciona um campeonato
- 3- Sistema verifica o campeonato e regista o jogador nesse campeonato

Fluxo Exceção 1 [não existem campeonatos] [passo 1]:

- 1.1 - Não existem campeonatos
- 1.2 - Sistema não consegue registar o jogador em nenhum campeonato



Fluxo Exceção 2 [campeonato lotado] [passo 3]:

3.1 Sistema verifica que campeonato está lotado

3.2 Sistema não consegue registar o jogador no campeonato escolhido

### UC - Escolher Piloto

Cenário: Cenário 4

Pré-condição: Jogador autenticado

Pós-Condição: Sistema regista piloto escolhido

Fluxo Normal:

1-Sistema apresenta todos os pilotos disponíveis

2- Jogador escolhe o piloto

3 - Sistema regista o piloto selecionado

Fluxo Exceção 1 [não existem pilotos disponíveis] [passo1]:

1 - Não existem pilotos

2 - O sistema não regista nenhum piloto escolhido

### UC- Escolher Carro

Cenário: Cenário 3

Pré-condição: Jogador autenticado

Pós-Condição: Sistema regista o carro escolhido pelo jogador

Fluxo Normal:

1-Sistema apresenta todos os carros disponíveis

2- Jogador escolhe o carro

3 - Sistema regista o carro selecionado



Fluxo Exceção 1 [não existem carros disponíveis] [passo1]:

- 1 - Não existem carros
- 2 - O sistema não regista nenhum carro escolhido

### UC - Escolher pneu

Cenário: Cenário 5

Pré-condição: Jogador autenticado

Pós-Condição: Sistema regista pneu escolhido

Fluxo Normal:

- 1-Sistema apresenta todos os pneus disponíveis
- 2- Jogador escolhe o pneu
- 3 - Sistema verifica que o pneu é "macio"
- 4- Sistema regista o pneu escolhido

Fluxo alternativo 1 [escolhe pneu duro] [passo 3]:

- 3.1 Sistema verifica que o pneu é "duro"
- 3.2 regressa ao 4

Fluxo alternativo 2 [escolhe pneu chuva] [passo 3]:

- 3.1 Sistema verifica que o pneu é "chuva"
- 3.2 regressa ao 4





### UC - Escolheu tipo de motor

Cenário: Cenário 5

Pré-condição: Jogador autenticado

Pós-Condição: Sistema regista tipo de motor escolhido

Fluxo Normal:

- 1-Sistema apresenta todos os tipos de motores disponíveis
- 2- Jogador escolhe o tipo de motor
- 3 - Sistema verifica que o tipo de motor é "normal"
- 4- Sistema regista o pneu escolhido

Fluxo alternativo 1 [escolhe motor conservador] [passo 3]:

- 3.1 Sistema verifica que o tipo de motor é "conservador"
- 3.2 regressa ao 4

Fluxo alternativo 2 [escolhe motor agressivo] [passo 3]:

- 3.1 Sistema verifica que o pneu é "conservador"
- 3.2 regressa ao 4

### UC - Jogar Corrida

Cenário: Cenário

Pré-condição: Jogador já está no campeonato

Pós-Condição: Sistema inicia a corrida

Fluxo Normal:

- 1-Jogador indica que está pronto
- 2- Sistema recebe a indicação e verifica que todos os outros jogadores estão prontos
- 3-Sistema inicia a corrida



Fluxo Alternativo1 [jogadores não estão prontos] [passo 2]:

- 2.1 Sistema verifica que todos jogadores não estão prontos
- 2.2 Sistema espera que todos os jogadores fiquem prontos
- 2.3 regressa a 3

- Ambos dispõem das seguintes funções:

#### UC- Registar

Cenário: Cenário 9

Pré-condição: True

Pós-Condição: Utilizador passa a ficar registado

Fluxo Normal:

1. Utilizador insere os dados (email, password e nome de utilizador)
2. Valida os dados inseridos por utilizador
3. Sistema regista novo utilizador

Fluxo Exceção1[email já existe] [passo 2]:

- 2.1 Sistema verifica que email já está a ser utilizado
- 2.2 Sistema cancela registo



## UC - Fazer login

Cenário: Cenário 9

Pré-condição: True

Pós-Condição: O utilizador entra no jogo

Fluxo Normal:

- 1- Sistema pede o nome de utilizador e palavra-passe
- 2- O utilizador indica o seu nome de utilizador e palavra-passe
- 3 - Sistema valida o nome de utilizador e palavra-passe
- 4 - Sistema verifica que é jogador
- 5 -sistema regista que utilizador entrou no jogo

Fluxo Exceção 1 [nome de utilizador não existente] [passo 3]:

- 3.1 Sistema verifica que nome de utilizador não existe
- 3.2 login não é efetuado

Fluxo Exceção 2 [palavra-passe errada] [passo 3]:

- 3.1 Sistema verifica que palavra-passe errada
- 3.2 login não é efetuado

Fluxo alternativo 3 [utilizador é um administrador] [passo 4]:

- 4.1 Sistema verifica que é um administrador
- 4.2 regressa ao 5



# Conclusão

Ao realizar esta fase do projeto, conseguiu-se concluir que esta fase foi útil para perceber e estruturar qual seria a aplicação a desenvolver. Apesar de ser uma fase ainda inicial, permitiu criar já certas ideias para a realização das fases seguintes, permitindo analisar o que irá ser necessário fazer e usar para englobar todas as funcionalidades da aplicação, que é na sua essência um jogo.

O Modelos de Domínio e o Modelo de Use case foram criados analisando os cenários de utilização fornecidos no enunciado, percebe-se a partir do enunciado e em conversa com um dos docente da cadeira, que existem diversas formas de interpretar estes cenários e desenvolver estes modelos, podendo existir outros modelos até mais simples, contudo os modelos apresentados nesta fase foram os que de certa forma pareceram mais eficazes e melhores para o desenvolvimento do projeto, no ponto de vista do grupo.