



UNIVERSIDADE DO MINHO

LICENCIATURA EM ENGENHARIA INFORMÁTICA

**GitHub do Projeto**

Desenvolvimento de Sistemas de Software- Entrega

Intermédia 1

Grupo 24

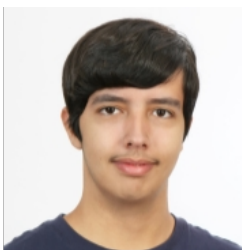
Eduardo Pereira (A94881)

Gonçalo Vale (A96923)

Gonçalo Freitas (A96136)

José Martins (A97903)

Ano Letivo 2022/2023



**URL:**

# Introdução

No âmbito da UC de Desenvolvimento de Sistemas de Software, foi-nos proposta a realização de um jogo inspirado em *F1 Manager*, em que os jogadores podem simular corridas, escolher o seu piloto, carro e campeonato que pretendem concorrer. Para além disso, o jogador poderá ter acesso a uma versão premium do jogo, onde a simulação terá em conta mais variáveis. Quanto à criação de campeonatos, carros, circuitos e pilotos serão feitos ao encargo dos administradores.

## Objetivos da fase

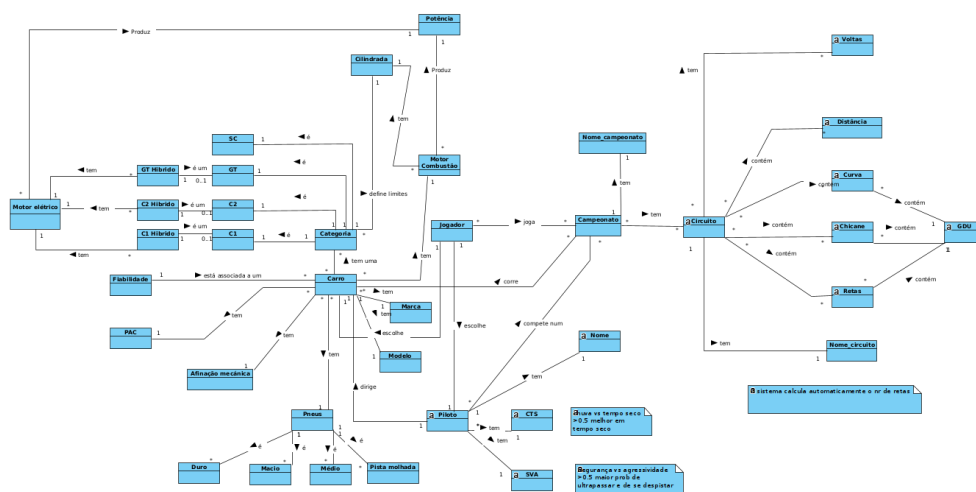
Nesta fase do trabalho foi-nos proposta a realização de dois diagramas (class diagram e use case diagram) com o objetivo de orientar e organizar o nosso projeto de forma a ter uma melhor ideia do que fazer e de como organizá-lo de modo a simplificar as partes mais avançadas. Juntamente com o diagrama de use case revimos os vários cenários possíveis e as eventuais exceções e cenários alternativos, tudo isto para conseguir a melhor preparação possível para um trabalho bem construído.

### Descrição da fase

### Diagrama de classes

No âmbito do desenvolvimento da aplicação proposta, e com o objetivo de criar um plano de desenvolvimento baseado em modelos, criamos, nesta fase de análise de requisitos, um modelo de domínio, assim como um modelo de use case e a sua respetiva lista de exceções.

De forma a cumprir os objetivos da fase propostos, a equipa de trabalho juntou-se e começou por desenvolver o modelo de domínio.

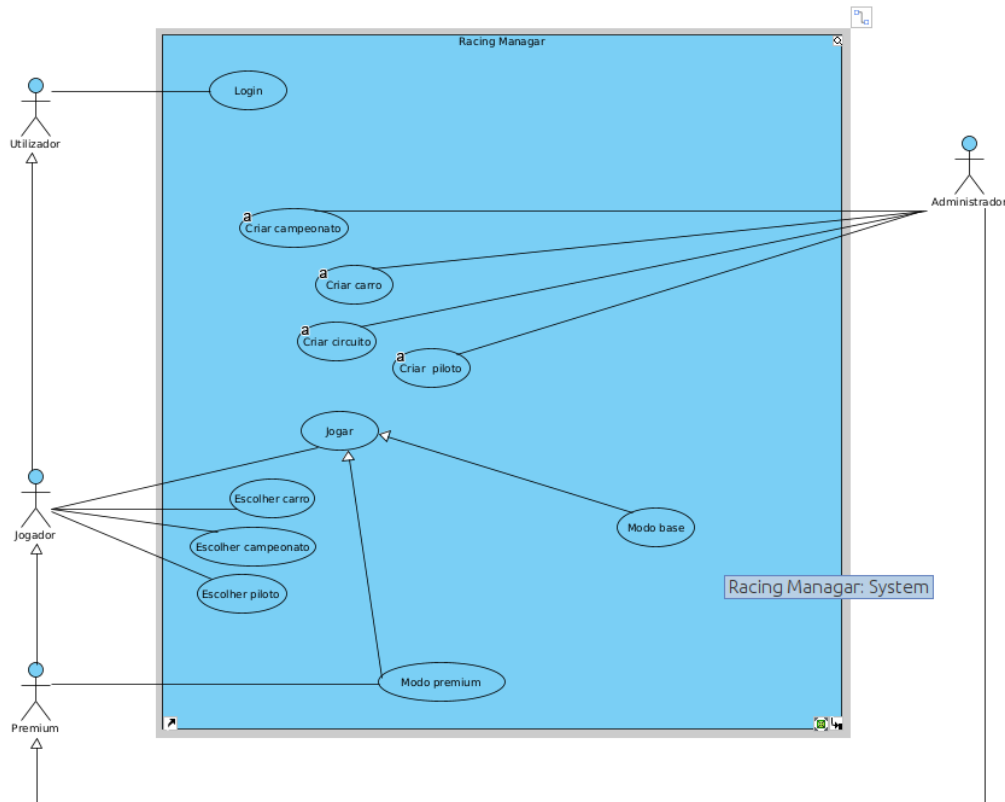


Usando a classe **Jogador** como ponto de partida, o jogador pode fazer 3 escolhas, logo 3 ligações: Piloto, Carro e Campeonato.

- **Campeonato:** O jogador escolhe um campeonato da lista de campeonatos disponíveis para jogar; o campeonato está associado a NomeCampeonato e Circuito.
  - **NomeCampeonato:** O nome do campeonato serve como identificador deste.
  - **Circuito:** O campeonato é composto por um ou mais circuitos; o Circuito está associado a Voltas, Distância, Curva, *Chicane*, Retas e NomeCircuito.
    - \* **Voltas:** A cada circuito está associado um numero de voltas que cada corrida deve ter.
    - \* **Distância:** Cada circuito tem uma distância de inicio ao fim, dado pelo administrador.
    - \* **Curva:** Cada circuito tem um numero de curvas, dado pelo administrador. Está associado ao GDU.
    - \* **Chicane:** Cada circuito tem um numero de *chicane*, dado pelo administrador. Está associado ao GDU.
    - \* **Retas:** Cada circuito tem um numero de retas, calculado pelo sistema através da distância, numero de curvas e *chicane*. Está associado ao GDU.
    - **GDU:** Grau de dificuldade de ultrapassagem associado a cada curva, *chicane* e reta e representa a dificuldade de um carro ultrapassar o carro imediatamente á sua frente. Pode ser possível, impossível ou difícil.
    - \* **NomeCircuito:** O nome do circuito serve como identificador deste.
- **Carro:** Ao carro estão atribuídas as seguintes características: Categoria, Fiabilidade, PAC, Afinação, Tipo de pneus, Modelo, Marca e Motor, e por isso as associações com sentido do carro para cada um dos atributos. Categoria:
  - **Categoria:** Existem 4 categorias: SC, C1, C2 e GT. Apenas as classes C1, C2 e GT podem tem motores híbridos, logo SC não tem ligação a motor elétrico. A categoria também define limites para a cilindrada de cada carro
  - **Fiabilidade, PAC, Afinação, Modelo e Marca:** A cada carro está associado cada um destes atributos, que são pré-definidos pelo administrador, logo as associações diretas.
  - **Pneus:** Existem 4 tipos de pneus, logo 4 ligações.
  - **Motor:** Todos os carros têm motores de combustão, mas os carros das categorias C1, C2 e GT podem ter motores híbridos, logo possuem um motor de combustão e um motor elétrico, e por isso o motor de combustão apresenta associação direta ao carro, ao contrário do motor elétrico que a associação é feita às categorias que o podem possuir. Cada um destes motores produz potência, logo a ligação de ambos.
- **Piloto:** O jogador escolhe um piloto para participar no campeonato a ser simulado; o piloto está associado a Carro, Campeonato, Nome, CTS e SVA.
  - **Carro:** Ao piloto pode atribuído um carro para dirigir num campeonato.
  - **Campeonato:** O piloto pode ser escolhido para participar num campeonato.
  - **Nome:** O piloto possui um nome, usado para identificação.
  - **CTS:** O piloto tem um critério, de 0 a 1, que define se o piloto dirige melhor na chuva ou no piso seco, sendo um menor numero um melhor desempenho em tempo chuvoso e um maior numero, um melhor desempenho em tempo seco.
  - **SVA:** O piloto tem um medidor de segurança vs agressividade, valores que variam de 0 a 1, que é inversamente proporcional à probabilidade de bater.

## Diagrama use case

Ainda no seguimento do cumprimento dos objetivos desta fase, elaboramos um diagrama de use case.



Neste diagrama assumimos a existência de quatro atores a interagirem com o sistema. Sendo eles, um administrador que tem controlo sobre a criação de carros, campeonatos, circuitos e pilotos. O outro, um utilizador que faz *login* no jogo, um jogador que está autorizado a escolher carro, piloto e campeonato, e por fim, jogar, e um *premium*, que será um jogador com mais vantagens no jogo em relação a um jogador normal. A única diferença visível no esquema seria o modo de jogar, que seria avaliado pelo sistema, caso o jogador fosse normal, jogaria no modo normal, senão, jogaria no modo *premium*.

## Fluxos use case

Para completar a informação do diagrama de use case, especificamos cada um deles.

**Use Case:** *Login*

**Cenários:** Cenário 1 – Campeonatos, Cenário 2 – Circuitos, Cenário 3 – Carros, Cenário 4 – Pilotos, Cenário 5 - Jogar

**Pré-Condição:** True

**Pós-Condição:** Utilizador fica autenticado no sistema

### **Fluxo Normal:**

1. Sistema apresenta opções disponíveis.
2. Utilizador escolhe entrar como jogador base.
3. Sistema pede nome de utilizador.
4. Sistema pede palavra passe.

**Fluxo Alternativo 1** [utilizador escolhe entrar como jogador premium] (passo 2)

- 2.1. Regressa a 3.

**Fluxo Alternativo 2** [utilizador escolhe entrar como administrador] (passo 2)

- 2.1. Regressa a 3.

**Fluxo de Exceção 3** [nome de utilizador não existe] (passo 3)

- 3.1. Sistema verifica que nome de utilizador não existe.
- 3.2. Sistema termina processo.

**Fluxo de Exceção 4** [palavra passe incorreta] (passo 3)

- 3.1. Sistema verifica que palavra passe está incorreta.
- 3.2. Sistema termina processo.

**Use Case:** Criar campeonato

**Cenários:** Cenário 1 - Campeonatos

**Pré-Condição:** Utilizador autenticado como administrador

**Pós-Condição:** Sistema fica com mais um campeonato disponível para jogar

### **Fluxo Normal:**

1. Administrador dá um nome ao campeonato.
2. Administrador escolhe um ou mais circuitos da lista de circuitos disponíveis.
3. Sistema adiciona o campeonato à lista de campeonatos disponíveis.

**Fluxo de Exceção 1** [nome dado já existe] (passo 1)

- 1.1. Sistema verifica que o nome dado já existe.
- 1.2. Sistema termina o processo.

**Use case:** Criar Circuito

**Cenários:** Cenário 2 – Circuitos

**Pré-Condição:** Utilizador autenticado como administrador

**Pós-Condição:** Sistema fica com mais um circuito disponível para integrar campeonatos

**Fluxo Normal:**

1. O administrador indica o nome que pretende atribuir ao circuito.
2. O administrador indica o comprimento do circuito, o número de curvas chicanes.
3. O Sistema calcula o número de retas e apresenta a lista de curvas e retas.
4. O administrador indica o GDU em cada uma (curvas, retas e chicanes).
5. O administrador indica o número de voltas do circuito.
6. O Sistema coloca o circuito disponível para integrar campeonatos.

**Fluxo de Exceção 1** [nome dado já existe] (passo 1)

- 1.1. O Sistema verifica que já existe um circuito com o nome escolhido pelo administrador.
- 1.2. Sistema termina o processo.

**Fluxo de Exceção 2** [o GDU indicado pelo administrador é inválido] (passo 4)

- 4.1. O Sistema verifica que o GDU introduzido é inválido.
- 4.2. Sistema termina o processo.

**Fluxo de Exceção 3** [comprimento indicado pelo administrador é 0] (passo 2)

- 2.1. O sistema verifica que o comprimento é inválido.
- 2.2. Sistema termina o processo.

**Fluxo de Exceção 4** [número de voltas indicado pelo administrador é 0] (passo 5)

- 5.1. O sistema verifica que o número de voltas é inválido.
- 5.2. Sistema termina processo.

**Use Case:** Registrar carro

**Cenários:** Cenário 3 - Carros

**Pré-Condição:** Jogador 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 potência.
3. Sistema verifica que carro é C1 e que necessita de fiabilidade (pode ser híbrido).
4. Administrador indica fiabilidade.
5. Sistema verifica que fiabilidade é aproximadamente 95%.
6. Administrador indica que não é híbrido.
7. Administrador indica PAC.
8. Sistema regista carro.

**Fluxo Alternativo 1** [carro é C2] (passo 3)

- 3.1. Sistema verifica que carro é C2.
- 3.2. regressa a 6.

**Fluxo Alternativo 2** [carro é GT] (passo 3)

- 3.1. Sistema verifica que carro é GT.
- 3.2. regressa a 6.

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

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

**Fluxo de Exceção 4** [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 5** [carro é híbrido] (passo 6)

- 6.1. Administrador indica que é híbrido e indica potência do motor elétrico.
- 6.2. Regressa a 7.

**Fluxo de Exceção 6** [administrador indica que carro é SC e híbrido] (passo 6)

- 6.1. Administrador indica que é híbrido.
- 6.2. Sistema termina processo.

**Fluxo de Exceção 7** [administrador indica que carro é C2 e fiabilidade abaixo de 80%] (passo 5)

- 5.1. Sistema verifica que fiabilidade é inferior a 80%.
- 5.2. Sistema termina processo.

**Use Case:** Criar piloto

**Cenários:** Cenário 4 - Pilotos

**Pré-Condição:** Jogador autenticado

**Pós-Condição:** Sistema fica com mais um piloto disponível para jogar

**Fluxo Normal:**

1. Administrador indica o nome.
2. Administrador indica os níveis de perícia.
3. Piloto fica disponível.

**Fluxo de Exceção 1** [nome do piloto já existe] (passo 1)

- 1.1. Sistema verifica que nome já existe.
- 1.2. Sistema termina processo.

**Fluxo de Exceção 2** [níveis de perícia não estão entre 0 e 1] (passo 2)

- 2.1. Sistema verifica que valores de CTS e SVA não estão entre 0 e 1.
- 2.2. Sistema termina processo.

**Use Case:** Escolher carro

**Cenários:** Cenário 5 - Jogar

**Pré-Condição:** Autenticado

**Pós-Condição:** Jogador com o carro selecionado

**Fluxo normal:**

1. Jogador seleciona a categoria de carros a competir.
2. Jogador seleciona o carro da lista de carros.
3. Jogador escolhe o tipo de pneus a utilizar.

**Use Case:** Escolher Piloto

**Cenários:** Cenário 5 - Jogar

**Pré-Condição:** Autenticado

**Pós-Condição:** Jogador com o piloto selecionado

**Fluxo normal:**

1. O administrador escolhe o piloto.

**Use Case:** Escolher Campeonato

**Cenários:** Cenário 5 - Jogar

**Pré-Condição:** Autenticado

**Pós-Condição:** Jogador com o campeonato selecionado

**Fluxo normal:**

1. O administrador escolhe o campeonato.

**Use Case:** Jogar

**Cenários:** Cenário 5 - Jogar

**Pré-Condição:** Autenticado e variáveis selecionadas

**Pós-Condição:** Simulação iniciada

**Fluxo normal:**

1. Sistema verifica que jogador é base.
2. Sistema adapta-se a modo normal.
3. Simulação inicia.

**Fluxo alternativo 1** [jogador é *premium*] (passo 1)

- 1.1. Sistema verifica que jogador é *premium*.
- 1.2. Sistema adapta-se a modo *premium*.
- 1.3. Regressa a 3.



## Conclusão

Neste ponto damos por concluída a primeira fase deste trabalho. Nesta fase, foi-nos possível elaborar os diagramas da modelagem de domínios e de use cases, assim como a especificação dos últimos, o que nos facilitará bastante as próximas fases do trabalho.

A próxima fase será a fase de conceção da solução, onde nos competirá elaborar uma arquitetura conceptual do sistema capaz de suportar os requisitos identificados e os modelos comportamentais necessários para descrever o comportamento pretendido para o sistema.