



# **Desenvolvimento de Sistemas de Software**

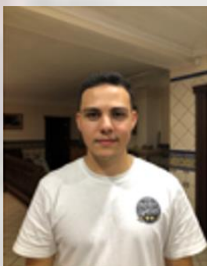
## **- Trabalho Prático -**

### **- Jogo Formula 1 Manager (Fase 2) -**

**Universidade do Minho**

**Licenciatura em Engenharia Informática**

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



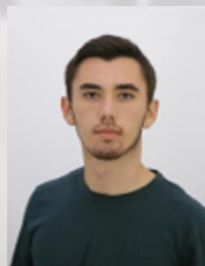
- A95442 -

André Oliveira Gonçalves



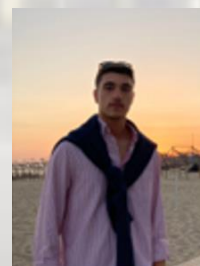
- A95460 -

Diogo Luís Almeida Costa



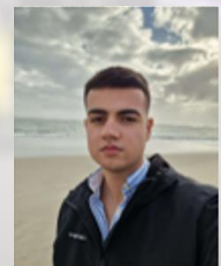
- A95323 -

Henrique Ribeiro Fernandes



- A96726 -

Ivo Miguel Alves Ribeiro



- A95641 -

João Pedro Moreira Brito

Ano Letivo 2022/2023

Submetido a: 20/11/2022



# Índice

<b>Objetivos desta fase .....</b>	<b>3</b>
<b>Modificações efetuadas em relação à primeira fase .....</b>	<b>3</b>
<b>Descrição do trabalho.....</b>	<b>4</b>
<b>Diagrama de Componentes .....</b>	<b>5</b>
<b>Diagrama de Classes .....</b>	<b>6</b>
<b>Diagramas de Sequências.....</b>	<b>7</b>
3.1 – Diagrama de Sequência fazLogin() .....	7
3.2 - Diagrama de Sequência registaPlayerSet() .....	7
3.4 – Diagrama de Sequência turnReady() .....	8
3.3 – Diagrama de Sequência registaJogador().....	8
3.6 – Diagrama de Sequência trocaAgressividade() .....	9
3.5 – Diagrama de Sequência trocaPneu().....	9
3.8 – Diagrama de Sequência criaPiloto() .....	10
3.7 – Diagrama de Sequência criaCarro() .....	10
3.9 – Diagrama de Sequência criaCircuito() .....	11
3.11 – Diagrama de Sequência criaCampeonato .....	12
3.10 – Diagrama de Sequência registaAdmin().....	12
<b>Conclusão.....</b>	<b>13</b>

## Repositório do trabalho:

<https://github.com/jpmbrito123/DSS>



## Objetivos desta fase

Para esta segunda fase do projeto foi-nos proposto uma estruturação de uma arquitetura conceptual capaz de suportar os requisitos que identificamos na primeira fase deste projeto. Para além disso fizemos a estruturação dos modelos comportamentais que descrevem o comportamento pretendido para o sistema.

## Modificações efetuadas em relação à primeira fase

Alterou-se o Use Case: Criar carro para:

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 potência
- 3 - Sistema calcula fiabilidade
- 4 - Administrador indica que é não híbrido
- 5 - Administrador indica PAC
- 6 - Sistema regista carro

Fluxo alternativo 1 [carro é SC] [passo3]

- 3.1 Sistema verifica que carro é SC
- 3.2 regressa a 5

Fluxo alternativo 2 [carro é híbrido] [passo 6]

- 4.1 Administrador indica que é híbrido e a potencia do motor elétrico
- 4.2 regressa a 5



Nos Use Cases: Escolher Campeonato, Escolher Piloto e Escolher Carro alterou-se o cenário deles para 5.

No Use Case Jogar Corrida, Registrar e Fazer Login mudou-se o cenário para 5.

## **Descrição do trabalho**

De forma a evoluir o trabalho já alcançado na fase anterior, começou-se por dividir os métodos em vários subsistemas, de forma a ajudar na implementação de cada subsistema.

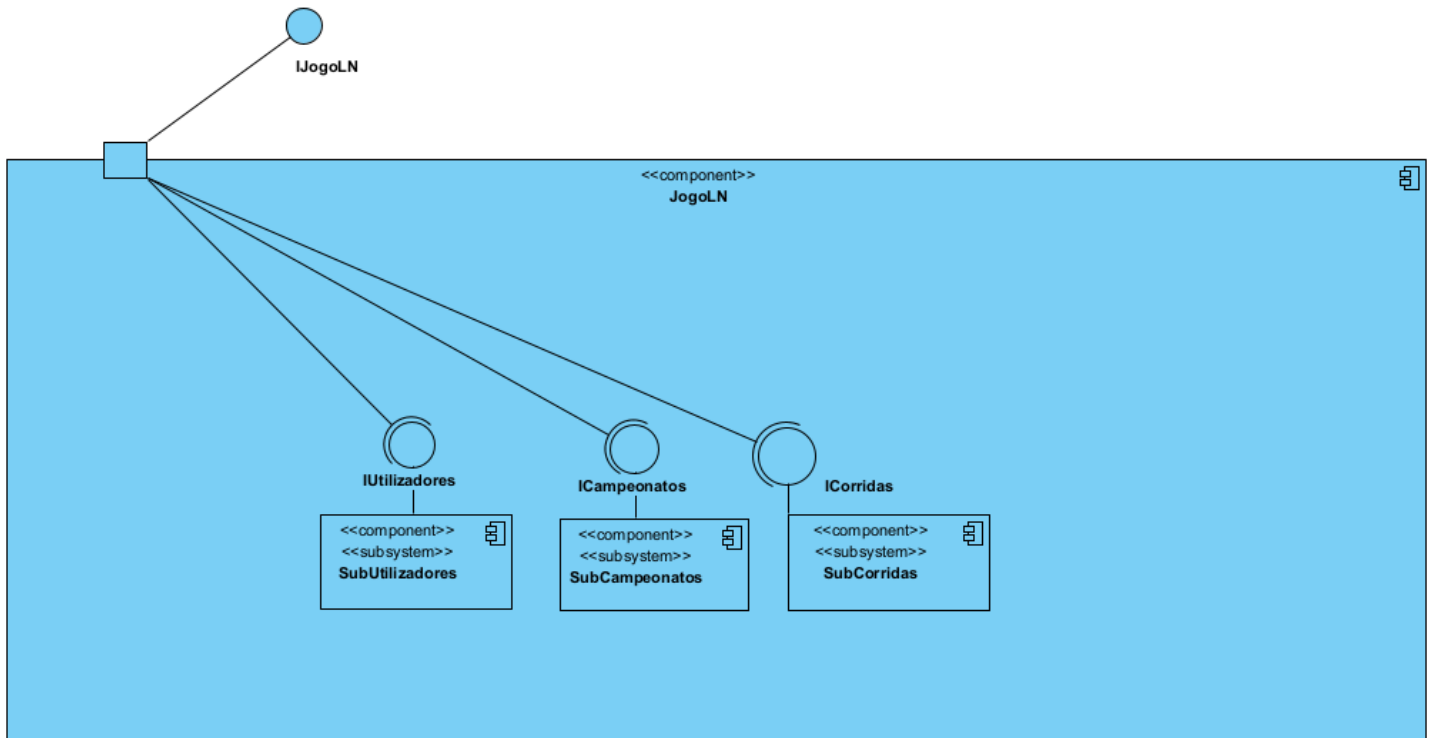
De seguida, fez-se um Diagrama de Componentes para mostrar as relações existentes entre os subsistemas e entre a interface principal da lógica de negócio.

Depois, desenvolveu-se o Diagrama de Classes, visto que qualquer alteração neste diagrama iria alterar os diagramas de Sequência começou-se por realizar este diagrama. Este diagrama fez-se de forma a ser capaz de responder a todas as necessidades existentes durante a execução da aplicação.

Por último, realizou-se os diversos Diagramas de Sequência, tendo em conta o Diagrama de Classes realizado anteriormente, estes diagramas permitiram perceber melhor as necessidades de armazenamento na aplicação e consequentemente ajudam na realização de uma aplicação mais consistente e fiável.



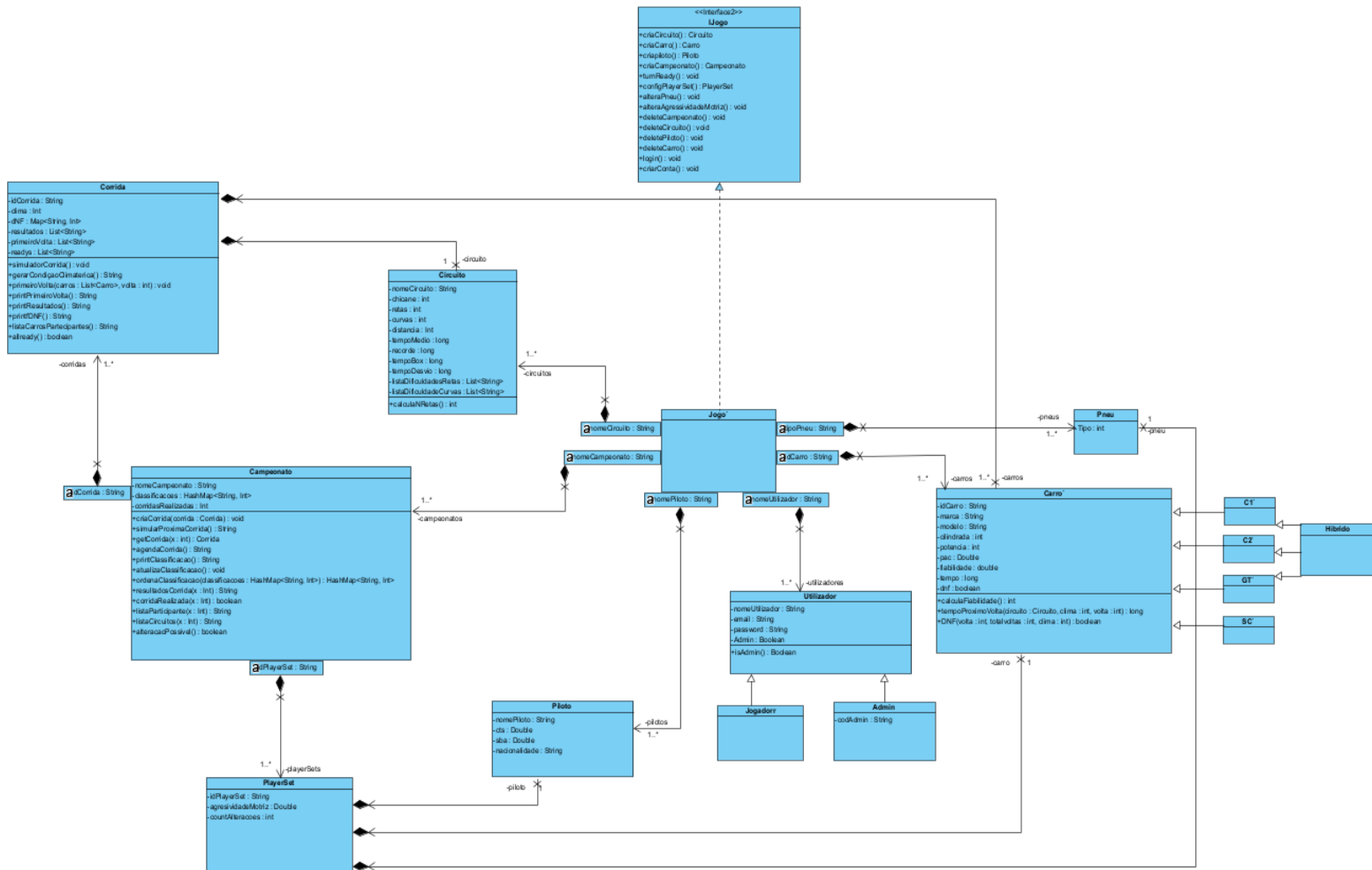
## Diagrama de Componentes



1 – Diagrama de Componentes



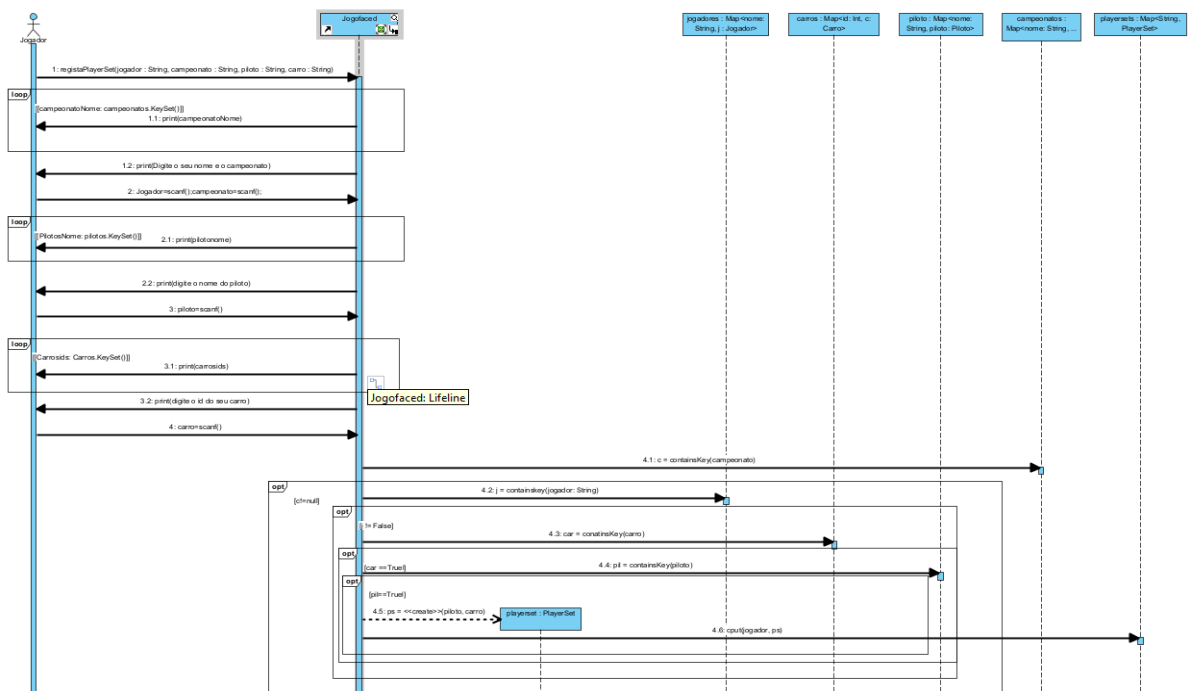
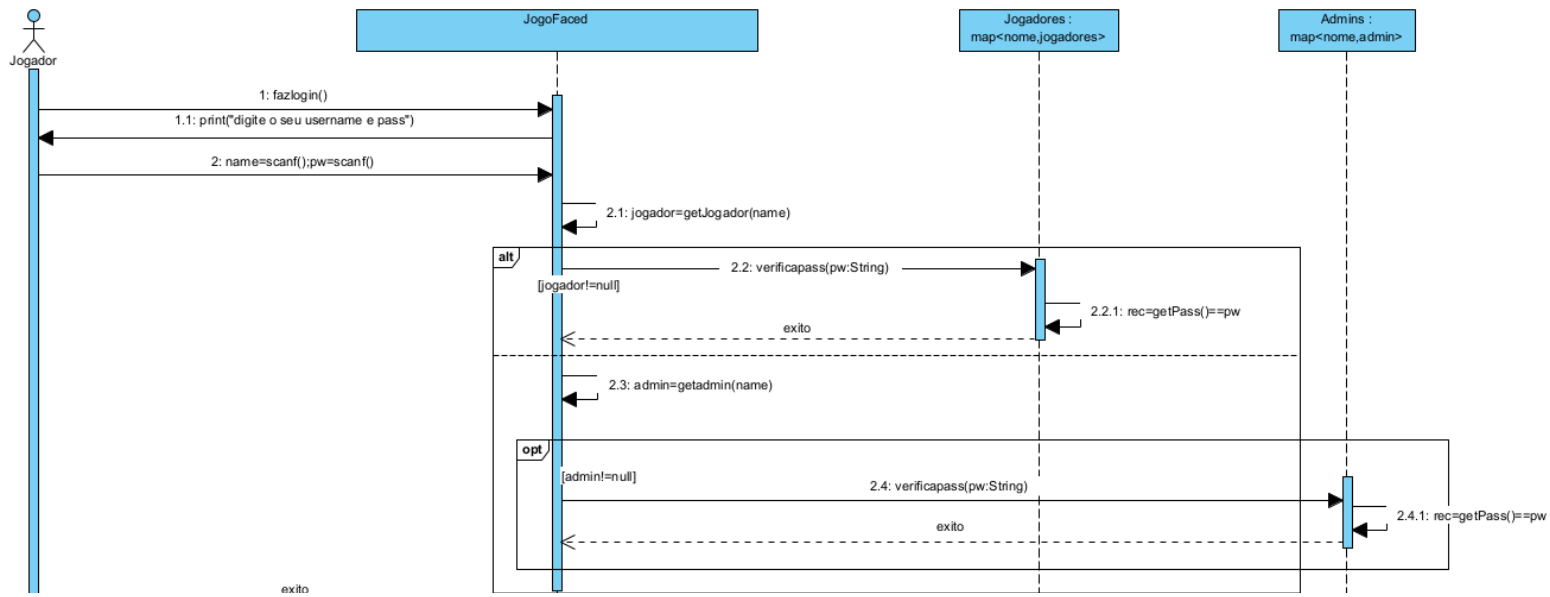
## Diagrama de Classes

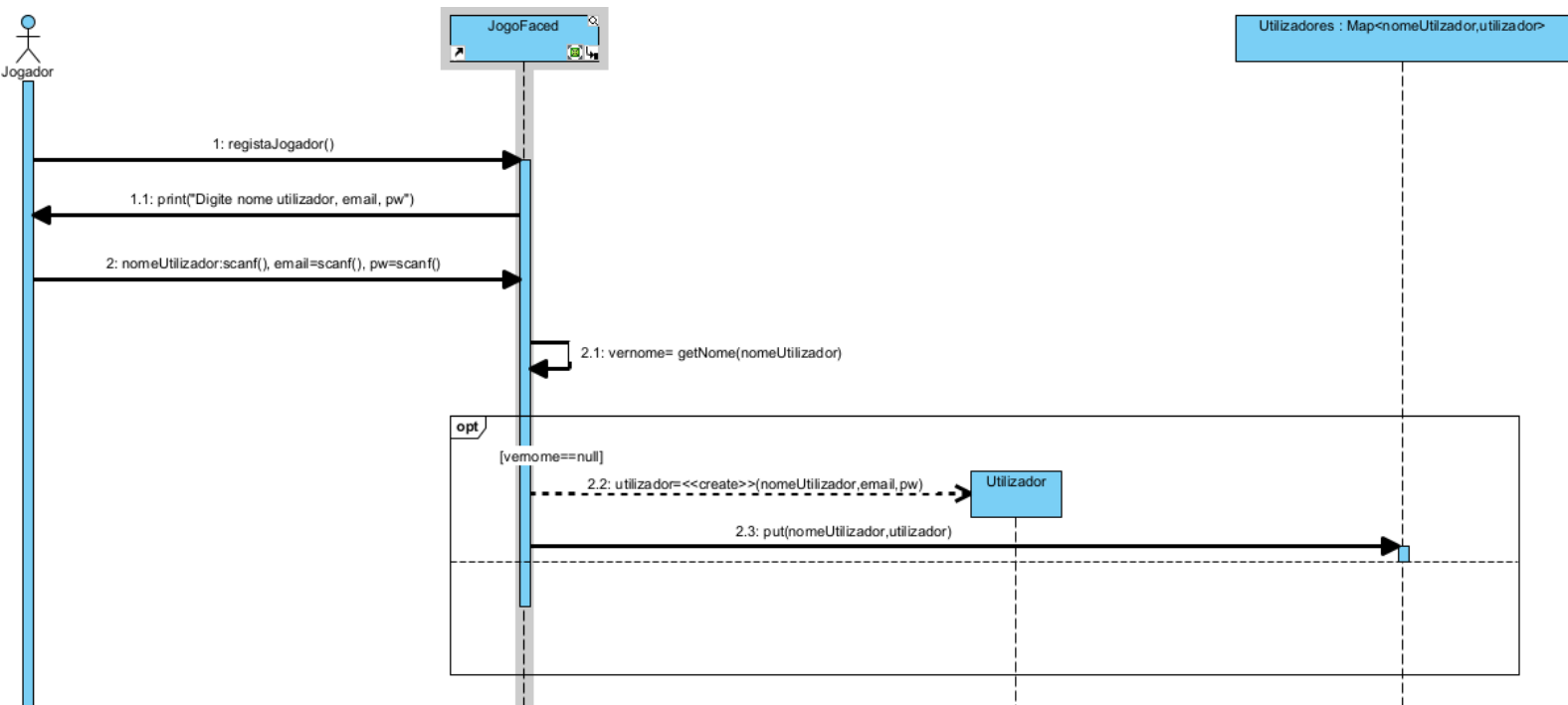


2 – Diagrama de Classes

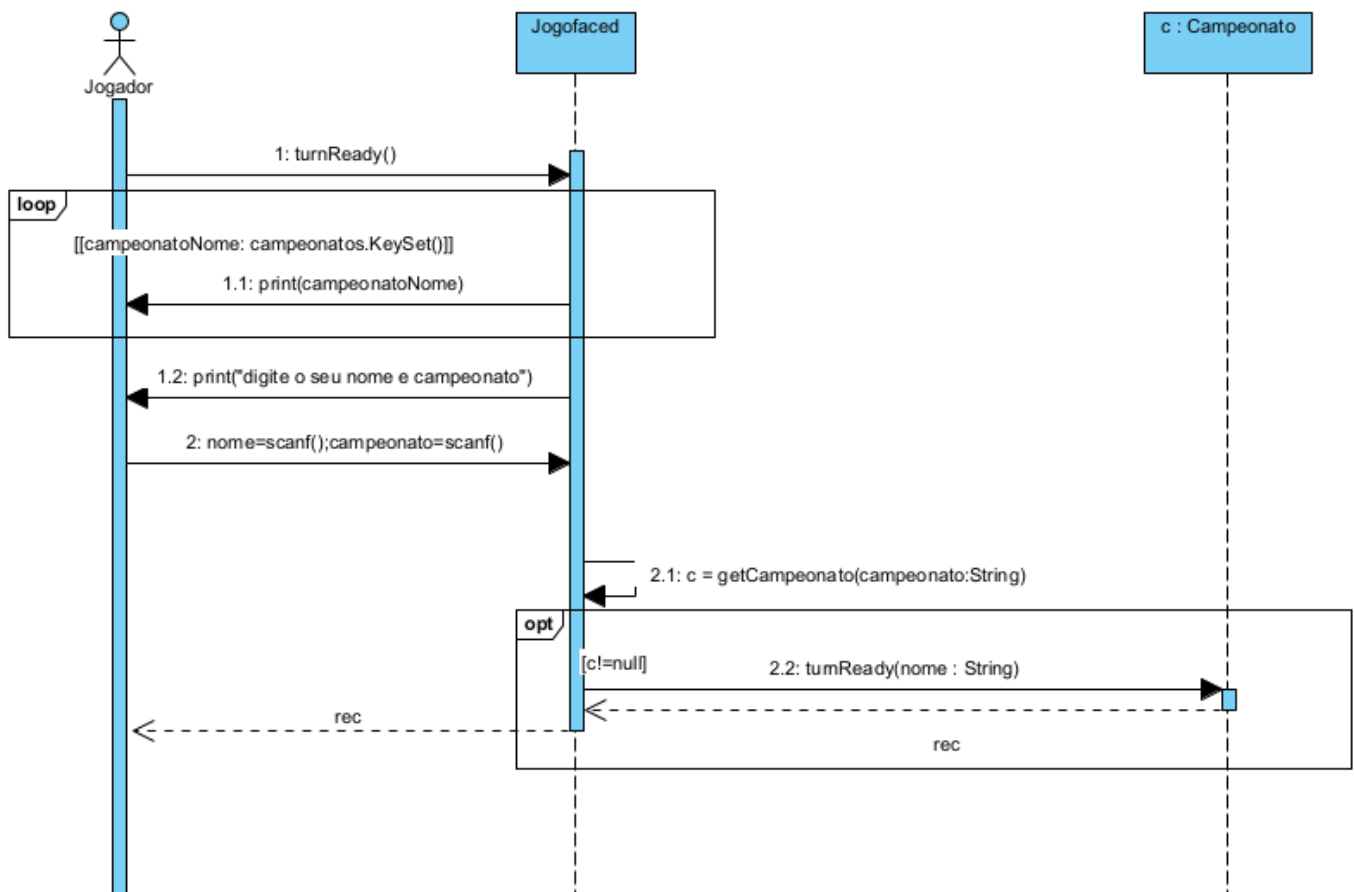


## Diagramas de Sequências



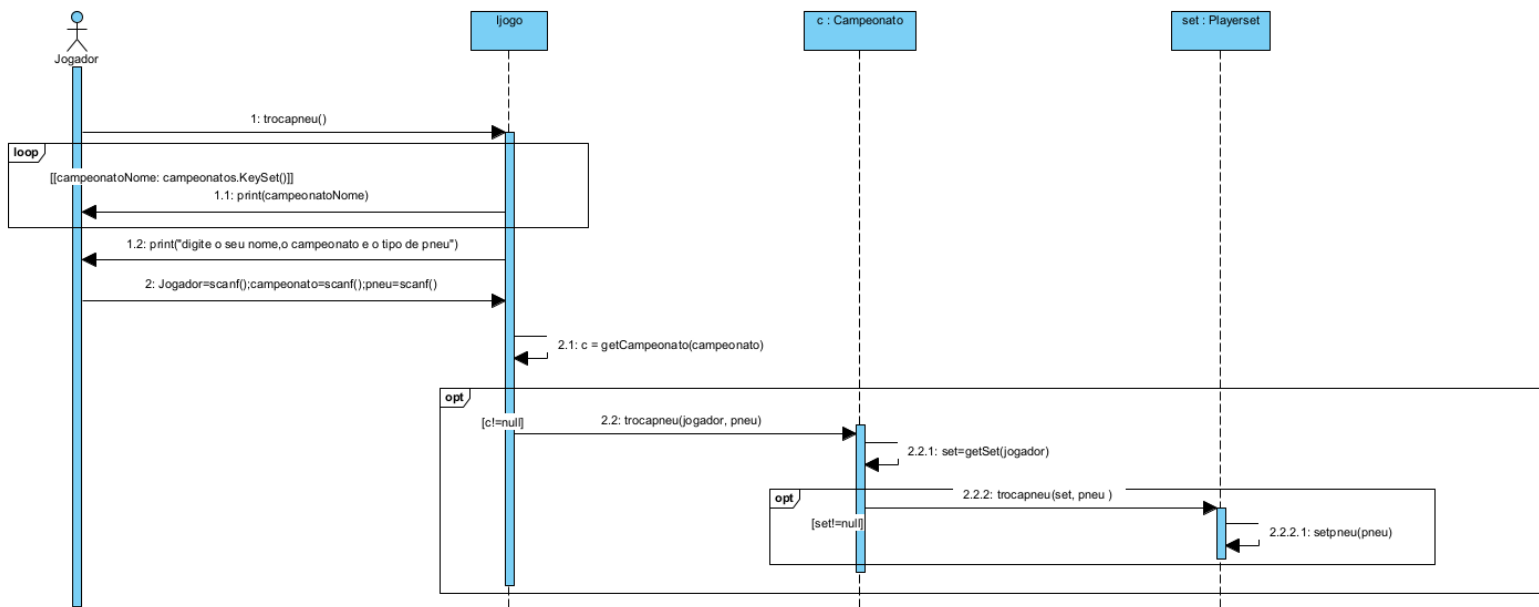


3.3 – Diagrama de Sequência registaJogador()

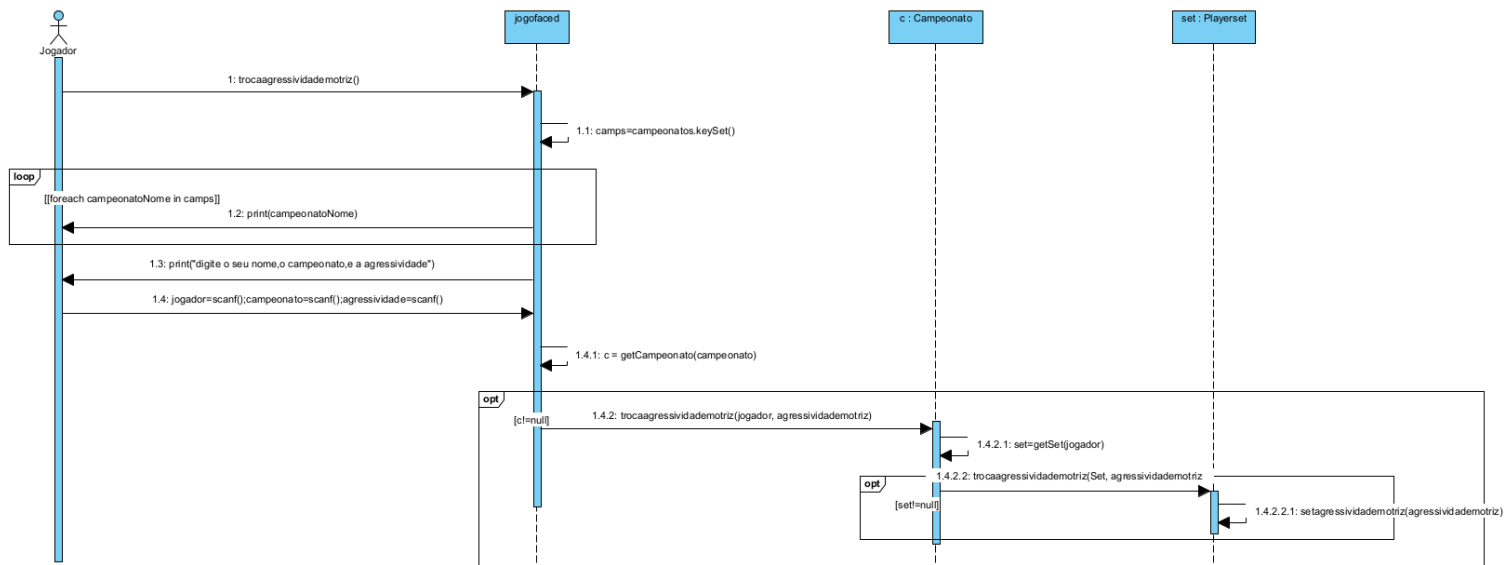


3.4 – Diagrama de Sequência turnReady()

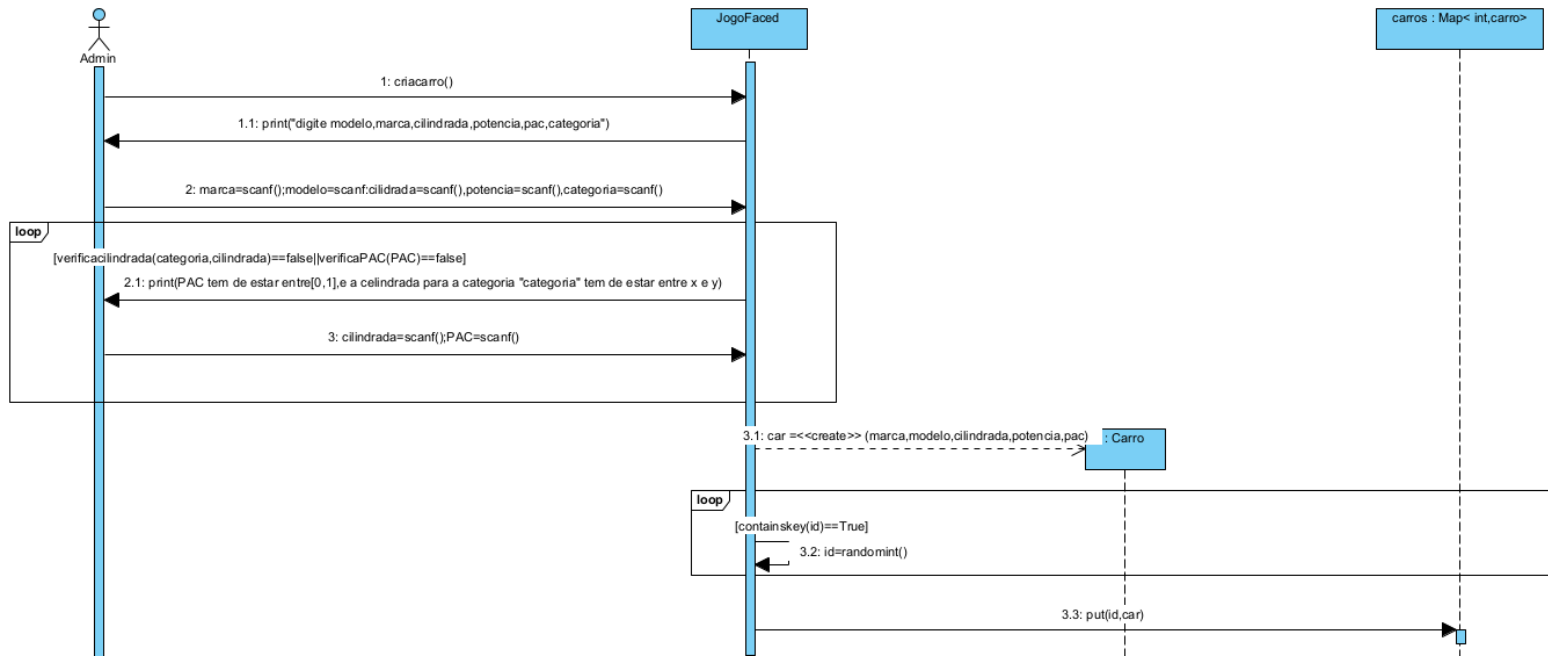




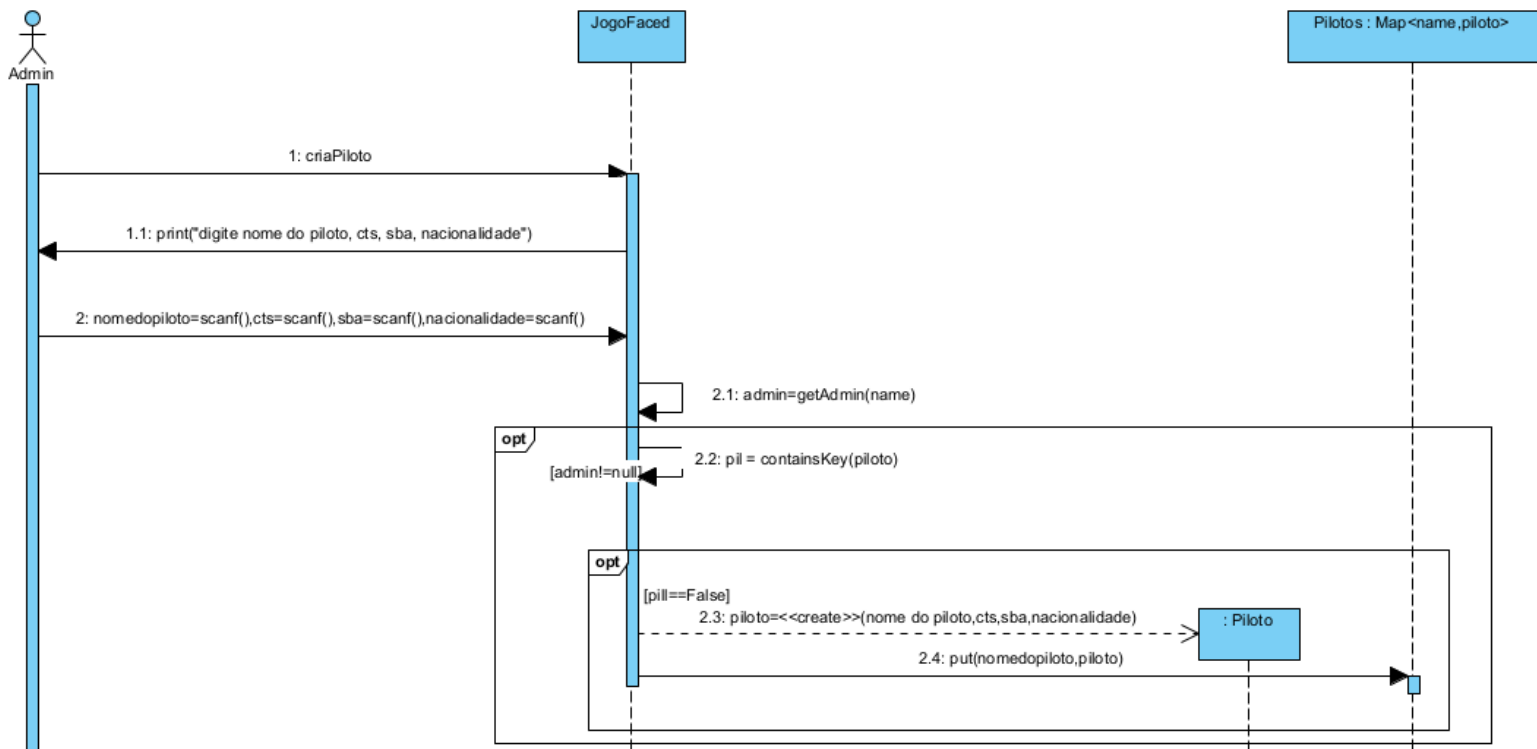
3.5 – Diagrama de Sequência trocaPneu()



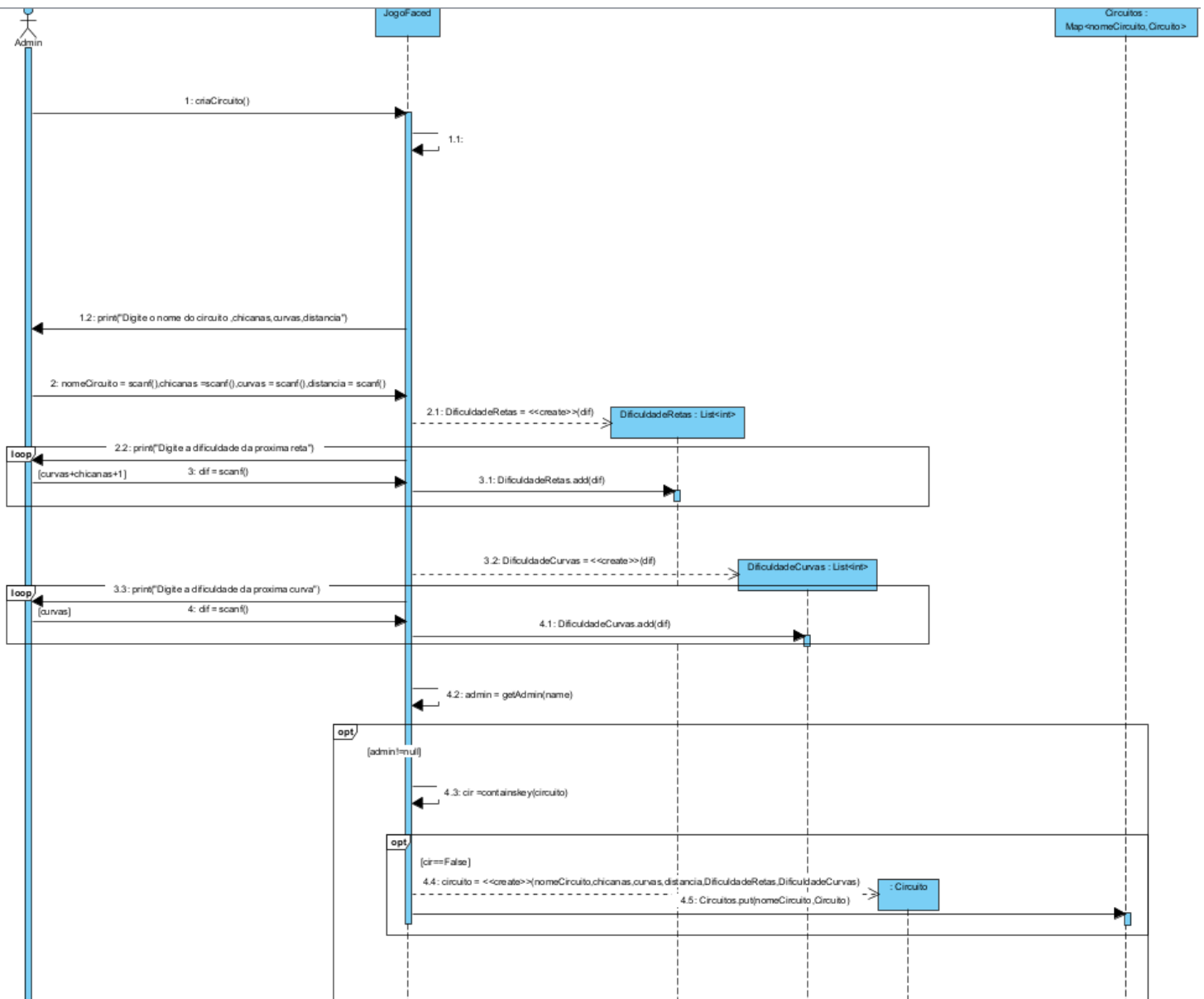
3.6 – Diagrama de Sequência trocaAgressividade()



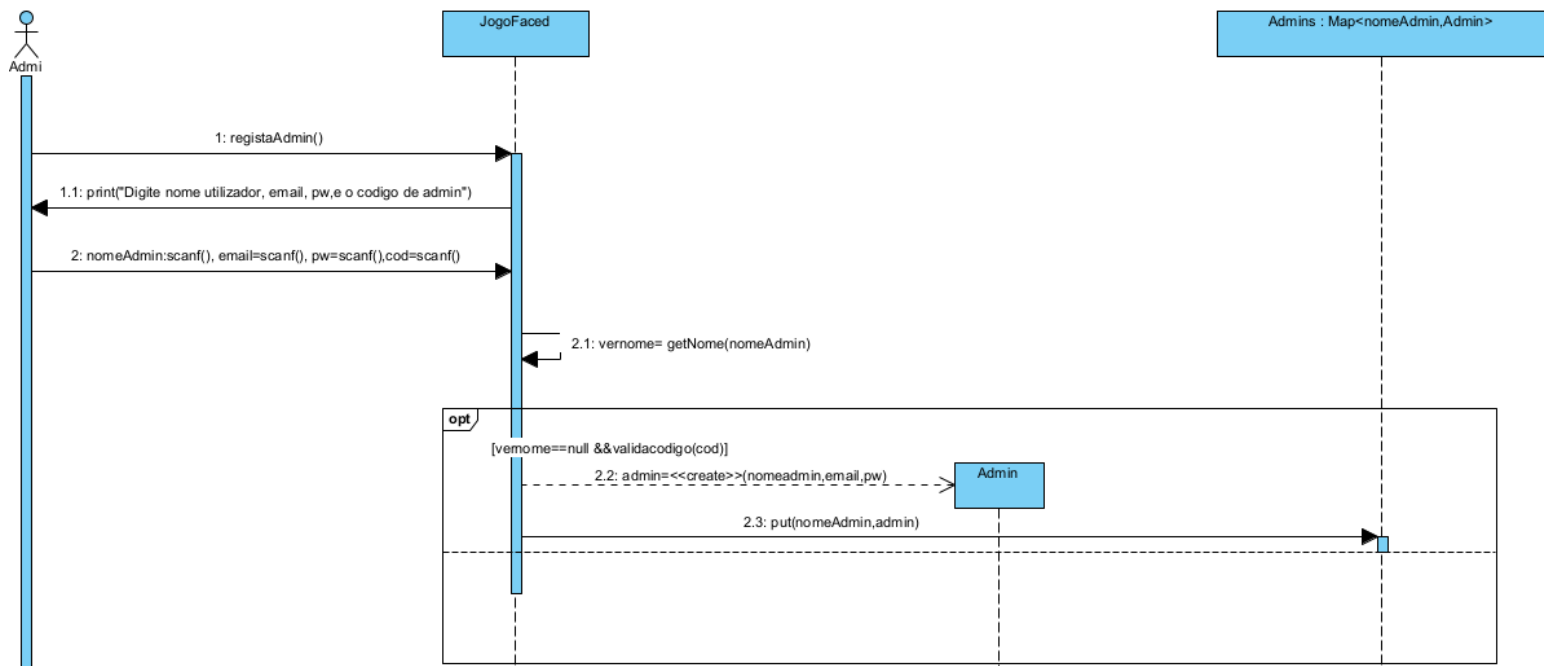
3.7 – Diagrama de Sequência criaCarro()



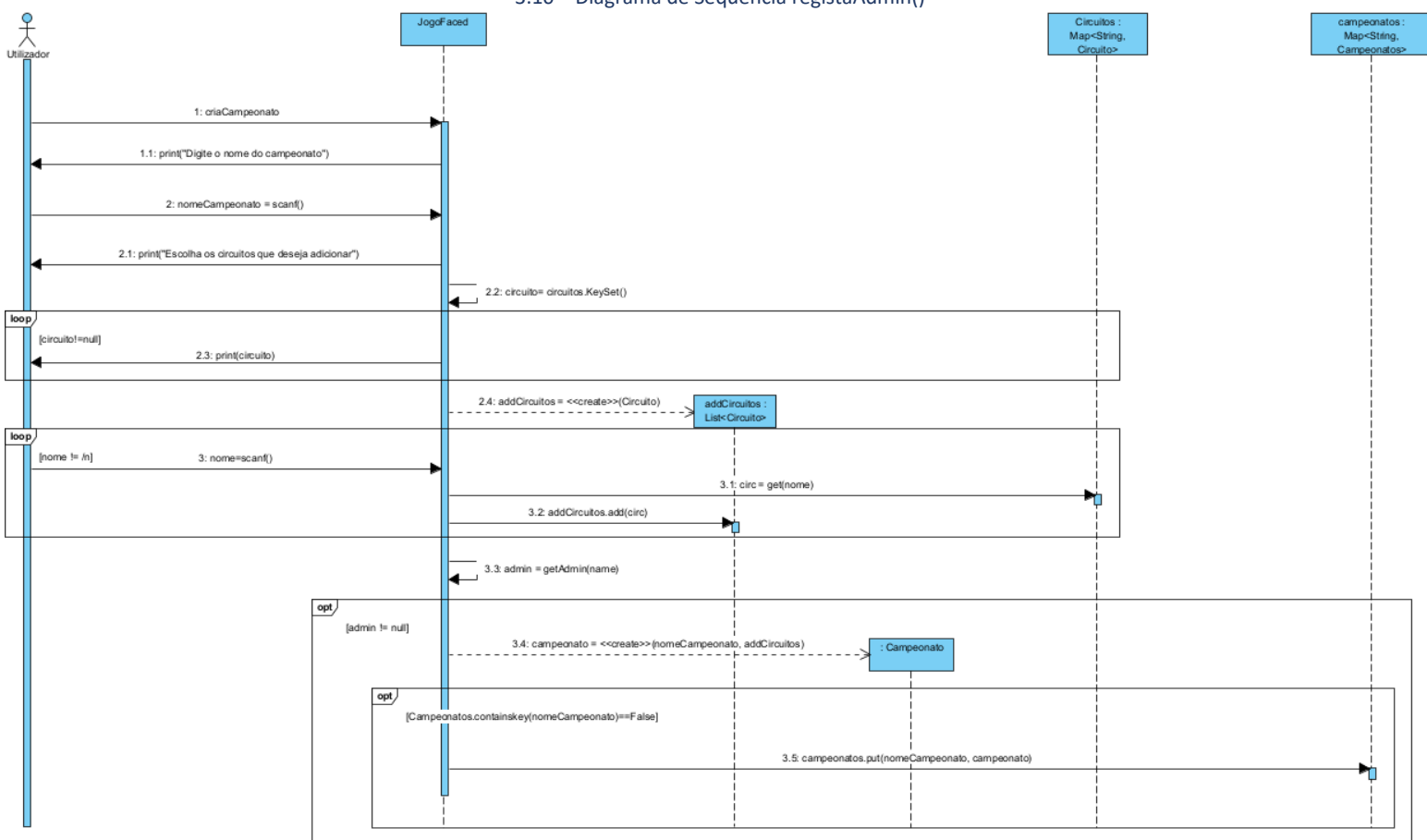
3.8 – Diagrama de Sequência criaPiloto()



3.9 – Diagrama de Sequência criaCircuito()



3.10 – Diagrama de Sequência registaAdmin()



3.11 – Diagrama de Sequência criaCampeonato



# Conclusão

Com a realização desta fase do projeto, podemos concluir que o planeamento foi realizado de forma correta e satisfaz todos os requisitos necessários para a realização da aplicação final.

Em suma, esta fase permitiu obter uma melhor perspetiva da importância de um planeamento na realização de projetos e de que os vários diagramas lecionados nas aulas da UC são realmente úteis na especificação do código a ser implementado e permitem explicar toda a estrutura da aplicação a desenvolver na próxima fase.