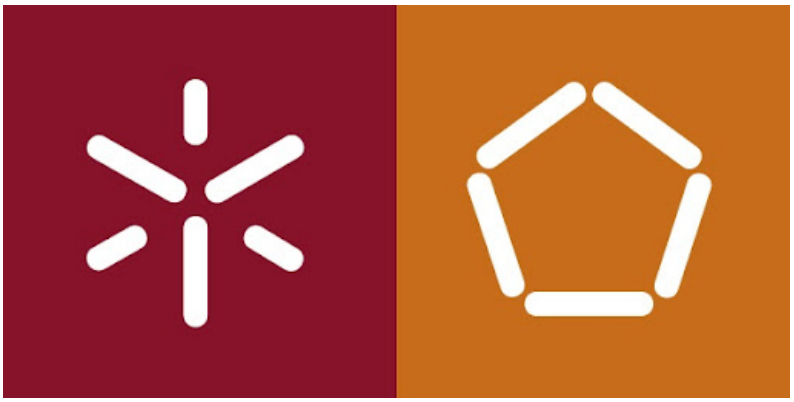
Universidade do Minho

Departamento de Informática
Curso : MIEI / LEI



Guião 2 - Relatório de DSS

Grupo nº2

Alexandre Eduardo Vieira Martins A93242

Guilherme Rodrigues do Outeiro Cunha Marques A94984

Hugo dos Santos Martins A95125

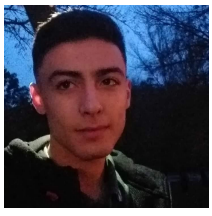
José Eduardo da Cunha Rocha A97270

João Bernardo Teixeira Escudeiro A96075

https://github.com/Eduard0Rocha/ProjetoDSSGrupo2

19 de outubro de 2022

Hugo Martins



José Rocha

João Escudeiro

Alexandre Martins

Guilherme Marques

Índice

Introdução

Para esta entrega intermediária foi nos dado como tarefa a criação de uma arquitetura concetual do sistema e a criação de diagramas comportamentais. Abordamos então estas tarefas da seguinte forma: Criamos 5 subsistemas, campeonato, piloto, carro, circuito e user e um diagrama de componentes que engloba todos os subsistemas. Fizemos também 5 diagramas de sequência com base no cenário 5 - Jogar.

Objetivos

Para esta fase do projeto, o objetivo proposto foi desenvolver a conceção do sistema que se suportaria na arquitetura concetual do sistema e nos modelos necessários para descrever o comportamento pretendido do sistema em questão.

A arquitetura concetual do sistema, na prática, seriam os diagramas de classes que descrevem a estrutura e modularidade do código a desenvolver na próxima fase.

Os modelos comportamentais, na prática, seriam os diagramas de sequência que criam uma imagem gráfica das interações entre os componentes de software do nosso sistema e os utilizadores.

Diagrama de classes

Após uma análise detalhada dos “Use Case” especificada na primeira fase do projeto, observou-se um grande destaque de métodos relativos a cada uma destas entidades : Circuitos , Campeonatos , Usuários , Pilotos e Carros . Assim , optou-se por separar o código em 5 packages , cada um com o seu respectivo Subsistema.

Subsistema Piloto

Este subsistema contém as classes Piloto bem como PilotoFacade ,esta última que implementa a interface “SGestPiloto” que aborda todas as operações a serem realizadas relativamente a pilotos como criar piloto , remover piloto , alterar níveis de confiança entre outros. Resumidamente o subsistema piloto permite fazer alterações aos pilotos, alterações essas que podem ser bastante significativas nas corridas.

Subsistema Carro

Este subsistema contém as classes Carro, ligada a uma classe Pneu que contém informações relativas aos pneus do veículo, bem como CarroFacade que dispõe de uma coleção (Map<String,Carro>) que possui a coleção de Carros existentes no sistema, esta última que implementa a interface “SGestCarro” aborda todas as operações a serem realizadas relativamente a Carros como criar carro,alterar pneus, escolher o tipo de carro (C1, GT, SC e C2) e escolher se este é hibrido ou não.

Subsistema Campeonato.

Este subsistema contém as classes Campeonato bem como CampeonatoFacade, Corrida, Registo e utiliza as classes Circuito (relativo ao subsistema SGestCircuito), Jogador (relativo ao subsistema SGestUser), Piloto (relativo ao subsistema SGestPiloto) e Carro

(relativo ao subsistema SGestCarro). É neste subsistema que o jogador pode indicar as afinações a efetuar no carro e que são tratadas as classificações do campeonato e globais. Para auxílio disto temos uma classe registo que vai incorporar o carro, piloto e jogador numa só classe, e vai ainda guardar o número de afinações já efetuadas. Cada registo é associado a um único jogador.

Subsistema Users.

Este subsistema contém as classes relativas aos utilizadores do sistema .Assim , para além da classe Facade que implementa a interface capaz de suportar as operações relativas à coleção (Map<String,User>) , contém a classe jogador , em que um jogador pode ser guest ou authenticated player, ou então pode ser um Administrador , que tem associadas as suas credenciais guardados numa classe auxiliar.

Subsistema Circuitos

Este subsistema contém as classes relativas aos circuitos do sistema .Assim , para além da classe Facade que implementa a interface capaz de suportar as operações relativas à coleção (Map<String,Circuito>) , contém a classe Circuito , que tem associado uma classe correspondente à condição atmosférica, uma lista de retas , curvas e chicanes , cada uma com o respetivo GDU e uma lista de DRS , contendo as retas em que é permitido DRS.

Desta forma , estão satisfeitos todos os métodos identificados após a leitura detalhada dos requisitos, auxiliando-nos do método lecionado nas aulas práticas (recorrendo a um ficheiro excel).

Surgem em anexo os screenshots , bem como os ficheiros .vpp respectivos aos subsistemas.

Diagrama de Componentes

O diagrama de componentes auxilia na percepção da forma como se interligam as interfaces com as classes, bem como com a lógica de negócio e a base de dados. Assim, considerando os subsistemas já descritos ao pormenor no diagrama de classes este interliga os diagramas de classe possibilitando os seus mútuos acessos através de uma interface.

Diagramas de sequências

Um diagrama de sequências, como o próprio nome diz, apresenta uma sequência de eventos que determina o comportamento de um Use Case.

Assim, baseando-se nos Use Case da fase anterior , e após a realização do diagrama de Classes , o grupo optou por fazer alguns diagramas de sequência que suportam as ações a realizar maioritariamente no cenário 5.

A partir destes é nos permitido observar a ordem em que os eventos acontecem, as mensagens que são trocadas, todos os métodos chamados e a maneira como os objetos interagem entre si.

Assim, temos em anexo os seguintes diagramas de sequência:

- **Adicionar Registo** - Para este use Case em que o Jogador apenas seleciona o campeonato , o piloto e o carro , temos a interação entre o jogador e a classe F1Facade , que implementa uma interface com todos os métodos que o sistema suporta e que verifica que tanto o carro,o piloto e o campeonato escolhidos existem, adicionando um registo contendo esta informação.
- **Configurar corrida** - Neste cenário o único papel do Jogador é efetuar as afinações, escolher valor da downforce, escolher pneus e o modo do motor. Para isso temos uma interação com o Jogador e com a interface do Campeonato, e a própria trata de efetuar as afinações.
- **Apresentar Resultado**
    - **Resultado Final -** Neste cenário o papel do Utilizador é fazer o pedido do resultado final de um campeonato passando-lhe como atributo o código do campeonato. Este por sua vez irá buscar a pontuação das corridas e somá-las;
    - **Resultado Jogador** - Neste cenário o papel do Utilizador é fazer o pedido do resultado atual de um jogador passando-lhe como atributo o código do jogador. Este por sua vez irá buscar a pontuação das corridas e somá-las;
    -
    - **Adicionar Pontos -** Neste cenário o papel do Utilizador é fazer o pedido para que os pontos ganhos num dado campeonato sejam adicionados à sua pontuação global. Para isto o Utilizador passa o seu código de utilizador assim como o código do campeonato em questão, a partir daí o sistema vai buscar a pontuação do utilizador no dado campeonato e por fim soma estes aos pontos do Utilizador;
    - **Ranking -** Neste cenário o Utilizador pede ao sistema que este apresenta o Ranking global de todos os utilizadores. O Sistema vai consultar a posição de todos os Utilizadores e a partir daí organizá-los por pontuação.

Análise crítica dos resultados obtidos

O grupo considera que todos os objetivos apresentados para esta fase foram alcançados, formando uma boa base (junto com o trabalho desenvolvido na primeira fase) para os objetivos propostos na terceira e última fase, onde será implementado o sistema.

Em relação ao nosso diagrama de classes, acreditamos ter englobado todos os atributos e métodos para uma boa representação da modularidade do código e uma concisa relação entre todos os componentes.

O diagrama de sequências, baseado nas interações que ocorrem entre o utilizador e o sistema, descreve as mais relevantes trocas de informação e ações que o sistema permitirá realizar, quer entre componentes, quer entre um determinado componente e o utilizador.

De forma geral, e com base em todo o trabalho desenvolvido para esta fase, consideramos ter atingido todos os objetivos apresentados na íntegra.

Conclusão

Após a realização desta fase do trabalho consideramos que a mesma foi bastante importante para aprimorarmos os nossos conhecimentos relativamente à modelação concetual (mais concretamente diagramas de sequências, de classes e de componentes). Desta maneira, foi possível implementar os conceitos abordados nas aulas práticas e teóricas.

A maior dificuldade do grupo foi a construção dos diagramas de sequência, devido ao facto de serem estes os diagramas onde os elementos do grupo se sentiam menos preparados para a sua realização.

Apesar desta dificuldade, acreditamos ter atingido os resultados que eram esperados, construindo uma boa base para a última fase deste projeto.

Anexos

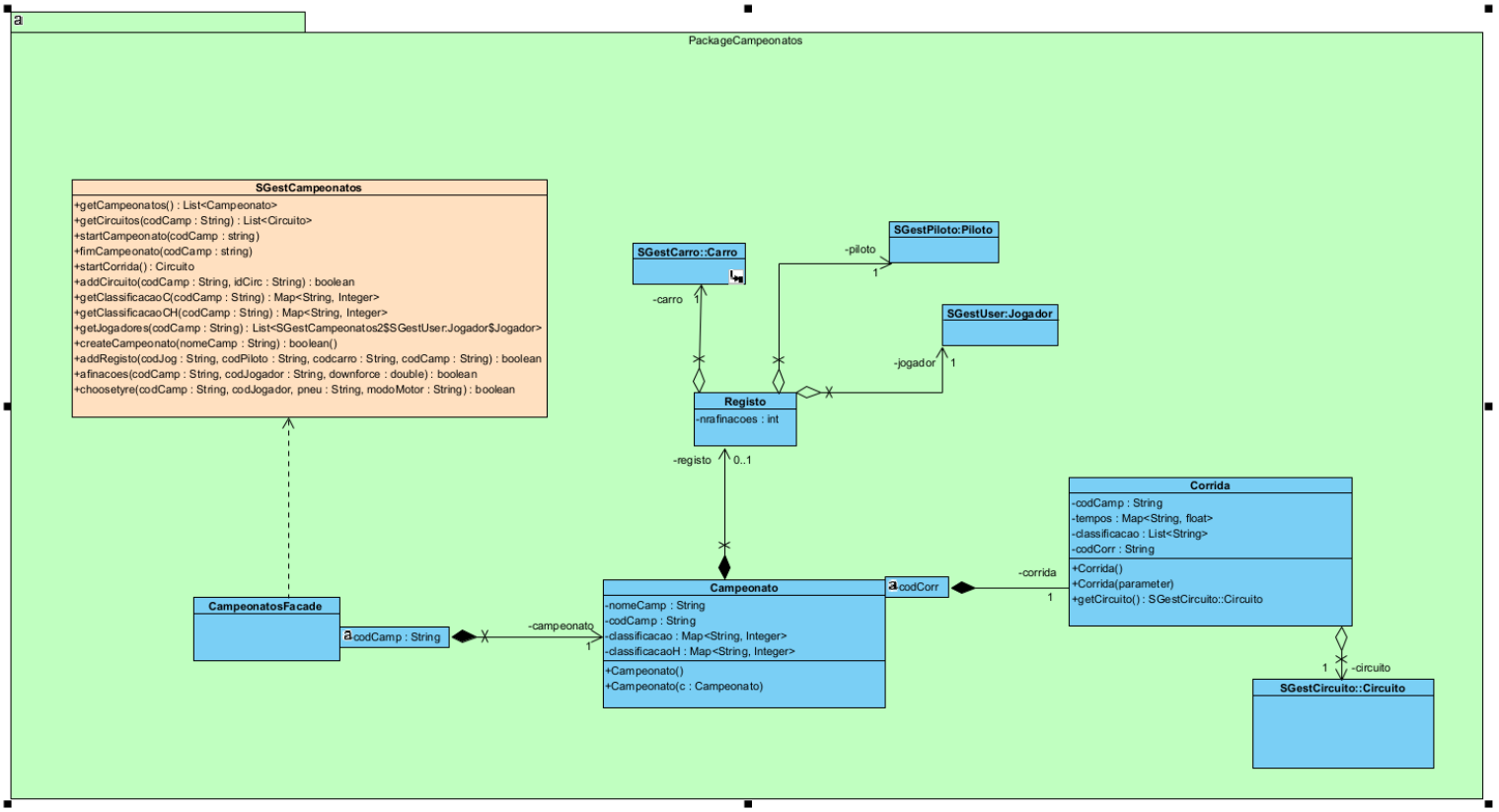


Figura 1-Subsistema de Gestão de Campeonatos

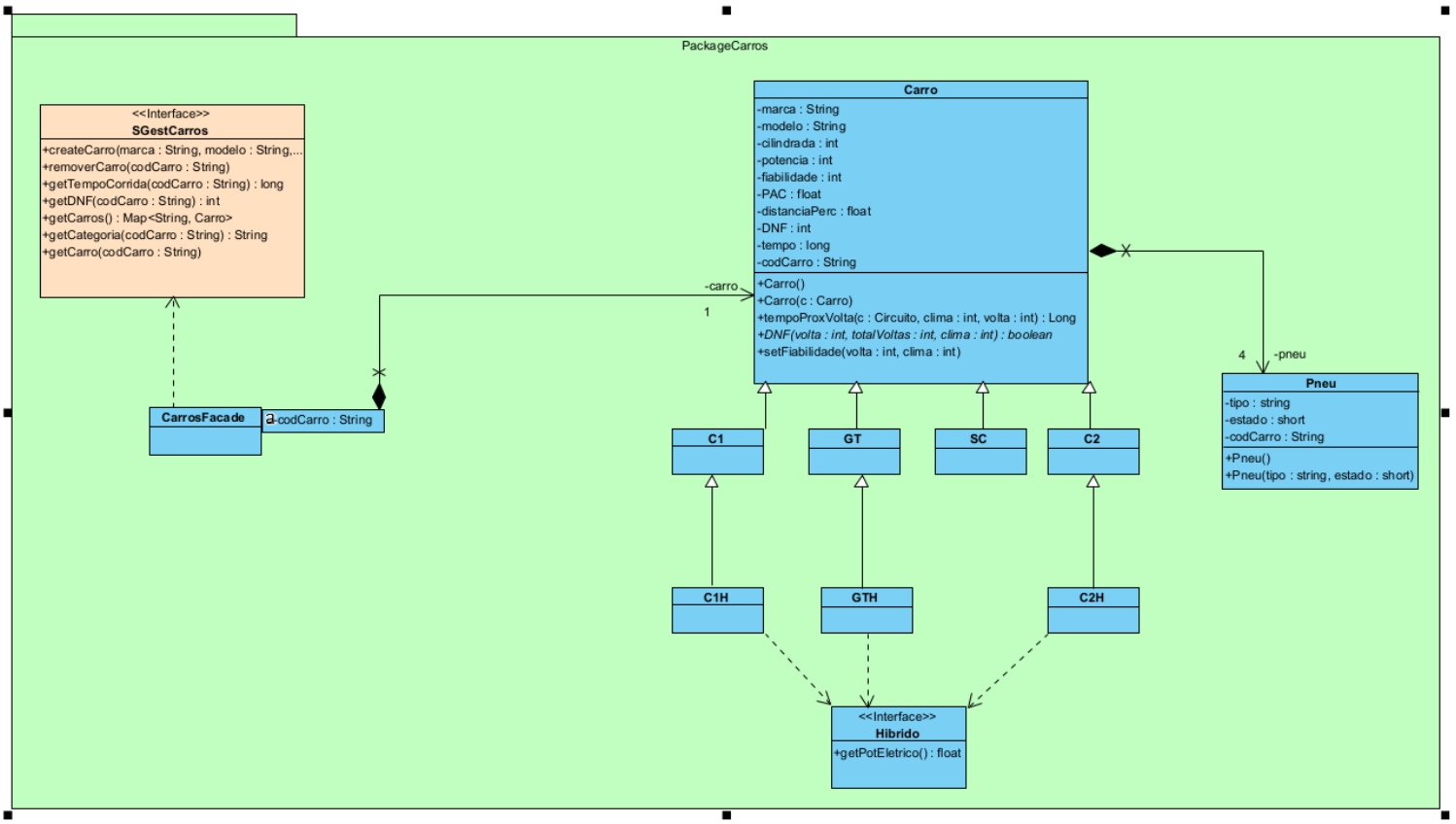


Figura 2 - Subsistema de Gestão de Carros

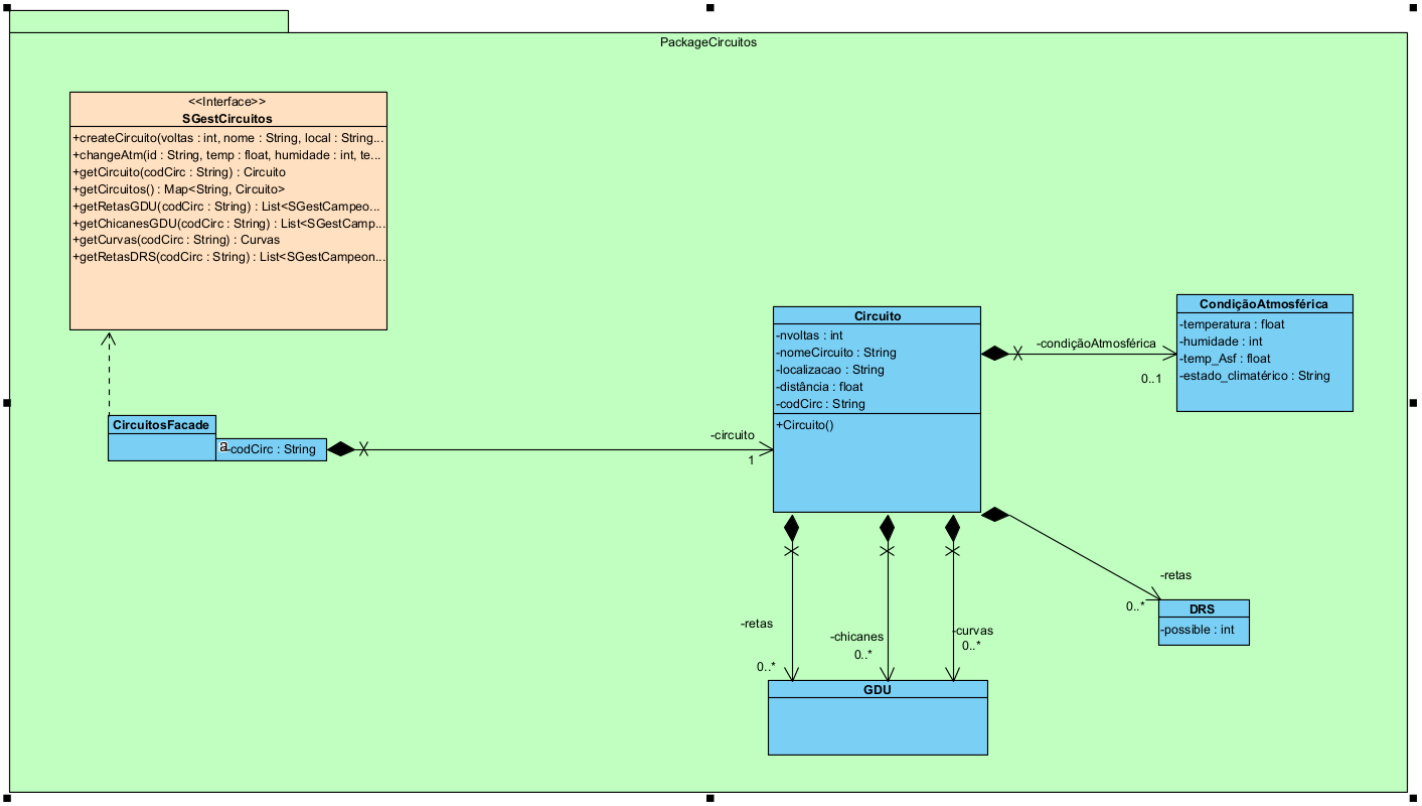


Figura 3- Subsistema de Gestão de Circuitos

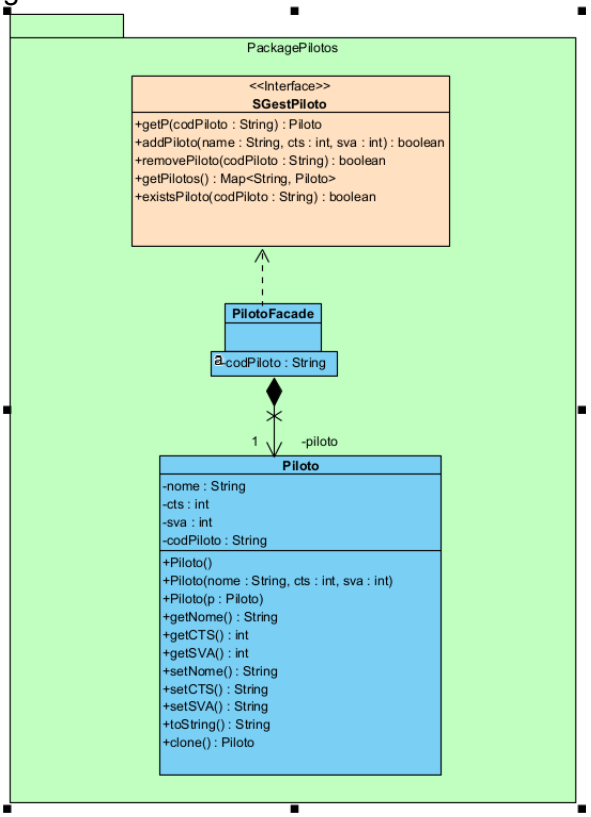


Figura 4 -Subsistema de Gestão de Pilotos

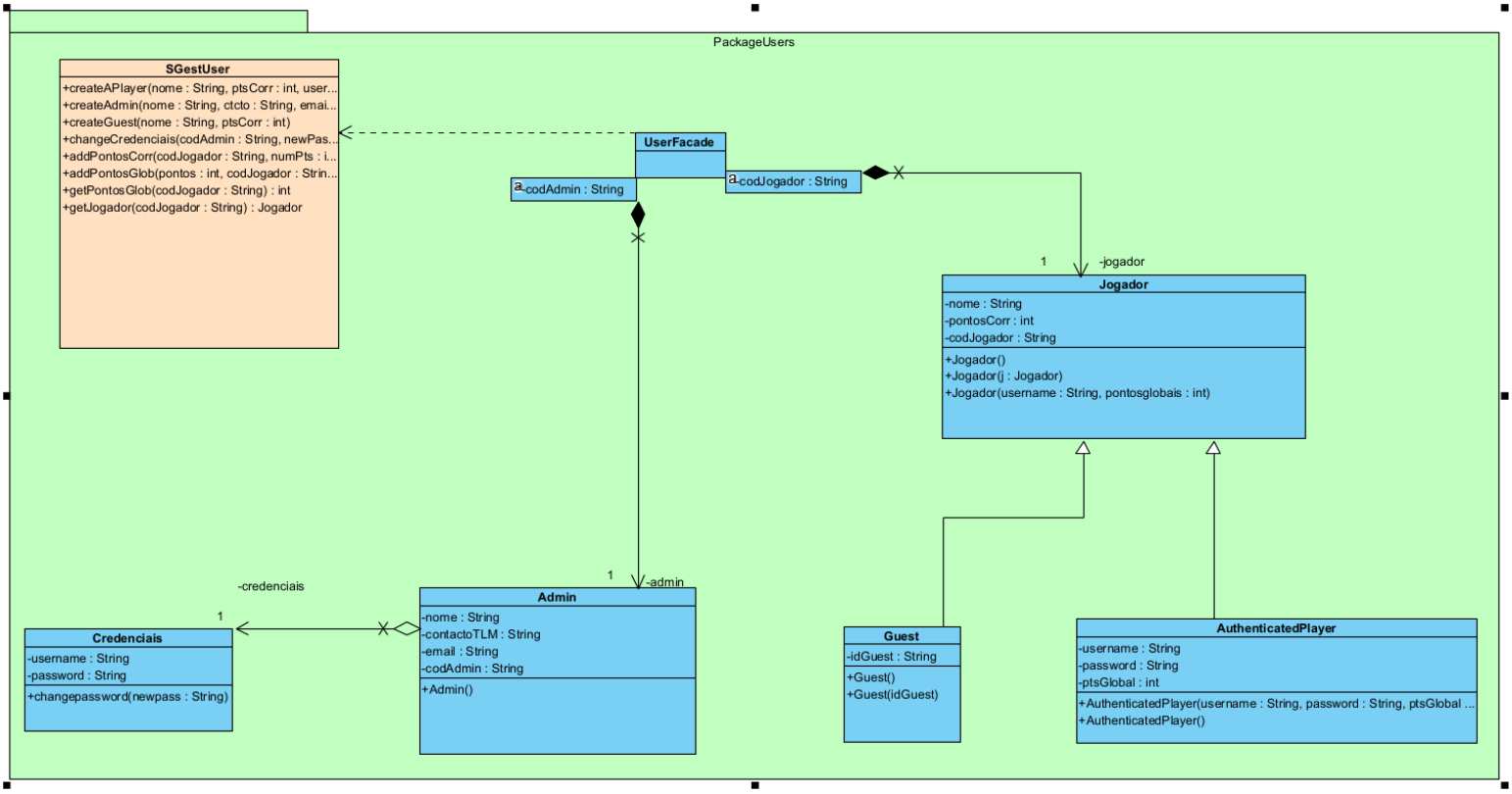


Figura 5-Subsistema de Gestão de utilizadores

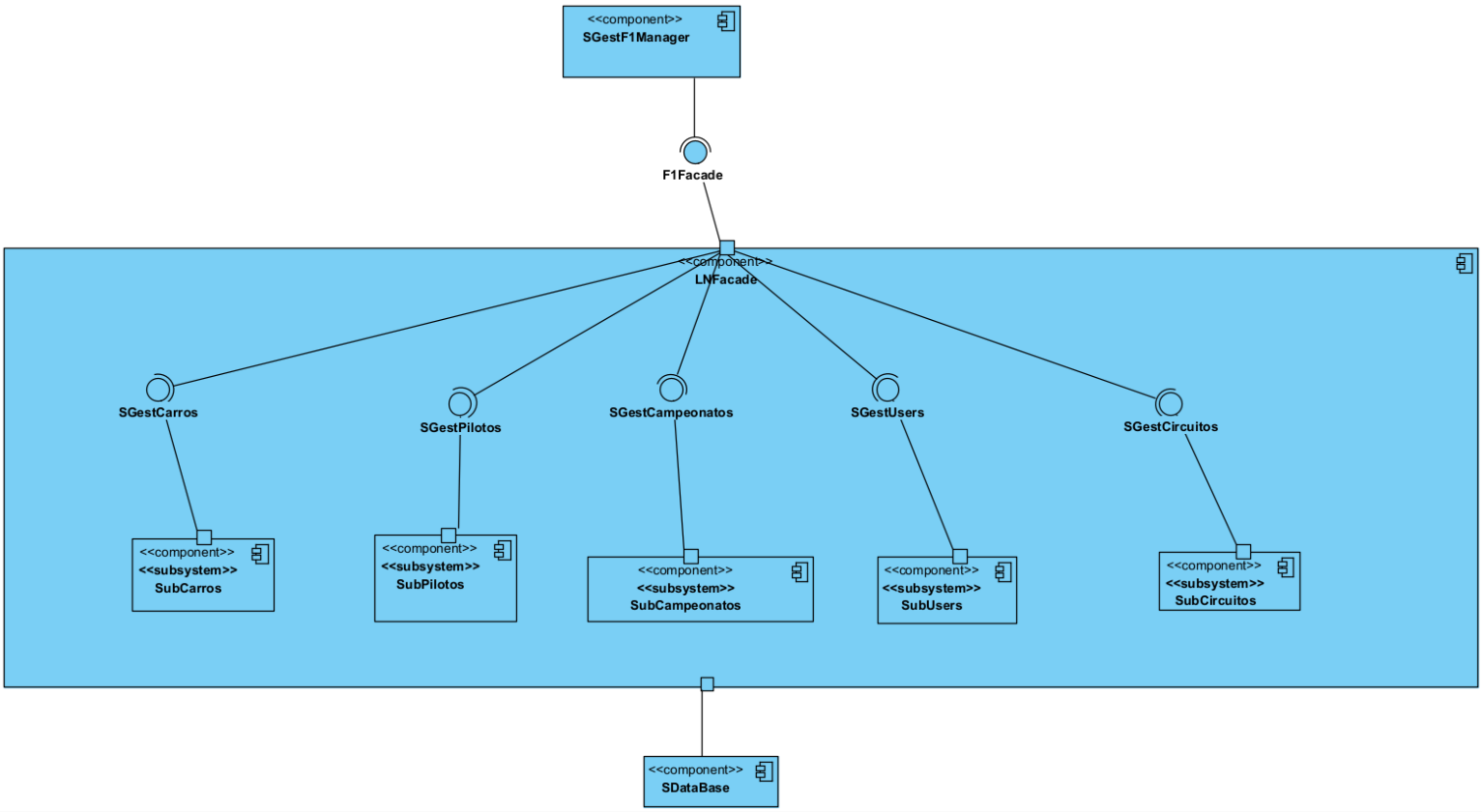


Figura 6-Diagrama de Componentes

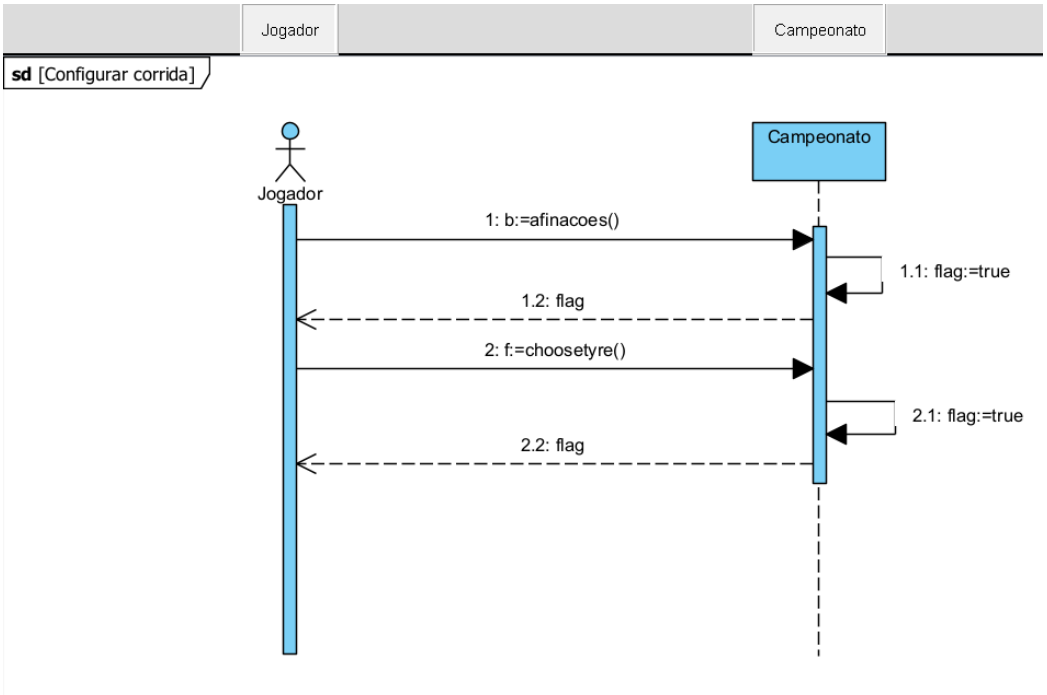


Figura 7 - Diagrama de Sequência (Configurar Corrida)

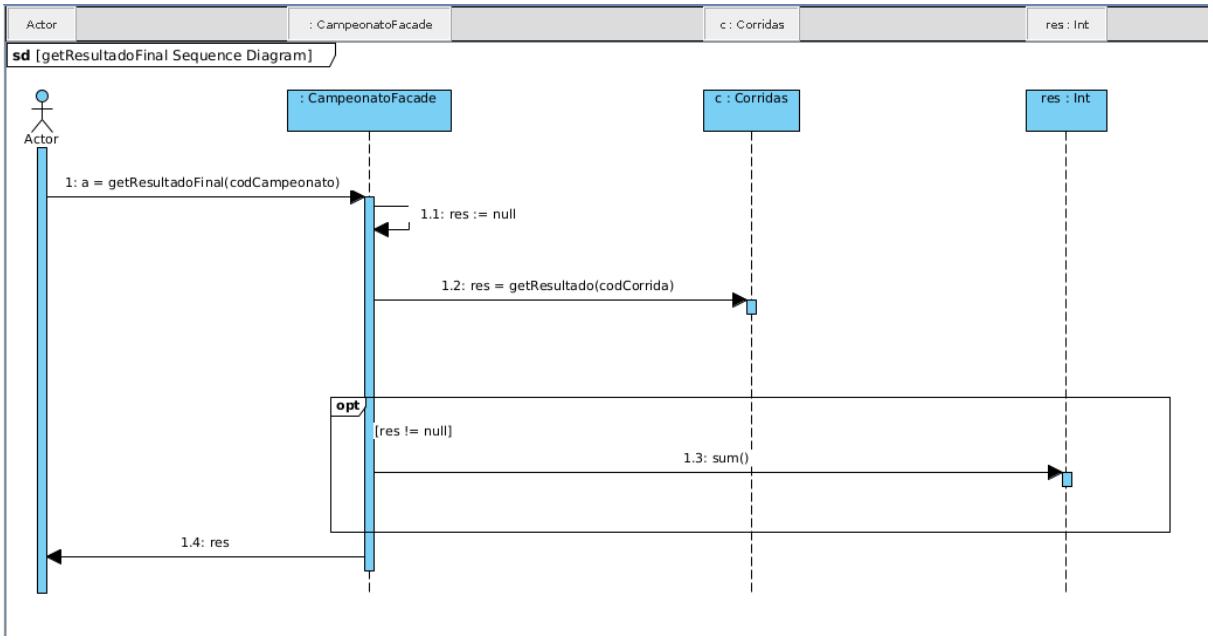


Figura 8- Diagrama de Sequência (Resultado Final)

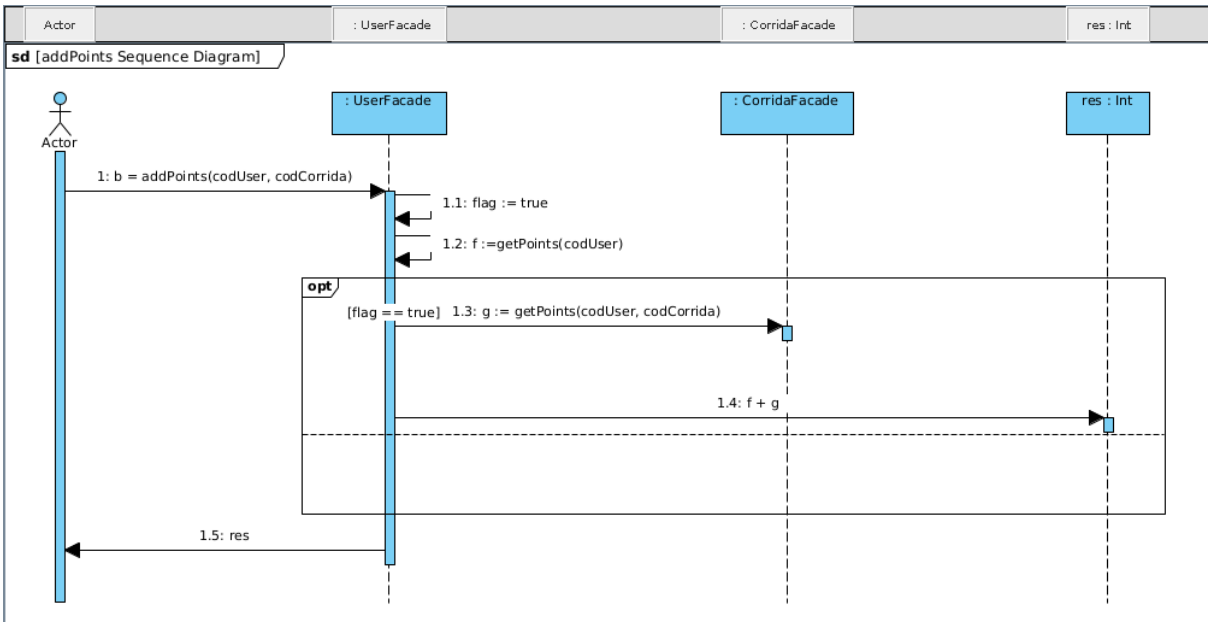


Figura 9- Diagrama de Sequência (Adiciona Pontos)

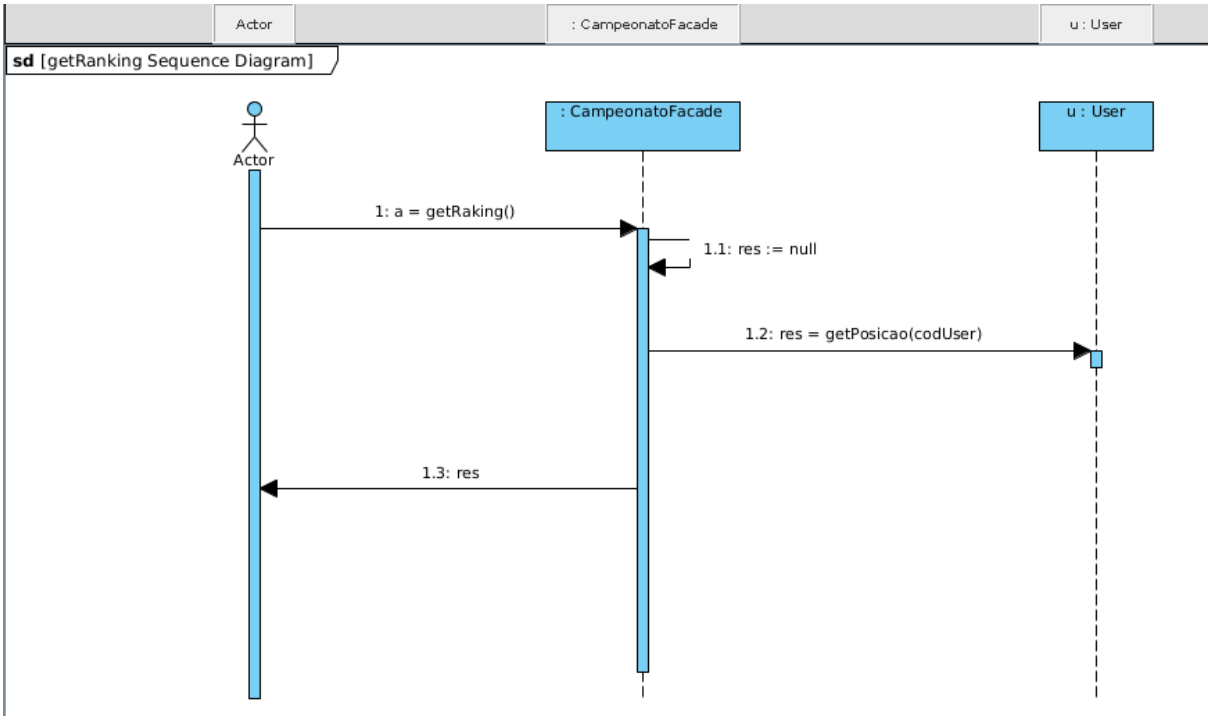


Figura 10 - Diagrama de Sequência (Ranking)

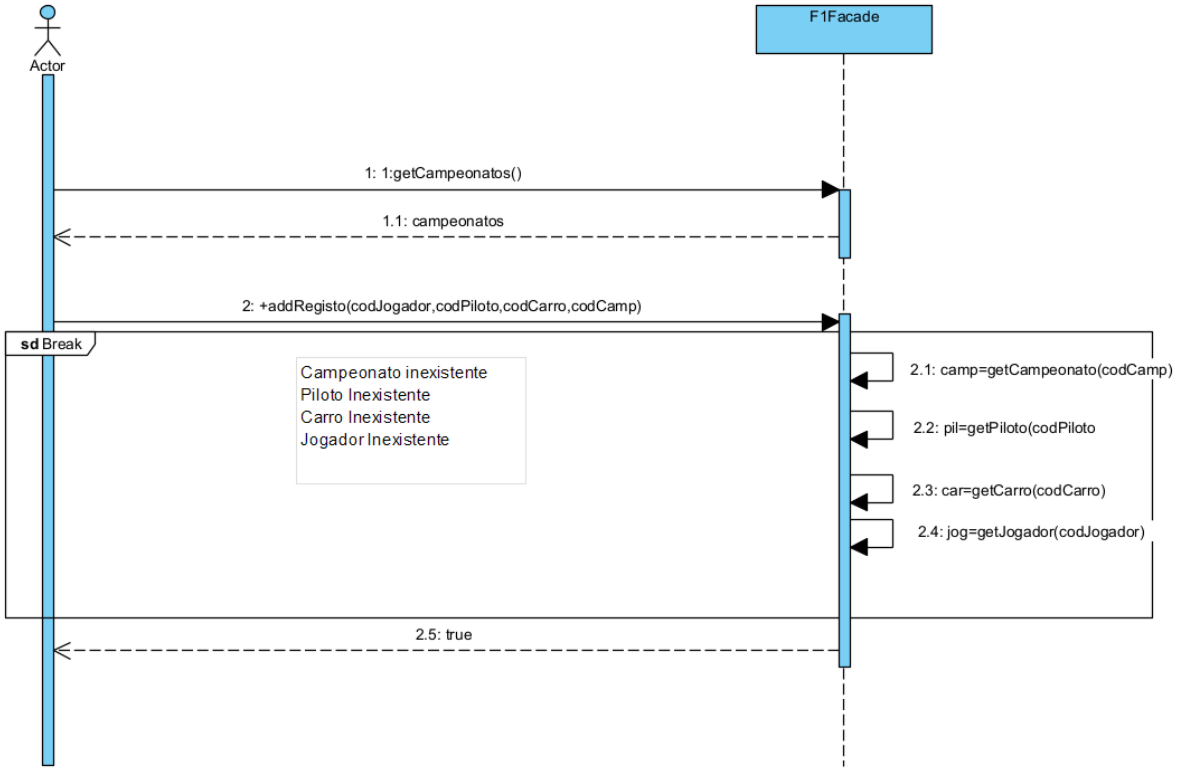


Figura 11 - Diagrama de Sequência (Adicionar Registo)

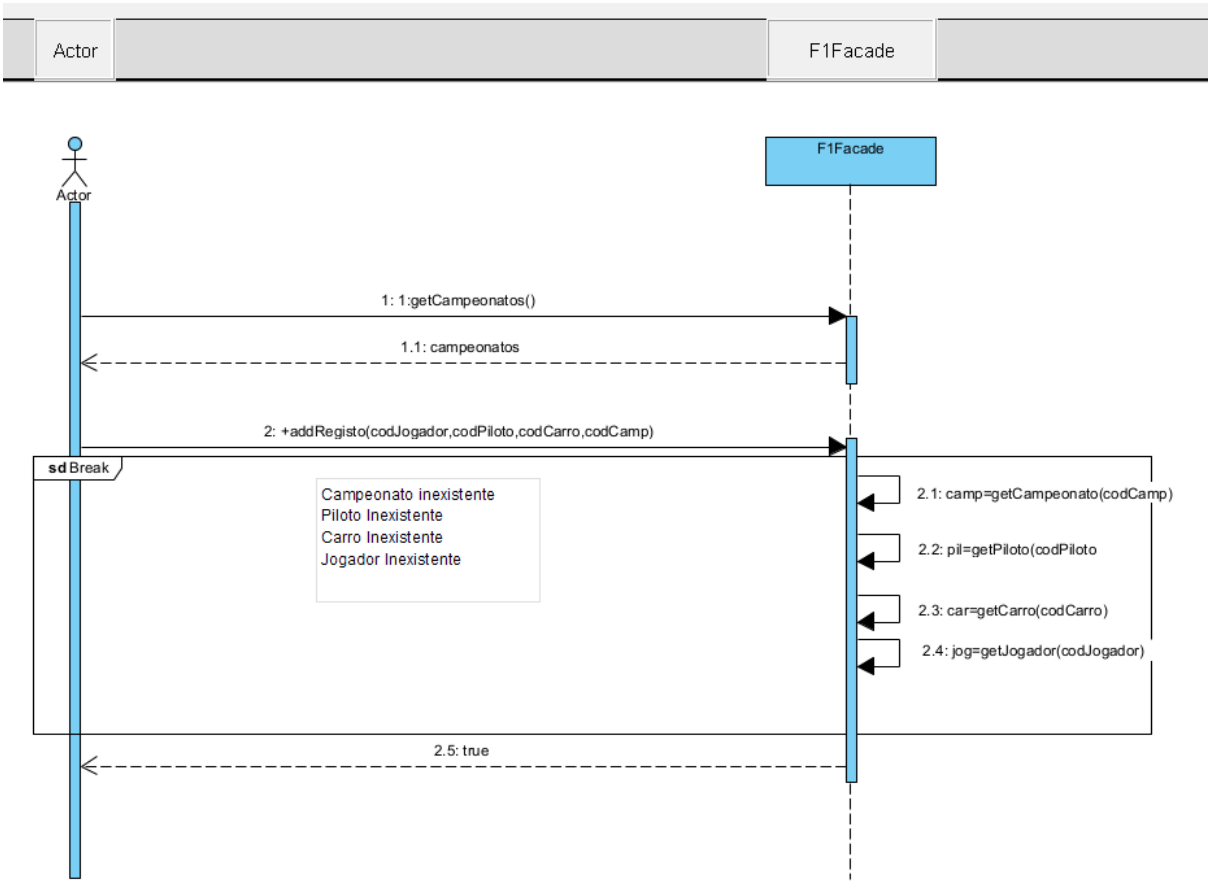


Figura 12 - Diagrama de Sequência (Adicionar Campeonato)