



**Universidade do Minho**  
Escola de Engenharia  
Mestrado Integrado em Engenharia Informática

## **Unidade Curricular de Laboratórios de Informática IV**

Ano Letivo de 2020/2021

### **WebGoal**

João Neves(A81366)  
Pedro Henriques(A84794)  
António Gonçalves(A85516)  
Guilherme Soares(A84527)

7 de junho de 2021

**LI4**

Data de Receção	
Responsável	
Avaliação	
Observações	

# WebGoal

João Neves(A81366)  
Pedro Henriques(A84794)  
António Gonçalves(A85516)  
Guilherme Soares(A84527)

7 de junho de 2021

# Resumo

Iremos ao longo deste relatório, referente ao projeto proposto da unidade curricular de Laboratórios de Informática IV, apresentar a nossa ideia de um Sistema de *Software* para uma aplicação de monitorização de eventos desportivos, mais concretamente na área do futebol, bem como fundamentar e estruturar toda a base necessária para que esta seja desenvolvida corretamente.

Para tal, iremos analisar os requisitos do projeto que terão de ser realizados para as fases seguintes e também realizar o planeamento necessário para todo o trabalho futuro para que o sistema esteja completo.

**Área de Trabalho:** Engenharia de *Software*, Desporto

**Palavras-Chave:** Sistema de *Software*, Diagrama de *Gantt*, *Waterfall Model*, Diagrama de Sequência de Subsistemas, Use Case, Diagrama de Classe, Diagrama de Pacotes, Diagrama ORM, Base de Dados

# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização . . . . .	2
1.2	Apresentação do Caso de Estudo . . . . .	2
1.3	Motivação e Objectivos . . . . .	3
1.4	Estrutura do Relatório . . . . .	4
<b>2</b>	<b>Definição do sistema e seus componentes</b>	<b>5</b>
2.1	Funcionalidades do Sistema . . . . .	5
2.2	Maquete do Sistema . . . . .	6
2.3	Recursos necessários . . . . .	7
2.4	Medidas de Sucesso . . . . .	7
2.5	Plano de Desenvolvimento . . . . .	8
<b>3</b>	<b>Fase de Especificação</b>	<b>11</b>
3.1	Análise de Requisitos . . . . .	12
3.1.1	Requisitos Funcionais . . . . .	12
3.1.2	Requisitos não Funcionais . . . . .	14
3.2	Especificação do Sistema . . . . .	15
3.2.1	Modelo do Domínio . . . . .	15
3.2.2	Diagrama de Use Cases . . . . .	16
3.2.3	Especificação dos Use Cases . . . . .	18
3.2.4	Arquitectura da Solução . . . . .	27
3.2.5	Diagrama de Sequência de Subsistemas . . . . .	28
3.3	Arquitetura da Camada de negócios . . . . .	36
3.3.1	Camada de Negócios . . . . .	36
3.3.2	Dicionário das principais Classes . . . . .	36
3.3.3	Descrição da arquitectura . . . . .	37
3.3.4	Diagrama de Classes . . . . .	37
3.3.5	Diagrama de ORM . . . . .	38
3.4	Diagrama de Pacotes . . . . .	39
3.5	Projeto da base de dados . . . . .	40
3.6	Protótipo da interface . . . . .	41
<b>4</b>	<b>Fase de Construção</b>	<b>49</b>
4.1	Base de Dados . . . . .	50
4.1.1	Alojamento da Base de Dados . . . . .	50
4.1.2	Atualização da Informação da Base de Dados . . . . .	51

4.2	Interface . . . . .	52
4.3	Funcionalidades Finalizadas/Pretendidas . . . . .	53
<b>5</b>	<b>Conclusão</b>	<b>56</b>

# Lista de Figuras

2.1	Maquete Inicial do Sistema. . . . .	6
2.2	Planeamento e distribuição das tarefas da fase de Fundamentção. . . . .	9
2.3	Planeamento e distribuição das tarefas da fase de Especificação. . . . .	9
2.4	Planeamento e distribuição das tarefas da fase de Construção. . . . .	9
2.5	Diagrama de Gantt da fase de Fundamentção. . . . .	10
2.6	Diagrama de Gantt da fase de Especificação. . . . .	10
2.7	Diagrama de Gantt da fase de Construção. . . . .	10
3.1	Modelo de Domínio. . . . .	15
3.2	Diagrama de Use Case . . . . .	17
3.3	Consultar jogos de uma Liga por Data . . . . .	18
3.4	Consultar jogo de uma liga por Status . . . . .	19
3.5	Consultar jogos de um Clube com determinado Status . . . . .	20
3.6	Consultar jogos de uma Equipa numa determinada Data . . . . .	20
3.7	Consultar plantel de uma Equipa . . . . .	21
3.8	Consultar classificações de uma liga . . . . .	21
3.9	Consultar melhores marcadores de uma liga . . . . .	22
3.10	Consultar eventos no mapa . . . . .	22
3.11	Registar na aplicação . . . . .	23
3.12	Login na aplicação . . . . .	24
3.13	Consultar favoritos . . . . .	24
3.14	Adicionar conteúdo à lista de favoritos . . . . .	25
3.15	Remover conteúdo dos Favoritos . . . . .	25
3.16	Use case editar configurações pessoais . . . . .	26
3.17	Use case terminar sessão. . . . .	26
3.18	Arquitetura do Sistema . . . . .	27
3.19	DSS Consultar jogos de uma liga por dia . . . . .	28
3.20	DSS Consultar jogos de uma liga por Status . . . . .	28
3.21	DSS Consultar jogos de uma Equipa por data . . . . .	29
3.22	DSS Consultar jogos de uma Equipa por Status . . . . .	29
3.23	DSS Consultar Plantel de uma Equipa . . . . .	30
3.24	DSS Consultar Classificações de uma liga . . . . .	30
3.25	DSS Consultar melhores marcadores de uma liga . . . . .	31
3.26	DSS Consultar eventos no mapa . . . . .	31
3.27	DSS Registar na aplicação . . . . .	32
3.28	DSS Consultar Favoritos . . . . .	33

3.29	DSS Login na aplicação . . . . .	33
3.30	DSS Adicionar conteúdo à lista de favoritos . . . . .	34
3.31	DSS Remover conteúdo dos Favoritos . . . . .	34
3.32	DSS Editar Configurações pessoais . . . . .	35
3.33	DSS Terminar sessão . . . . .	35
3.34	Diagrama de Classes . . . . .	37
3.35	Diagrama de Classes com DAO . . . . .	38
3.36	Diagrama de Package . . . . .	39
3.37	Modelo Lógico . . . . .	40
3.38	Interface de Login . . . . .	41
3.39	Interface de registo de utilizador . . . . .	41
3.40	Interface de terminar sessão . . . . .	42
3.41	Interface da consulta de jogos por data . . . . .	42
3.42	Interface da consulta de jogos por nome de equipa . . . . .	43
3.43	Interface do mapa com os jogos marcados . . . . .	43
3.44	Interface da consulta de jogos agendados . . . . .	44
3.45	Interface da consulta de jogos a decorrer . . . . .	44
3.46	Interface da consulta da informação das equipas . . . . .	45
3.47	Interface da consulta das classificações de ligas . . . . .	45
3.48	Interface de alterar a lista de favoritos . . . . .	46
3.49	Interface de alterar os dados pessoais . . . . .	46
3.50	Interface de consulta de jogos das equipas favoritas . . . . .	47
3.51	Interface da consulta dos melhores marcadores de uma liga . . . . .	47
3.52	Interface para aceitar o envio de notificações . . . . .	48
4.1	Interface da Página Inicial . . . . .	53
4.2	Interface principal e Consulta de jogos por Data/Liga . . . . .	54
4.3	Interface de consulta de jogos por Status e Liga . . . . .	55
4.4	Interface de Consulta da Classificação das Ligas . . . . .	55
4.5	Interface de Consulta dos melhores marcadores das Ligas . . . . .	55

# 1 Introdução

Inicialmente iremos contextualizar o tema, escolhido pelo grupo, para a realização do projeto, de forma a facilitar a percepção dos restantes assuntos que iremos abordar. Outro dos assuntos que iremos expor será a descrição dos motivos que nos motivaram a optar por um Sistema de Monitorização de eventos futebolísticos, nomeadamente jogos de futebol, assim como os objetivos pretendidos com a escolha deste tema e, também, resumir um pouco a estrutura deste relatório.

O projeto vai seguir o modelo em cascata ( *Waterfall Model*), e segue as seguintes fases:

1. Levantamento e análise dos requisitos.
2. Modulação do sistema.
3. Implementação do sistema.
4. Verificação do sistema.
5. Instalação do sistema.



## 1.1 Contextualização

O presente projeto faz parte integrante da unidade curricular de Laboratórios de Informática IV, tendo como objetivo principal, fundamentar, projetar e gerir o desenvolvimento de um Sistema de *software* tanto quanto analisar e especificar da forma mais completa possível, todos os requisitos funcionais, não funcionais e requisitos de negócio. Para além disso, o presente projeto pretende também transmitir um contacto mais “profissional” e real com o mundo do desenvolvimento de *software*.

Com isto, a equipa docente da unidade curricular em questão decidiu que o presente projeto seria um sistema de monitorização de eventos, sendo que cabe a nós, grupo de trabalho, decidir que tipo de sistema iremos implementar.

Após algum debate, o grupo chegou à conclusão que seria do agrado de todos desenvolver um sistema que monitorize resultados, assim como momentos importantes de eventos futebolísticos.

## 1.2 Apresentação do Caso de Estudo

O nome que escolhemos para o nosso projeto foi *WebGoal* e, como foi dito anteriormente, tem como objetivo monitorizar eventos futebolísticos e disponibilizar estatísticas detalhadas ao utilizador em tempo real.

O utilizador poderá registar-se na aplicação, ter uma conta pessoal para a qual é preciso autenticar-se para a aceder, embora não seja necessário ter uma conta para poder aceder à informação disponibilizada. Qualquer utilizador tem acesso à informação presente no projeto em tempo real, mas só utilizadores registados podem ter favoritos e receber notificações relativas aos mesmos. Para além disto, a aplicação irá permitir observar eventos no mapa, bem como consultar informação referente às ligas presentes, e às equipas dos jogos.

Qualquer utilizador pode facilmente encontrar o evento que procura, é só procurar com algumas palavras chave e chega ao destino pretendido. Depois de procurar o evento, é só carregar nele e terá disponível toda a informação desejada e atualizada.

## 1.3 Motivação e Objectivos

Hoje em dia, assistir a um jogo de futebol é cada vez mais uma regalia, visto que está cada vez mais privatizado na medida em que é necessário subscrições ou bilhetes caros para poder assistir a um jogo específico. Assim, a aplicação a desenvolver consistirá numa forma gratuita, simples e intuitiva de uma pessoa poder seguir o seu clube através de uma página/app que lhe fornecerá todos os dados necessários para que possa estar a par de tudo que acontece durante o(s) jogo(s).

Além disso, consideramos também que o país onde estamos é um bom recetor de um projeto como este, devido ao enorme interesse em desporto (mais especificamente, o futebol) por parte da população.

Também procuramos inovar em relação aos outros projetos com o mesmo objetivo, o utilizador poderá, por exemplo, ver os eventos que estão a decorrer no momento num mapa global, podendo aceder ao evento desejado num clique.

E, para reforçar o que foi dito anteriormente, o nosso objetivo é proporcionar uma alternativa aos serviços pagos e oferecer um serviço gratuito que forneça o máximo de informação possível ao utilizador.

## 1.4 Estrutura do Relatório

Após apresentada a contextualização em que surge o projeto, assim como o caso de estudo em que se insere e motivações associadas ao desenvolvimento do mesmo, as secções seguintes explicam de uma forma mais detalhada como o nosso grupo irá distribuir e gerir o projeto. Serão também especificados todos os passos desde a modelação e especificação até ao desenvolvimento em si e os testes do sistema final.

O presente relatório será dividido em 3 partes: a fundamentação, a especificação e a construção.

Relativamente à Fundamentação, a parte inicial, iremos definir o sistema a desenvolver bem como os seus componentes. Serão apresentados os principais objetivos e funcionalidades do sistema, evidenciando e suportando a veracidade e sustentabilidade do mesmo e, também, os recursos necessários à realização do mesmo. Iremos, também, apresentar uma maquete do sistema e apresentar justificações em termos do modelo de negócio. Por fim iremos definir as medidas de sucesso e o plano de desenvolvimento para o projeto.

Posteriormente, partiremos para a especificação, onde será efetuada uma análise dos requisitos do sistema. Posteriormente, e de uma forma ilustrativa, serão elaborados vários diagramas, com o intuito de fazer uma especificação geral e detalhada do *software* a desenvolver e de todas as funcionalidades nele presentes. Com isto, serão efetuados Use cases e diagramas de sequência com o objetivo de responder as funcionalidades do sistema, assim como um modelo do domínio que representara todo o plano de desenvolvimento.

Consequentemente chegaremos à parte de construção, onde será explicada a implementação final realizada, sendo justificadas todas as ferramentas utilizadas no desenvolvimento do projeto em questão. Por fim será apresentada toda a arquitetura por detrás do projeto.

Por ultimo, apresentamos uma conclusão, que tem como objetivo sumariar aquilo que conseguimos fazer com aproveitamento, assim como aquilo que poderíamos ter melhorado.

## 2 Definição do sistema e seus componentes

Tal como já foi explicado anteriormente, a principal utilidade desta aplicação será a possibilidade de seguir, em tempo real, os resultados de vários eventos futebolísticos.

### 2.1 Funcionalidades do Sistema

Primeiro, o sistema irá permitir a qualquer pessoa aceder à informação presente na sua base de dados relativa a vários eventos futebolísticos. Para além disto, será possível efetuar o registo do utilizador. Para isto o mesmo terá que fornecer alguma informação, nomeadamente o nome de utilizador, uma *password*, e o seu email.

Apesar de a informação ser visível mesmo para quem não se encontra registado, os utilizadores com conta poderão definir Clubes favoritos, recebendo notificação sobre atualizações referentes aos mesmos.

Iremos então analisar mais em detalhe as possíveis interações dos utilizadores com o sistema em si, a pesquisa de eventos e a análise de eventos.

Para procurar algum jogo, o utilizador poderá inserir algumas palavras chave, como por exemplo o nome do clube que quer que tenha participado no evento ou uma data na qual tenha acontecido algum evento. Com isto o manuseamento da informação torna-se bastante mais simples, permitindo ao utilizador ver apenas aquilo que está realmente à procura.

Quanto à análise dos eventos, após selecionar um evento específico, o utilizador poderá consultar a informação referente ao mesmo, por exemplo, o resultado, a duração e outras estatísticas recolhidas durante o evento.

## 2.2 Maquete do Sistema

Como referido anteriormente, existem dois tipos de utilizadores neste sistema: utilizadores normais, que podem aceder a toda a informação, e utilizadores com conta, que possuem as suas preferências guardadas. Para além disto, o utilizador registado pode alterar estas preferências. Assim obtemos a seguinte maquete:

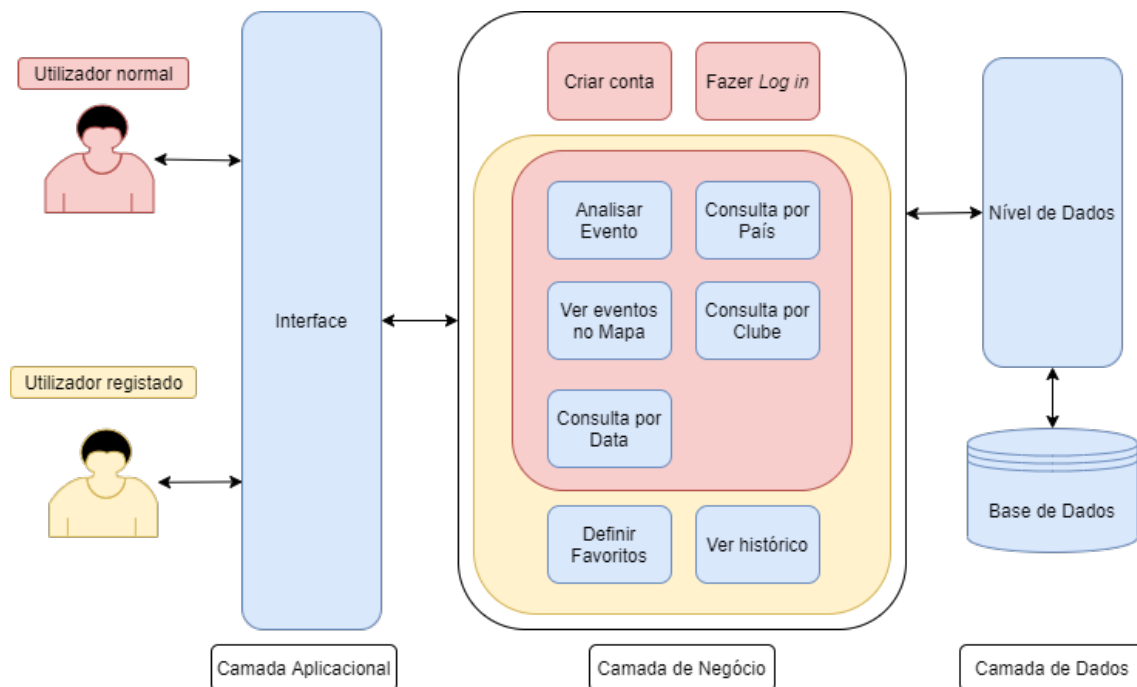


Figura 2.1: Maquete Inicial do Sistema.

Ambos os tipos de utilizadores poderão efetuar várias operações utilizando a interface. Com isto poderão inserir informação para encontrar eventos específicos ou analisar aqueles que desejam. Qualquer utilizador normal poderá criar uma conta e fazer *log in* na mesma.

Dentro da camada de Negócio temos as várias operações que os utilizadores poderão realizar. Estas operações irão recolher a informação da Camada de Dados de forma a conseguir responder aos pedidos, sendo depois a resposta enviada novamente aos utilizadores.

Esta versão é uma primeira abordagem criada. Com o desenvolver do sistema foram adicionadas várias funcionalidades e alteradas algumas, sendo estas devidamente referenciadas em partes mais avançadas do relatório.

## 2.3 Recursos necessários

Tendo em conta todas as funcionalidades do sistema descritas até agora, será então necessário analisar os recursos fundamentais para que este seja implementado corretamente. Sendo assim temos então os seguintes recursos:

- Pessoas para pesquisarem sobre o tema;
- Pessoas para encontrar a informação a usar no sistema;
- Pessoas para delinear os objetivos;
- Pessoas para desenhar os Diagramas de *Use Cases* e de Sequência;
- Pessoas para desenvolver a Base de Dados a utilizar;
- Pessoas para desenvolver o código;
- Pessoas para documentar e escrever o relatório do desenvolvimento;
- Pessoas para testar o sistema;

## 2.4 Medidas de Sucesso

Este projecto será, para o grupo de trabalho, um caso de sucesso, se, numa fase final do trabalho, o sistema de *software* por nós implementado cumprir todos os objetivos e requisitos propostos na fase inicial e, acima de tudo, ser aceite e agradar ao cliente, que é o ponto fulcral do projeto. Outros aspetos por nós considerados fundamentais ao sucesso serão, por exemplo, a realização do projeto dentro dos limites de tempo impostos inicialmente e, também, seguir, sem muitos desvios, o plano de desenvolvimento delineado pelo grupo. No caso destes parâmetros se verificarem, poderíamos dizer que houve uma boa envolvência da equipa de trabalho, resultando num projeto bem controlado e organizado, algo essencial para a confiança no produto final.

Outros pontos que podem ajudar ao sucesso serão uma utilização ponderada dos recursos e a coerência do projeto ao longo do desenvolvimento do mesmo.

Tendo em conta as metas acima propostas, todo o projeto será realizado com o objetivo de cumprir as mesmas, atingindo assim, o tão esperado sucesso final do projeto.

## 2.5 Plano de Desenvolvimento

O desenvolvimento deste projecto será dividido em três partes, igualmente importantes para o sucesso do mesmo. Numa primeira fase teremos a fundamentação, posteriormente a fase de especificação, e por fim a construção do *software* desenvolvido.

A primeira fase, a fundamentação, consiste na definição do projeto e de todas as características importantes para a realização do mesmo. Esta definição será o pilar para o desenvolvimento de todo o restante projeto.

A segunda fase, a especificação, corresponde à modulação do sistema e ao desenvolvimento de uma base de dados, bem estruturada, que irá ser utilizada pelo produto final. Nesta fase iremos desenvolver o modelo funcional do *software*.

A terceira e última fase, correspondente à construção. Irá ser a fase de escrita do código do *software*. Este irá servir-se da modulação e da base de dados da fase de especificação. Iremos também realizar a interface da aplicação, onde irá ser possível a interação com o utilizador.

As etapas todas do projeto serão realizadas sequencialmente, visto que todas as etapas dependem da posterior etapa para o bom funcionamento da aplicação final. Desta forma, todas as fases devem ser realizadas da maneira mais pondera e detalhada possível, sendo que, para tal, é essencial um bom planeamento do projeto. Este planeamento será uma poderosa ferramenta para uma boa resolução do projeto, com o mínimo de erros possíveis e sem qualquer atraso na realização do mesmo.

Assim sendo, recorrendo à ferramenta *Microsoft Project*, construiu-se um diagrama de *Gantt* (*figura 2.5, 2.6 e 2.7*) tendo por base o plano de desenvolvimento do projeto (*figura 2.2, 2.3 e 2.4*), delineado pela equipa, para todas as fases do trabalho.

		<b>Fundamentação</b>	<b>19 dias</b>	<b>Qui 04/03/21</b>	<b>Seg 22/03/21</b>	
		Identificação e caracterização geral da aplicação a desenvolver	7 dias	Qui 04/03/21	Qua 10/03/21	Pedro;João;António;Guilherme
		Fundamentação do seu desenvolvimento e justificação em termos de modelo de negócio	6 dias	Qui 11/03/21	Ter 16/03/21	2 António;Pedro
		Medidas de sucesso	1 dia	Qua 17/03/21	Qua 17/03/21	3 Guilherme;João
		Plano de desenvolvimento	3 dias	Qui 18/03/21	Sáb 20/03/21	4 Guilherme;João
		Revisão e Conclusões	1 dia	Dom 21/03/21	Dom 21/03/21	5 António;Guilherme;João;Pedro
		Relatório	18 dias	Qui 04/03/21	Dom 21/03/21	António;Guilherme;João;Pedro
		Entrega	1 dia	Seg 22/03/21	Seg 22/03/21	Pedro

Figura 2.2: Planeamento e distribuição das tarefas da fase de Fundamentação.

		<b>Especificação</b>	<b>50 dias</b>	<b>Seg 22/03/21</b>	<b>Seg 10/05/21</b>	
		Levantamento e análise de requisitos	8 dias	Seg 22/03/21	Seg 29/03/21	António;Guilherme;João;Pedro
		Diagramas e Especificação do Use Case	8 dias	Ter 30/03/21	Ter 06/04/21	10 Guilherme;Pedro
		Diagramas de Sequência	6 dias	Qui 01/04/21	Ter 06/04/21	António;João
		Diagramas de Atividade	5 dias	Qua 07/04/21	Dom 11/04/21	12 João;Pedro
		Modelação da base de dados	19 dias	Seg 12/04/21	Sex 30/04/21	13 António;Guilherme;João;Pedro
		Prototipação	5 dias	Sáb 01/05/21	Qua 05/05/21	António;Guilherme
		Revisão e Conclusões	3 dias	Sex 07/05/21	Dom 09/05/21	António;Guilherme;João;Pedro
		Relatório	49 dias	Seg 22/03/21	Dom 09/05/21	António;Guilherme;João;Pedro
		Entrega	1 dia	Seg 10/05/21	Seg 10/05/21	Pedro

Figura 2.3: Planeamento e distribuição das tarefas da fase de Especificação.

		<b>Construção</b>	<b>29 dias</b>	<b>Seg 10/05/21</b>	<b>Seg 07/06/21</b>	
		Seleção das ferramentas a utilizar	1 dia	Seg 10/05/21	Seg 10/05/21	António;Guilherme;João;Pedro
		Implementação da base de dados	4 dias	Ter 11/05/21	Sex 14/05/21	20 João;Pedro
		Camada de Dados	17 dias	Sáb 15/05/21	Seg 31/05/21	21 Pedro;João
		Camada de Negócio	15 dias	Seg 17/05/21	Seg 31/05/21	António
		Camada de Apresentação	13 dias	Qua 19/05/21	Seg 31/05/21	Guilherme
		Testes	1 dia	Ter 01/06/21	Ter 01/06/21	24 António;Guilherme;João;Pedro
		Implementação do Sistema	3 dias	Qua 02/06/21	Sex 04/06/21	25 António;Guilherme;João;Pedro
		Revisão e Conclusões	1 dia	Sáb 05/06/21	Sáb 05/06/21	26 António;Guilherme;João;Pedro
		Instalação do Sistema	1 dia	Dom 06/06/21	Dom 06/06/21	27 António;Guilherme;João;Pedro
		Relatório	27 dias	Ter 11/05/21	Dom 06/06/21	António;Guilherme;João;Pedro
		Entrega	1 dia	Seg 07/06/21	Seg 07/06/21	Pedro

Figura 2.4: Planeamento e distribuição das tarefas da fase de Construção.



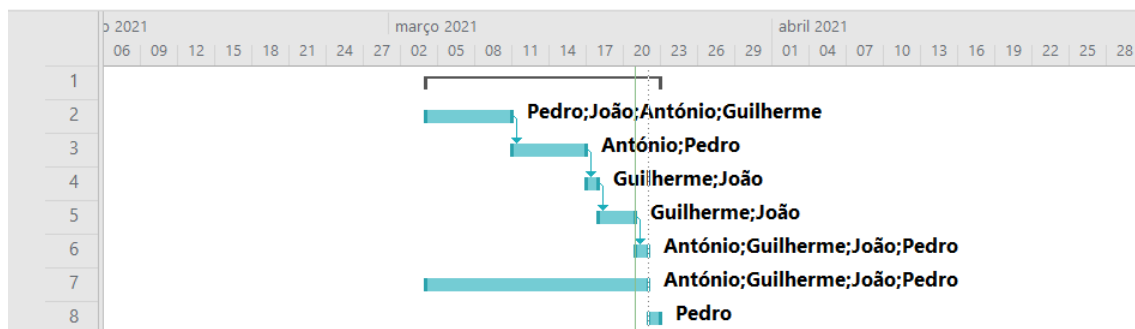


Figura 2.5: Diagrama de Gantt da fase de Fundamentação.

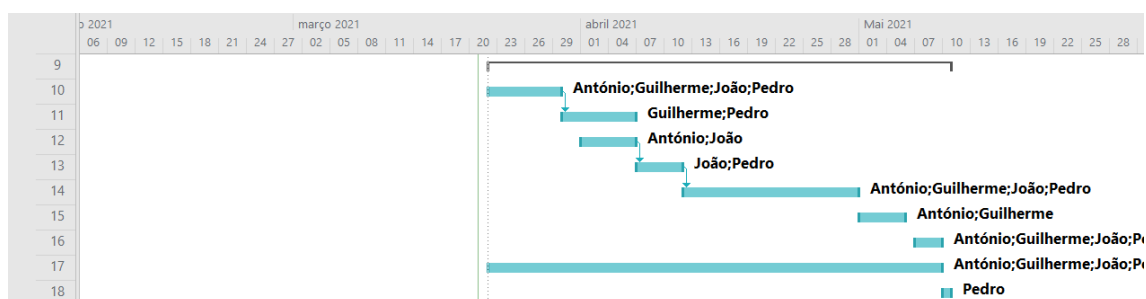


Figura 2.6: Diagrama de Gantt da fase de Especificação.

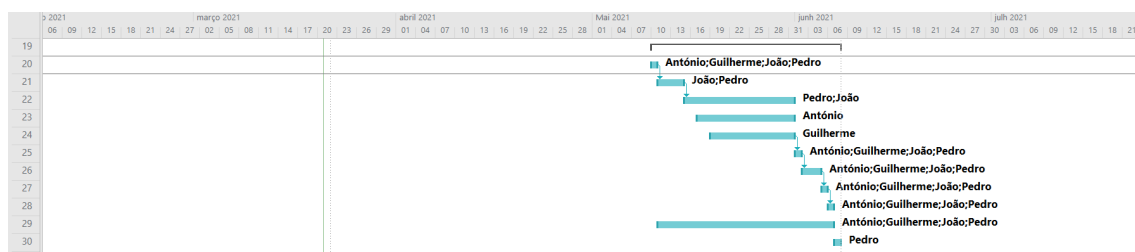


Figura 2.7: Diagrama de Gantt da fase de Construção.

### 3 Fase de Especificação

Tendo terminado a primeira fase do relatório, iremos agora discutir as nossas decisões à cerca da fase de especificação do projeto.

Nesta iremos, inicialmente, analisar todos os requisitos necessários para que o projecto satisfaça as necessidades do "cliente".

De seguida iremos desenvolver os vários modelos referentes às funcionalidades do nosso projeto. Para isto iremos analisar as funcionalidades idealizadas na fase anterior de forma a encontrar todas as interações possíveis entre o sistema e o utilizador. Com isto, iremos então desenvolver o modelo de domínio, bem como os vários diagramas necessários (de Sequência , de *Use Case*, entre outros) para organizar o funcionamento do nosso futuro trabalho.

Após isto, iremos analisar a base de dados que teremos de criar, fazendo então um modelo da mesma.

Finalmente, iremos apresentar um exemplo de possível interface, tendo em conta todas as funcionalidades que o projeto final deverá ter.

## 3.1 Análise de Requisitos

Começamos então por fazer o levantamento e análise dos requisitos do sistema. Observando a nossa idealização do projeto efetuada na parte anterior deste relatório conseguimos chegar a uma lista de funcionalidades necessárias para que o sistema seja o mais fiel ao planeamento. Para além disto, ao longo do desenvolvimento do trabalho houve a necessidade de alterar algumas funcionalidades ou até mesmo adicionar, sempre com a finalidade de melhorar o serviço prestado do produto final, sem se afastar daquilo que tinha sido inicialmente planeado.

### 3.1.1 Requisitos Funcionais

#### ***Consultar Jogo***

1. O Sistema deve conseguir apresentar a informação específica de cada jogo, quer esteja a decorrer, tenha terminado ou esteja agendado;
2. Caso o jogo seja agendado, o Sistema deve conseguir apresentar a data, a hora e os Clubes que vão participar;
3. Caso o jogo esteja a decorrer, o Sistema deve conseguir apresentar a informação do mesmo, sendo esta atualizada em tempo (semi) real;
4. Caso o jogo tenha terminado, o Sistema deve conseguir apresentar toda a informação referente ao jogo;
5. O Sistema deverá permitir pesquisar eventos a partir da data de realização;

#### ***Procurar Equipa ou Jogo***

1. O Utilizador pode pesquisar informação na aplicação;
2. O Sistema deverá conseguir, com a informação fornecida pelo Utilizador, seja esta nomes de equipas ou jogos em específico, procurar e apresentar uma lista de jogos retirada da sua base de dados;

#### ***Consultar Equipa***

1. O Sistema deverá conseguir guardar a informação referente aos jogos das várias equipas;

2. O Sistema deverá permitir pesquisar informação a partir do nome da equipa desejada;

### ***Consultar liga***

1. O Utilizador poderá consultar as informações referentes às ligas presentes na base de dados;
2. O Sistema deverá conseguir obter a informação da liga pretendida;
3. O Sistema deverá conseguir ter a informação referente às equipas, os jogos e as classificações;
4. O Sistema deverá conseguir guardar informação referente aos melhores marcadores de cada liga;

### ***Registo de eventos no Mapa***

1. O utilizador poderá visualizar eventos utilizando um mapa;
2. O Sistema deverá ser capaz de guardar informação referente à localização dos eventos;
3. O Sistema deverá conseguir representar a informação num mapa;

### **Registo**

1. O Utilizador ficará registado no sistema, permitindo depois iniciar sessão;
2. O Sistema deverá apresentar ao utilizador os campos necessários para criar a conta (nome utilizado para login, email, e palavra-passe);
3. O Sistema deverá verificar se os dados são válidos e se preencheu todos os dados;
4. O Sistema deverá ficar com a informação referente ao novo utilizador na sua base de dados;

### ***Login***

1. O Sistema deverá apresentar ao utilizador os campos para este introduzir a informação necessária para efetuar o *login* (nome e palavra passe);
2. O Utilizador deve introduzir as informações necessárias para fazer *login*;
3. O Sistema deverá informar se a informação introduzida for inválida;

4. O Sistema deverá informar o utilizador quando este conseguir iniciar a sessão

### ***Edição dos favoritos***

1. O Utilizador poderá ter uma lista de Favoritos, podendo alterar esta da forma que desejar;
2. O Sistema deverá permitir ao utilizador adicionar e/ou remover equipas desta lista;
3. O Sistema poderá enviar notificações ao utilizador, baseadas nos eventos relacionados com estas equipas;
4. O Sistema poderá enviar notificações via *email* ao utilizadores quando alguma informação dos seu favoritos seja atualizada;

### ***Editar definições de Perfil***

1. O Utilizador pode alterar a informação referente à sua conta;
2. O Sistema deverá verificar se a informação inserida é ou não válida;
3. O Sistema deverá ser capaz de alterar a informação que o utilizador pedir (nome, email e palavra-passe);

## **3.1.2 Requisitos não Funcionais**

Para além dos requisitos já explorados, temos também de identificar os requisitos não funcionais, ou seja, aqueles que o sistema deverá conseguir responder, independentemente das interações com os utilizadores. Temos então os seguintes requisitos:

1. A aplicação deve ser simples e apelativa, visto ser uma aplicação que abrange faixas etárias muito distintas, e também por se tratar de um software que apresenta muita informação, sendo necessário implementar algo apelativo;
2. O Sistema deverá correr em todos os browsers;
3. A aplicação deverá estar online 24 horas por dia, sete dias por semana;
4. A aplicação deverá ser atualizada em tempo real;

## 3.2 Especificação do Sistema

### 3.2.1 Modelo do Domínio

Após uma extensa análise de requisitos, é possível agora recorrer ao desenvolvimento do modelo do domínio. O modelo enunciado representa uma parte importante da especificação, sendo caracterizado pela capacidade de perspectivar uma visão idealizada daquela que é a realidade. Para além disto, ajuda a comunicar as ideias de forma simplificada e a documentar e organizar o raciocínio a um possível cliente e/ou colegas de trabalho. Por isto, podemos sintetizar, com o modelo apresentado abaixo, as 5 entidades principais do sistema, o utilizador, os jogos, as equipas, as ligas e os jogadores.

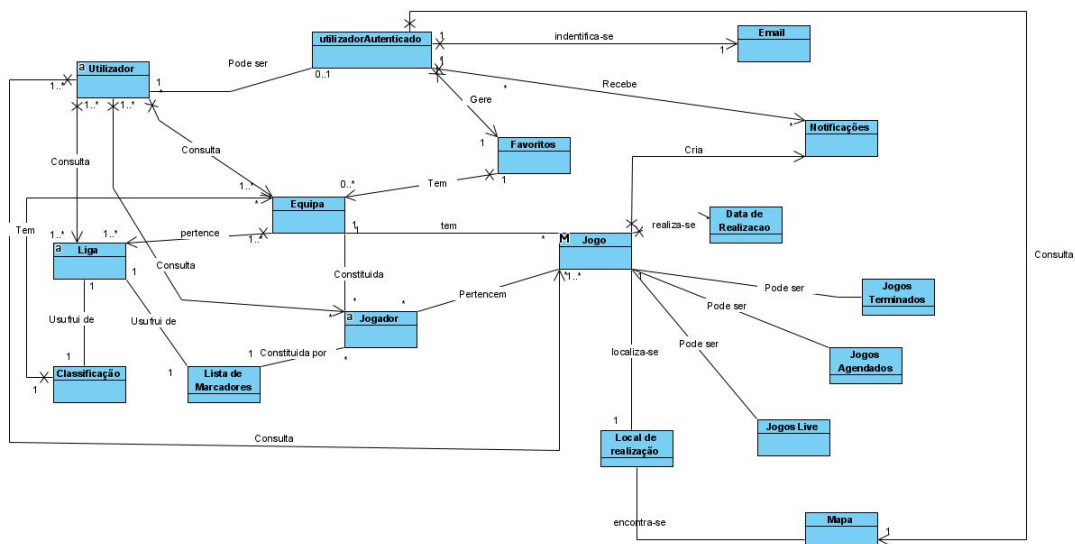


Figura 3.1: Modelo de Domínio.

### 3.2.2 Diagrama de Use Cases

Após a realização da análise dos requisitos do nosso sistema, segue-se a fase do desenvolvimento do diagrama de Use Case. O sistema a ser implementado será representado por dois actores: um utilizador comum, que tem acesso a toda a informação, e um utilizador autenticado, que tem certos privilégios referidos de seguida.

De forma a simplificar a explicação dos Use Case, podemos dividir estes em 3 categorias fundamentais: as funcionalidades básicas, as funcionalidades de monitorização e por fim as funcionalidades extra do utilizador autenticado.

Com isto, a primeira categoria representa as funcionalidades básicas, que correspondem aos use case de Registo, login e editar configurações, sendo que os dois últimos necessitam que o utilizador esteja autenticado.

Em segundo, temos as funcionalidades de consulta que podem ser consequentemente divididas em 3 grupos, as consultas de jogos, de equipas e de ligas. Para uma consulta mais detalhada, cada um destes use case contém também funcionalidades embutidas, e por isso, decidimos usar includes para definir a relação de dependência entre os mesmos. Assim, temos o Use Case Consulta de jogos que consequentemente inclui os use case de consulta por data e por *Status* (Agendado, ao Vivo ou Terminado). Por sua vez, o use case de consulta por ligas inclui também a consulta por classificação e a consulta dos melhores marcadores da respectiva liga. Por último, a consulta por equipas inclui também a consulta dos jogos por *Status*.

Por fim, temos as funcionalidades extra, que são exclusivas do utilizador autenticado. Em primeiro lugar, temos o use case consulta de eventos no mapa, que permite ao utilizador visualizar num mapa locais referenciados que correspondem a jogos agendados. Em segundo lugar temos as funcionalidades de incremento e remoção de equipas da lista de favoritos desenhada especialmente para o utilizador autenticado. Finalmente, temos o Use case das notificações que se define na apresentação em tempo real de momentos importantes das suas equipas favoritas.

Apresentamos de seguida o diagrama de use case, que representa todos os use cases referidos anteriormente.

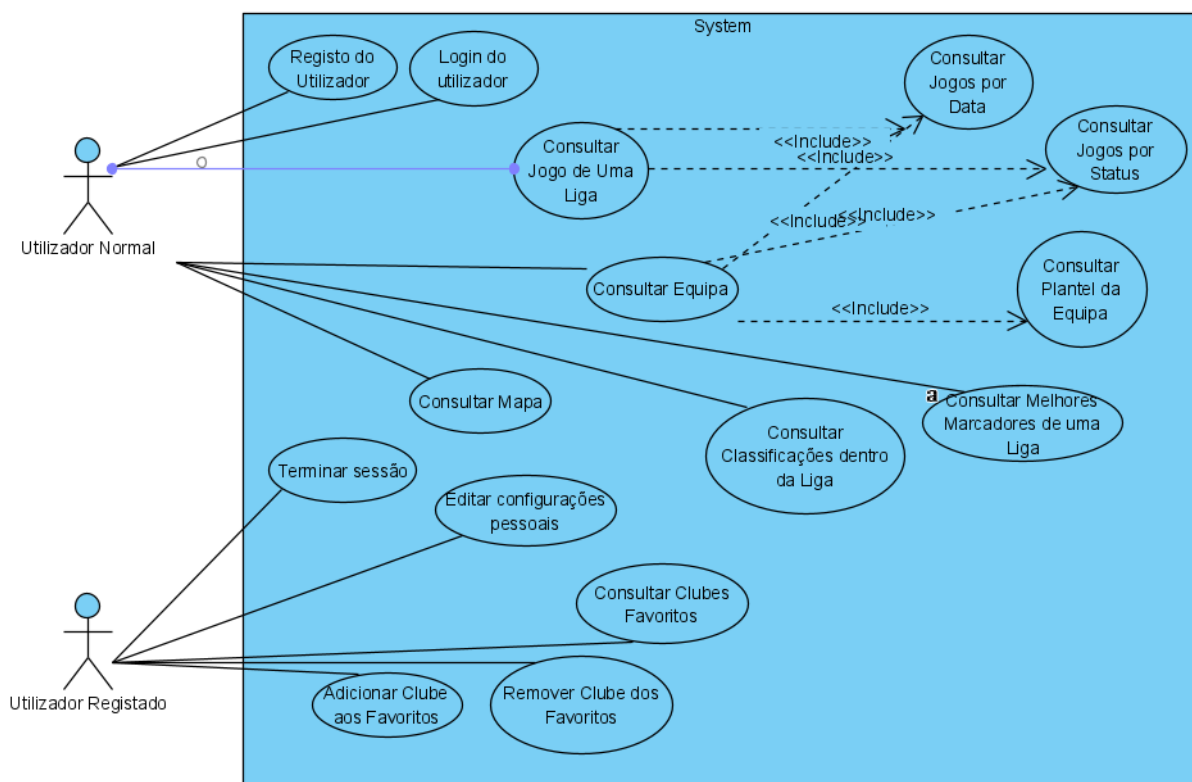


Figura 3.2: Diagrama de Use Case



### 3.2.3 Especificação dos Use Cases

De seguida iremos analisar os use cases mais detalhadamente, bem como as pré e pós condições necessárias para que este possam ocorrer no sistema.

<b>Função</b>	Consultar jogos de uma liga por data.		
<b>Descrição</b>	Sistema apresenta uma lista de jogos de uma determinada liga que ocorreram a uma determinada data.		
<b>Pré-condição</b>	-		
<b>Pós-condição</b>	É apresentada uma lista de jogos realizados no dia selecionado.		
		<b>Utilizador</b>	<b>Sistema</b>
<b>Eventos</b>	1.	Indica que pretende consultar os jogos de uma data;	
	2.		Apresenta um menu para seleccionar a data e a liga;
	3.	Seleciona a data e a liga que pretende;	
	4.		Apresenta ao utilizador os jogos que ocorreram nessa mesma data;
<b>Exceção 1</b> [Não existem jogos nessa data] (passo 5)	5.		Indica que não existem jogos nessa data;

Figura 3.3: Consultar jogos de uma Liga por Data

O Sistema pode apresentar uma lista de jogos de uma determinada liga, que ocorreram ou vão ocorrer numa determinada data, dos quais o utilizador pode escolher.

<b>Função</b>	Consultar jogos de uma liga por Status.		
<b>Descrição</b>	Sistema apresenta informação de jogos de uma liga com o Status que o utilizador pretender.		
<b>Pré-condição</b>	-		
<b>Pós-condição</b>	É apresentada a informação sobre vários jogos.		
		<b>Utilizador</b>	<b>Sistema</b>
<b>Eventos</b>	1.	Indica que pretende consultar jogos conforme o Status;	
	2.		Apresenta um menu para o Utilizador seleccionar o Status e a Liga;
	3.	Seleciona o Status e a Liga pretendida;	
	4.		Apresenta a lista de Jogos da Liga com o Status pretendido;
<b>Exceção 1</b> [ Não existe nenhum jogo com o Status pretendido] (passo 4)	5.		Informa que não há jogos com o Status pretendido;

Figura 3.4: Consultar jogo de uma liga por Status

O Utilizador pode escolher se quer ver Jogos ao Vivo, agendados ou terminados.

<b>Função</b>	Consultar jogos de um Clube por Status.		
<b>Descrição</b>	Sistema apresenta informação de jogos de um Clube com o Status que o utilizador pretender.		
<b>Pré-condição</b>	-		
<b>Pós-condição</b>	É apresentada a informação sobre vários jogos.		
		<b>Utilizador</b>	<b>Sistema</b>
<b>Eventos</b>	1.	Indica que pretende consultar jogos de um Clube conforme o Status;	
	2.		Apresenta um menu para o Utilizador seleccionar o Status e o Clube;
	3.	Selecciona o Status e o Clube pretendido;	
	4.		Apresenta a lista de Jogos do Clube com o Status pretendido;
<b>Exceção 1</b> [ Não existe nenhum jogo do Clube com o Status pretendido] (passo 4)	5.		Informa que não há jogos com o Status pretendido;

Figura 3.5: Consultar jogos de um Clube com determinado Status

O utilizador poderá procurar jogos com diferentes Status para cada Clube.

<b>Função</b>	Consultar jogos de um Clube por data.		
<b>Descrição</b>	Sistema apresenta uma lista de jogos de um determinado Clube que ocorreram a uma determinada data.		
<b>Pré-condição</b>	-		
<b>Pós-condição</b>	É apresentada uma lista de jogos realizados no dia seleccionado.		
		<b>Utilizador</b>	<b>Sistema</b>
<b>Eventos</b>	1.	Indica que pretende consultar os jogos de uma data de um Clube específico;	
	2.		Apresenta um menu para seleccionar a data e o Clube;
	3.	Selecciona a data e o Clube que pretende;	
	4.		Apresenta ao utilizador os jogos que ocorreram nessa mesma data e do Clube seleccionado;
<b>Exceção 1</b> [Não existem jogos desse Clube nessa Data] (passo 4)	5.		Indica que não existem jogos nessa data;

Figura 3.6: Consultar jogos de uma Equipa numa determinada Data

O sistema tem a capacidade de permitir ao utilizador pesquisar jogos de um determinado Clube para uma data específica.

<b>Função</b>	Consultar plantel de uma equipa.		
<b>Descrição</b>	Sistema apresenta os constituintes de uma determinada equipa escolhida pelo utilizador.		
<b>Pré-condição</b>	-		
<b>Pós-condição</b>	São apresentados os elementos de uma equipa.		
		<b>Ator</b>	<b>Sistema</b>
<b>Eventos</b>	1.	O utilizador insere o nome da equipa no motor de busca.	
	2.		O sistema apresenta as possibilidades de escolha relacionadas com a pesquisa.
	3.	O utilizador seleciona a equipa pretendida.	
	4.		O sistema apresenta os elementos constituintes da equipa referida.
<b>Exceção 1</b> [Não apresenta nenhum resultado] (passo 2)	5.		Informa o utilizador que não é possível encontrar nenhuma equipa de acordo com os parâmetros de pesquisa.

Figura 3.7: Consultar plantel de uma Equipa

O sistema tem a capacidade de permitir ao utilizador pesquisar uma determinada equipa e apresentar o plantel dessa mesma equipa.

<b>Função</b>	Consultar classificação de uma liga		
<b>Descrição</b>	O sistema apresenta as classificações de uma liga		
<b>Pré-condição</b>	-		
<b>Pós-condição</b>	É apresentada a classificação de uma liga		
		<b>Utilizador</b>	<b>Sistema</b>
<b>Eventos</b>	1.	O utilizador seleciona aba correspondente às classificações	
	2.		O sistema apresenta a classificação atual

Figura 3.8: Consultar classificações de uma liga

O utilizador após selecionar uma liga, tem a possibilidade de visualizar a classificação da mesma.

Função	Consultar melhores marcadores de uma liga	
Descrição	O sistema apresenta os melhores marcadores de uma determinada liga	
Pré-condição	-	
Pós-condição	É apresentada uma lista de jogadores e respetivos golos marcados	
	Utilizador	Sistema
Eventos	1. O utilizador seleciona a aba dos melhores marcadores	
	2.	O sistema apresenta os melhores marcadores atuais

Figura 3.9: Consultar melhores marcadores de uma liga

O utilizador para além da classificação da liga, pode também consultar os seus melhores marcadores.

Função	Consultar eventos no mapa	
Descrição	O utilizador visualiza um mapa com checkpoints de jogos agendados	
Pré-condição	-	
Pós-condição	São apresentados jogos agendados num mapa	
	Utilizador	Sistema
Eventos	1. O utilizador escolhe a opção de apresentar mapa	
	2.	O sistema apresenta um mapa com checkpoints
	3. O utilizador seleciona um checkpoint	
	4.	O sistema apresenta informações acerca do jogo selecionado.
Exceção1[O jogo selecionado já se encontra a decorrer] passo 3	5.	O sistema informa que o jogo já se encontra a decorrer.

Figura 3.10: Consultar eventos no mapa

O sistema tem integrada a possibilidade de qualquer utilizador visualizar um mapa com os jogos agendados.

<b>Função</b>	Registrar utilizador.	
<b>Descrição</b>	Utilizador regista-se na aplicação com os seus dados.	
<b>Pré-condição</b>	O utilizador não está registado.	
<b>Pós-condição</b>	O utilizador fica registado.	
	<b>Ator</b>	<b>Sistema</b>
<b>Eventos</b>	1.	Indica que pretende registar-se na aplicação.
	2.	Apresenta o menu de registo.
	3.	Insere o nome de utilizador, e-mail e palavra-passe.
	4.	Verifica se todos os campos foram preenchidos.
	5.	Verifica o nome de utilizador.
	6.	Verifica e-mail.
	7.	Verifica palavra-passe.
	8.	Regista a informação.
	9.	Informa o utilizador que o registo foi feito com sucesso.
<b>Exceção 1</b> [Campos por preencher] (passo 4)	10.	Informa o utilizador que existem campos por preencher.
<b>Exceção 2</b> [Nome de utilizador inserido já utilizado] (passo 5)	11.	Informa o utilizador que o registo não é possível porque o nome de utilizador inserido já existe.
<b>Exceção 3</b> [E-mail inserido inválido] (passo 6)	12.	Informa o utilizador que o registo não é possível porque e-mail inserido não é válido pois não contém o "@" e o ".".
<b>Exceção 4</b> [E-mail inserido já utilizado] (passo 6)	13.	Informa o utilizador que o registo não é possível porque o e-mail inserido já está a ser utilizado.
<b>Exceção 5</b> [Palavra-passe inserida inválida] (passo 7)	14.	Informa o utilizador que o registo não é possível pois a palavra-passe inserida tem menos de 8 caracteres.

Figura 3.11: Registrar na aplicação

O utilizador normal, tem a possibilidade de poder registar-se na aplicação, tendo como vantagem um conjunto de novas funcionalidades.

<b>Função</b>	Login do utilizador.		
<b>Descrição</b>	Utilizador entra na aplicação com os seus dados de registo.		
<b>Pré-condição</b>	O utilizador já está registado.		
<b>Pós-condição</b>	O utilizador fica autenticado na aplicação.		
	<b>Ator</b>	<b>Sistema</b>	
<b>Eventos</b>	1.	Indica que pretende autenticar-se na aplicação.	
	2.		Apresenta o menu de login.
	3.	Insere o seu nome de utilizador/e-mail e palavra-passe.	
	4.		Verifica se todos os campos foram preenchidos.
	5.		Verifica o nome de utilizador/e-mail
	6.		Verifica palavra-passe.
	7.		Informa o utilizador que foi autenticado na aplicação.
<b>Exceção 1</b> [Campos por preencher] (passo 4)	8.		Informa o utilizador que existem campos por preencher.
<b>Exceção 2</b> [Nome de utilizador/E-mail inválido] (passo 5)	9.		Informa o utilizador que o registo não é possível porque o nome de utilizador/e-mail inserido não existe na base de dados.
<b>Exceção 3</b> [Palavra-passe inserida inválida] (passo 6)	10.		Informa o utilizador que o registo não é possível pois a palavra-passe inserida não corresponde ao nome de utilizador/e-mail inserido.

Figura 3.12: Login na aplicação

O utilizador registado, tem a possibilidade de poder autenticar-se na aplicação, para aceder às funcionalidades destinadas aos utilizadores registados.

<b>Função</b>	Consultar favoritos.		
<b>Descrição</b>	Utilizador pretende consultar os jogos das equipas que tem como favoritas.		
<b>Pré-condição</b>	O utilizador está autenticado na aplicação.		
<b>Pós-condição</b>	Os jogos das equipas favoritas são apresentados ao utilizador.		
	<b>Ator</b>	<b>Sistema</b>	
<b>Eventos</b>	1.	Indica que pretende consultar os favoritos.	
	2.		Verifica as equipas favoritas.
	3.		Apresenta ao utilizador todos os jogos das suas equipas favoritas.
<b>Exceção 1</b> [Não possui equipas favoritas] (passo 2)	4.		Informa o utilizador que não é possível consultar os favoritos pois o utilizador não possui equipas favorita.

Figura 3.13: Consultar favoritos

O utilizador autenticado, tem a vantagem de poder consultar os favoritos, isto é, pode aceder a um conjunto de todos os jogos das equipas que seleccionou como favoritos.

<b>Função</b>	Adicionar conteúdo aos favoritos		
<b>Descrição</b>	O utilizador registado adiciona uma equipa aos seus favoritos		
<b>Pré-condição</b>	O utilizador está autenticado		
<b>Pós-condição</b>	O utilizador tem uma nova equipa favorita		
		<b>Utilizador</b>	<b>Sistema</b>
<b>Eventos</b>	1.	O utilizador selecciona a equipa como favorita	
	2.		O sistema adiciona a equipa seleccionada à lista de equipas favoritas
<b>Exceção 1</b> [equipa já se encontra nos favoritos] (passo 4)	3.		O sistema informa que a equipa seleccionada já se encontra na lista

Figura 3.14: Adicionar conteúdo à lista de favoritos

O utilizador autenticado, tem a vantagem de poder adicionar conteúdo, nomeadamente equipas à sua lista de favoritos.

<b>Função</b>	Remove conteúdo dos Favoritos		
<b>Descrição</b>	O utilizador registado remove uma equipa dos Favoritos		
<b>Pré-condição</b>	O utilizador está autenticado		
<b>Pós-condição</b>	O utilizador tem menos uma equipa na lista de favoritos		
		<b>Utilizador</b>	<b>Sistema</b>
<b>Eventos</b>	1.	O utilizador selecciona a opção de remover equipa dos favoritos	
	2.		O sistema retira a equipa da lista de favoritos
	3.		O Sistema apresenta os jogos da equipa seleccionada

Figura 3.15: Remover conteúdo dos Favoritos

O utilizar autenticado tem a possibilidade de remover conteúdo da sua lista de favoritos.



Função	Editar configurações Pessoais		
Descrição	O utilizador registado pretende editar as suas configurações pessoais		
Pré-condição	O utilizador encontra-se autenticado		
Pós-condição	O utilizador modificou as suas configurações de conta		
		Utilizador	Sistema
Eventos	1.	O utilizador seleciona opção de editar perfil	
	2.		O sistema apresenta as opções editáveis do perfil
	3.	O utilizador modifica o seu perfil	
	4.		O Sistema verifica o input.
	5.		O Sistema pede para confirmar
	6.	O utilizador confirma	
	7.		O Sistema aplica as modificações
Exceção 1[o input não corresponde ao esperado] (passo4)			O sistema informa as características do input necessárias
Exceção 1 [o utilizador cancela modificações] (passo 5)	8.		O sistema volta à página inicial

Figura 3.16: Use case editar configurações pessoais

O utilizador autenticado tem a possibilidade de modificar os seus dados pessoais, nomeadamente, nome , password e email.

Função	Terminar sessão.		
Descrição	Utilizador pretende terminar a sessão na aplicação.		
Pré-condição	O utilizador está autenticado na aplicação.		
Pós-condição	O utilizador deixa de estar autenticado na aplicação.		
		Ator	Sistema
Eventos	1.	Indica que pretende terminar sessão.	
	2.		Informa o utilizador que terminou a sessão.

Figura 3.17: Use case terminar sessão.

O utilizador autenticado tem a possibilidade de terminar a sua sessão, deixando de estar autenticado na aplicação.

### 3.2.4 Arquitectura da Solução

De forma a prespetivar a futura implementação foi desenvolvido um pequeno diagrama com o fluxo de informação e as principais partes do nosso sistema.

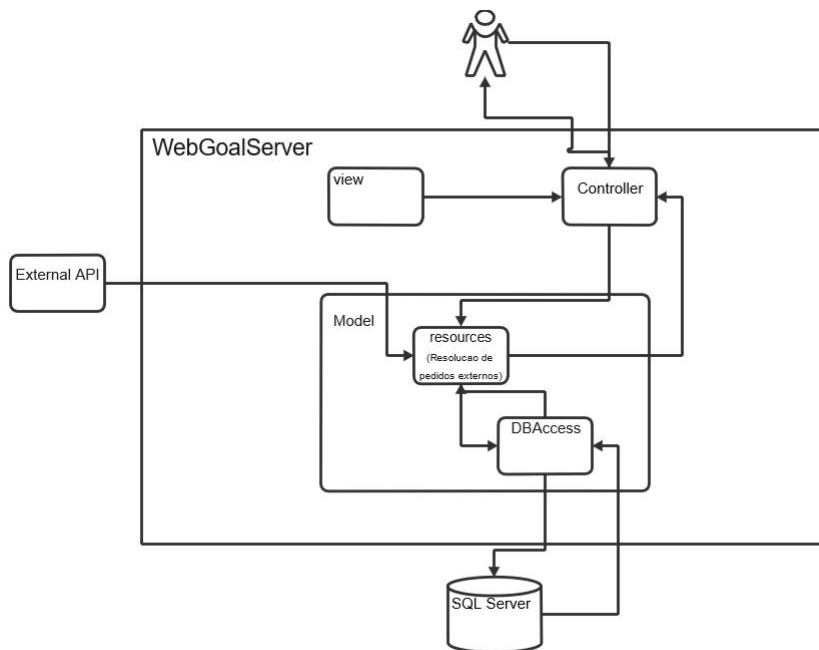


Figura 3.18: Arquitectura do Sistema

### 3.2.5 Diagrama de Sequência de Subsistemas

De forma a especificar os use cases utilizando a arquitetura apresentada anteriormente, criamos os seguintes Diagramas de sequência de Subsistemas:

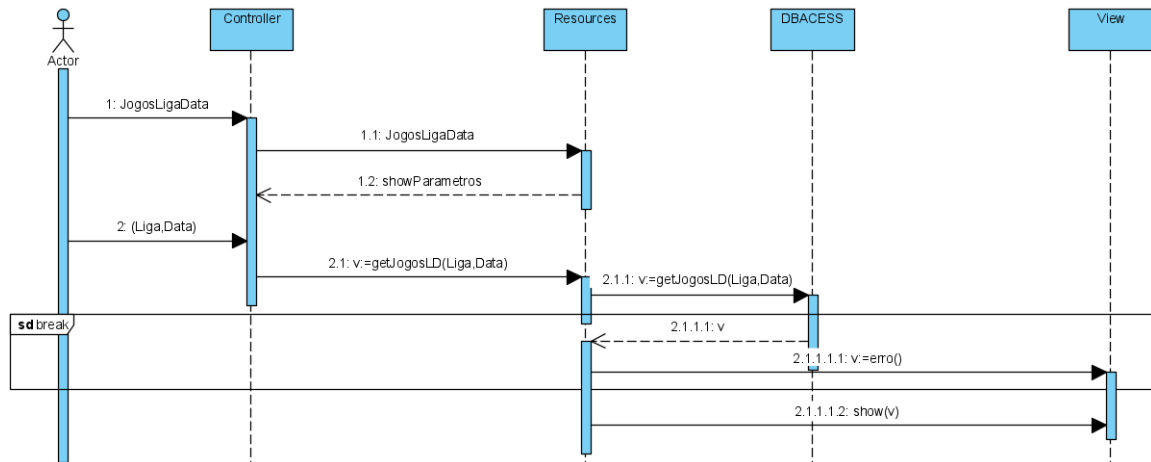


Figura 3.19: DSS Consultar jogos de uma liga por dia

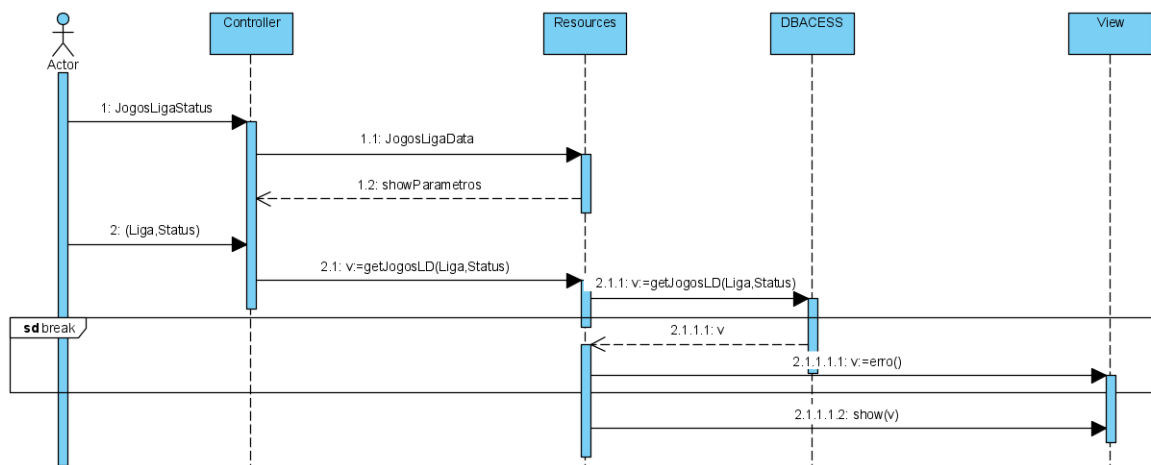


Figura 3.20: DSS Consultar jogos de uma liga por Status

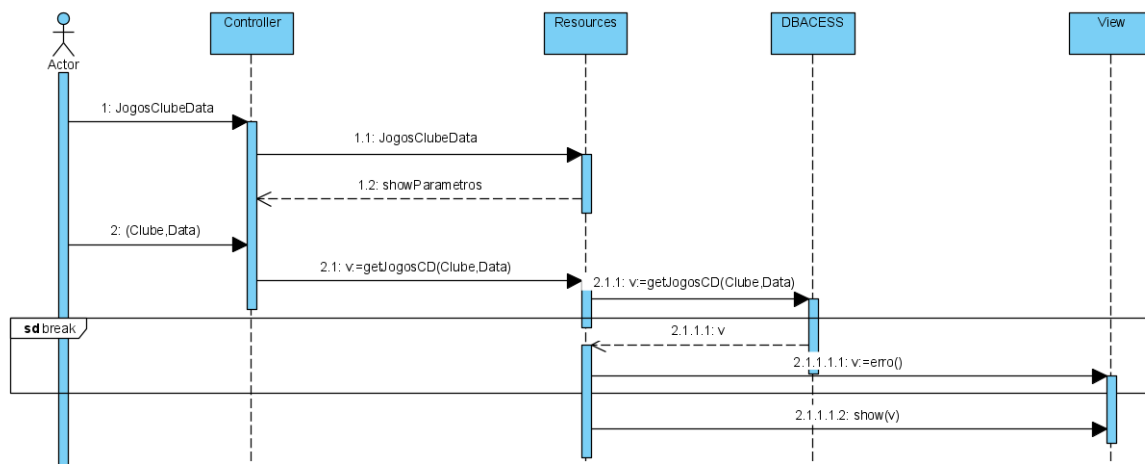


Figura 3.21: DSS Consultar jogos de uma Equipe por data

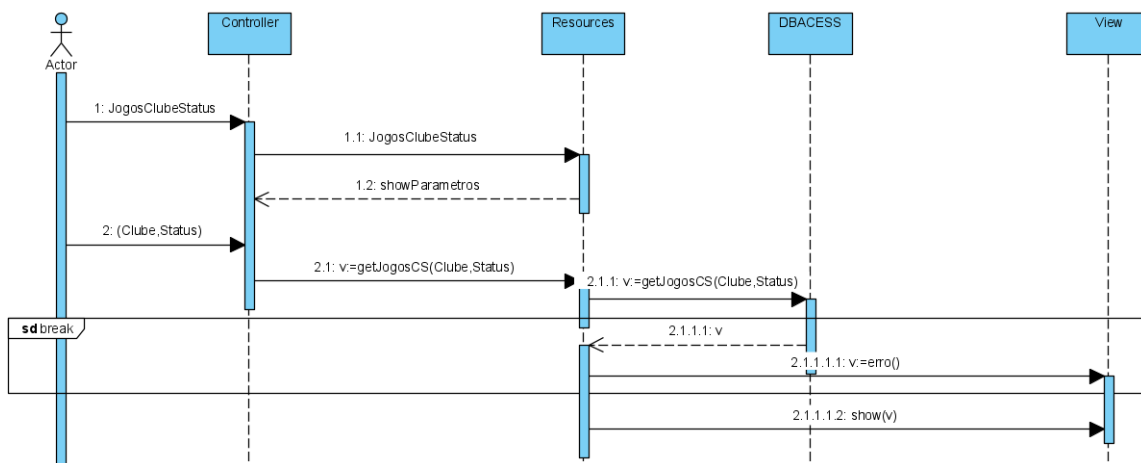


Figura 3.22: DSS Consultar jogos de uma Equipe por Status

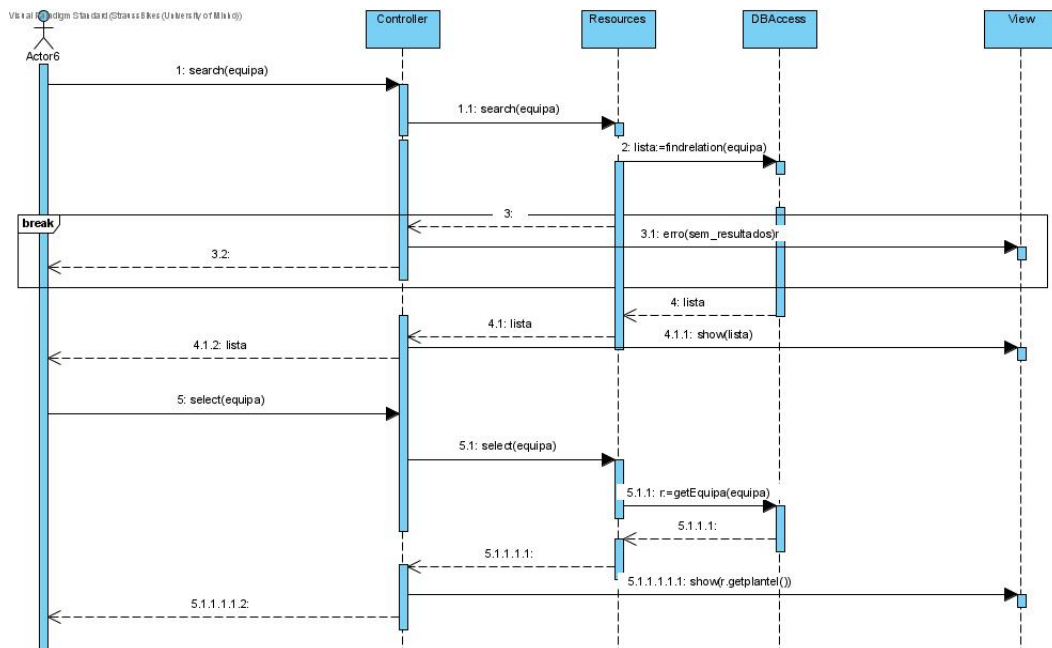


Figura 3.23: DSS Consultar Plantel de uma Equipa

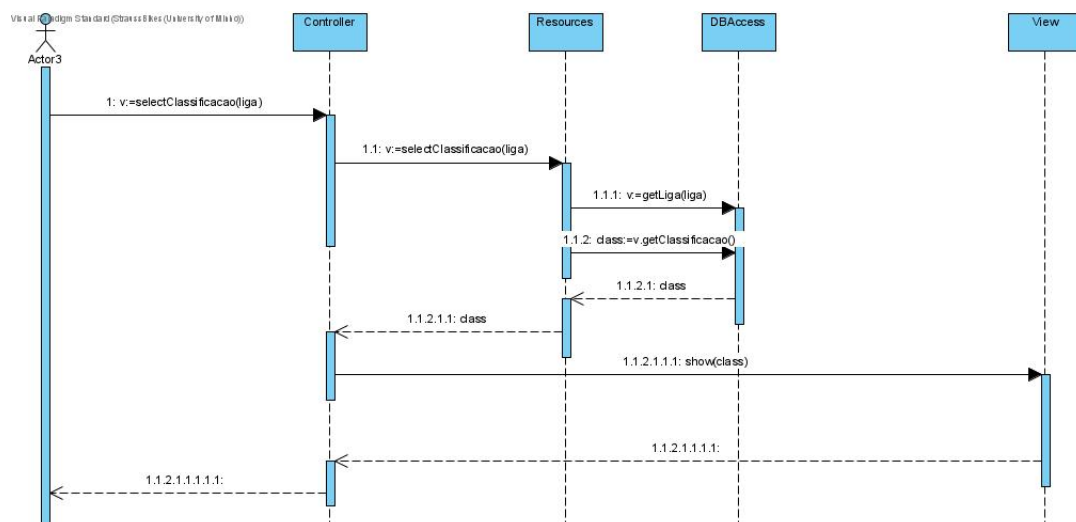


Figura 3.24: DSS Consultar Classificações de uma liga

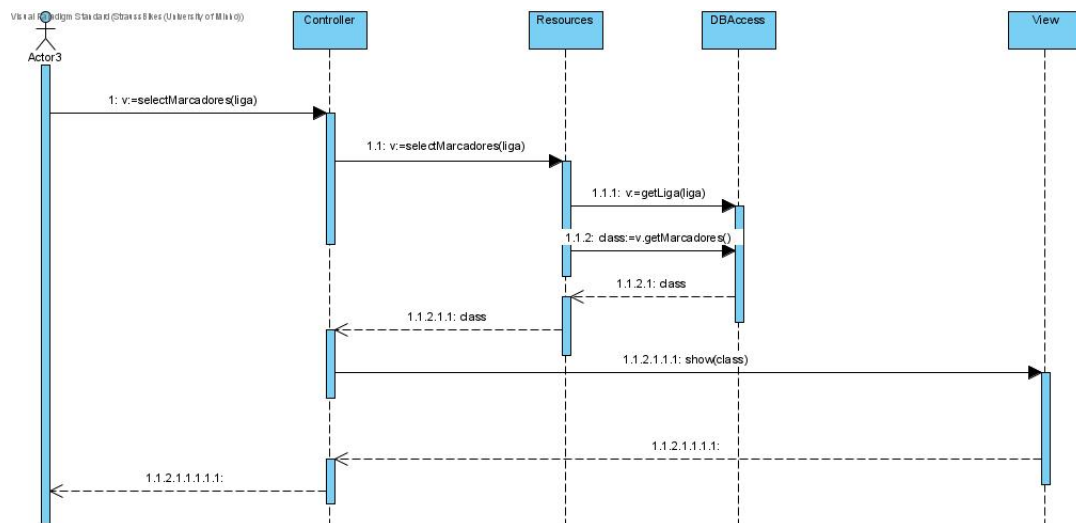


Figura 3.25: DSS Consultar melhores marcadores de uma liga

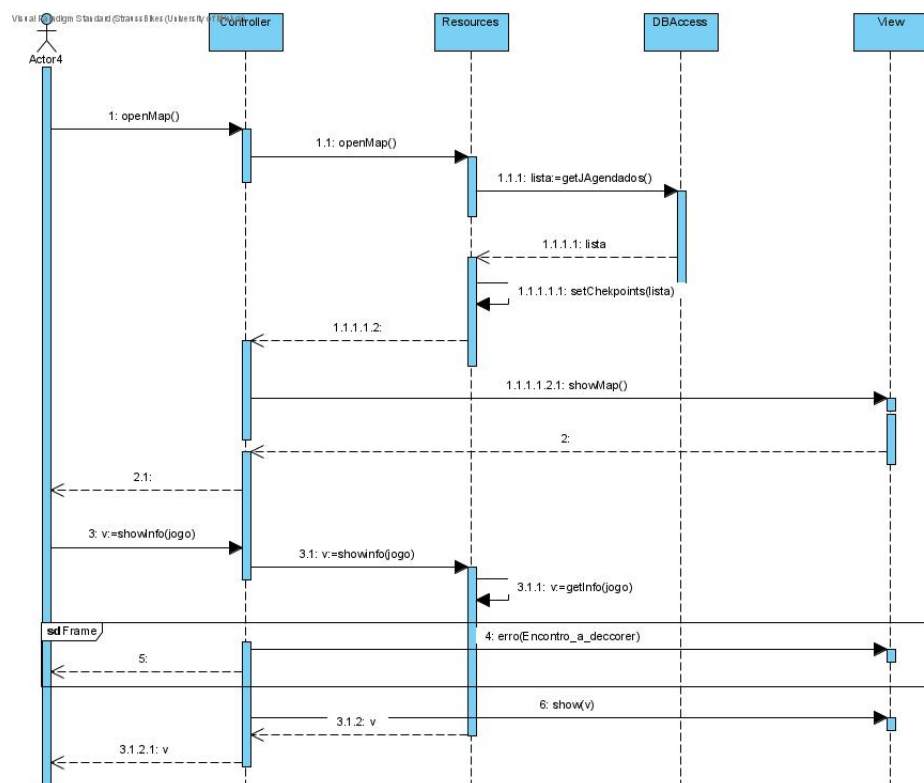


Figura 3.26: DSS Consultar eventos no mapa

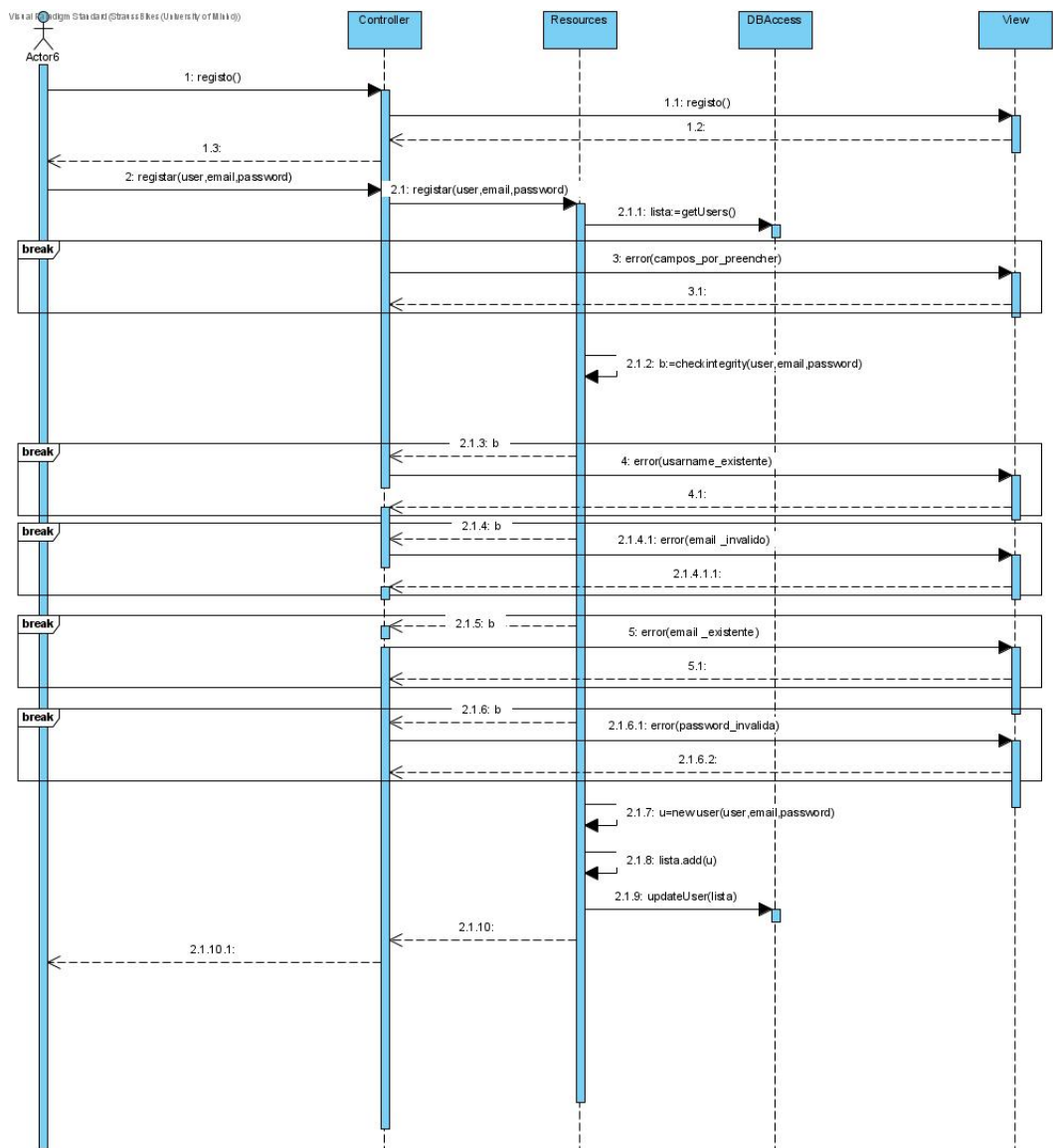


Figura 3.27: DSS Registrar na aplicação

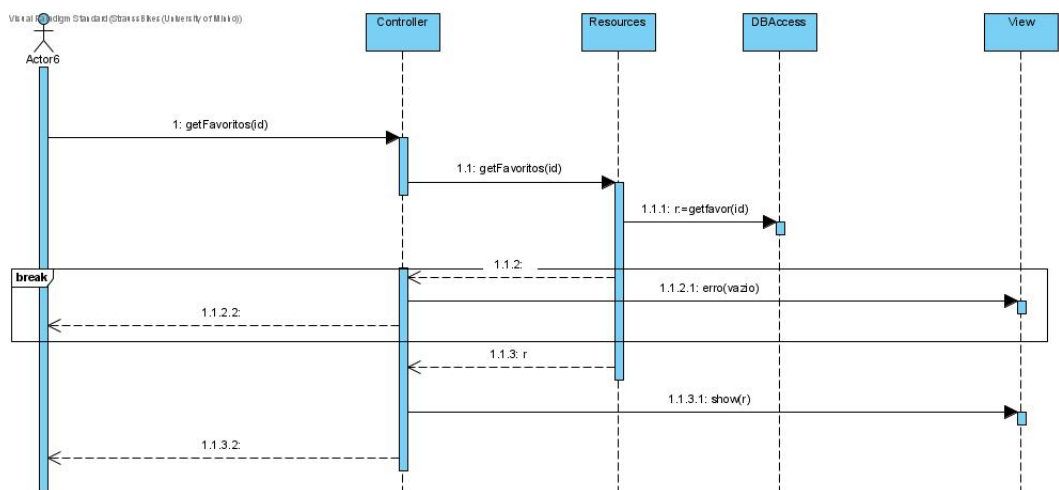


Figura 3.28: DSS Consultar Favoritos

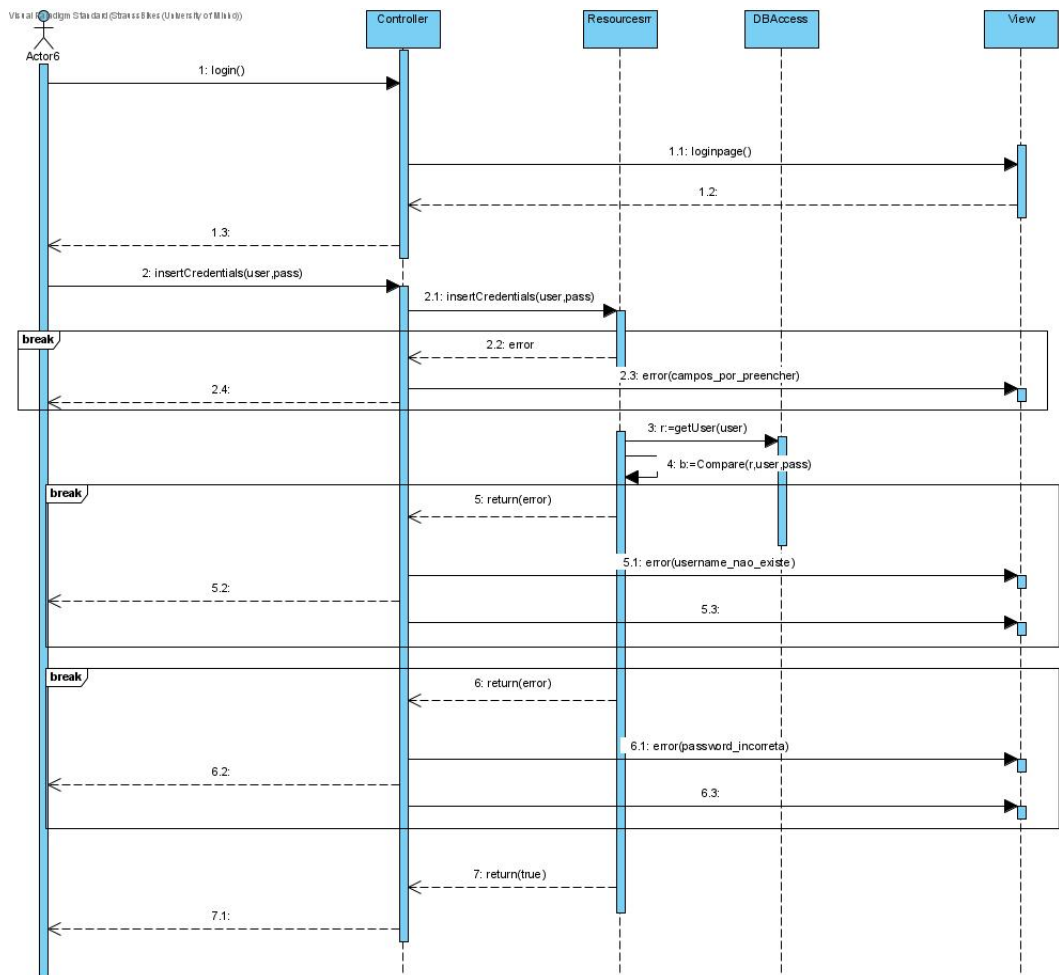


Figura 3.29: DSS Login na aplicação



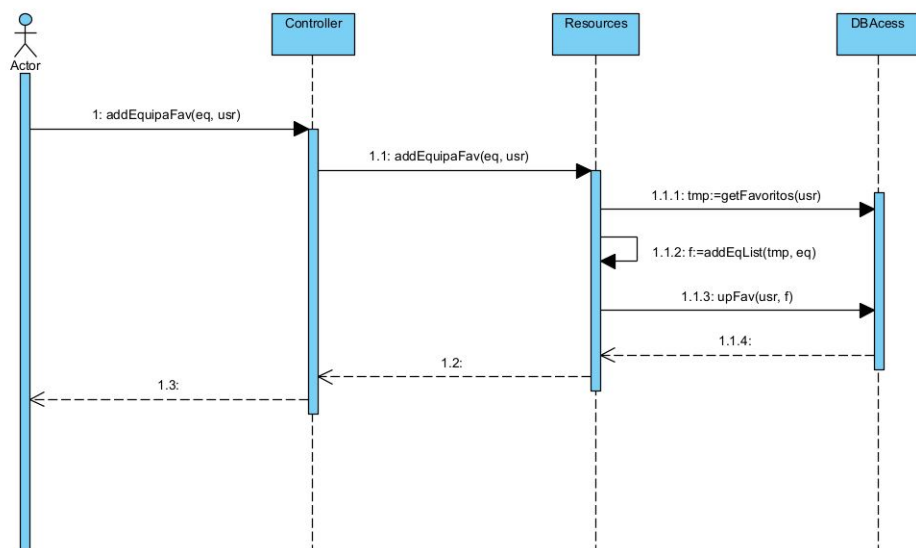


Figura 3.30: DSS Adicionar conteúdo à lista de favoritos

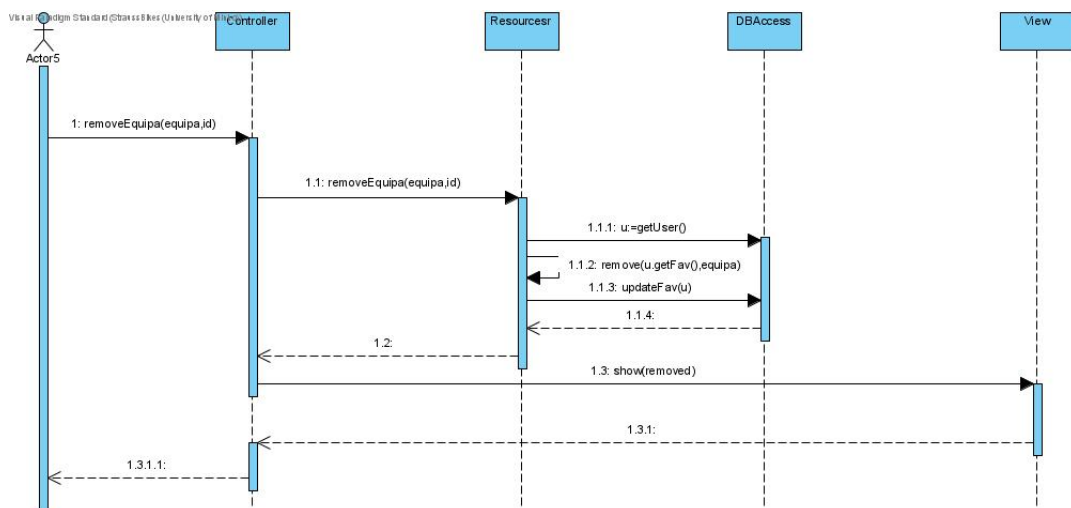


Figura 3.31: DSS Remover conteúdo dos Favoritos

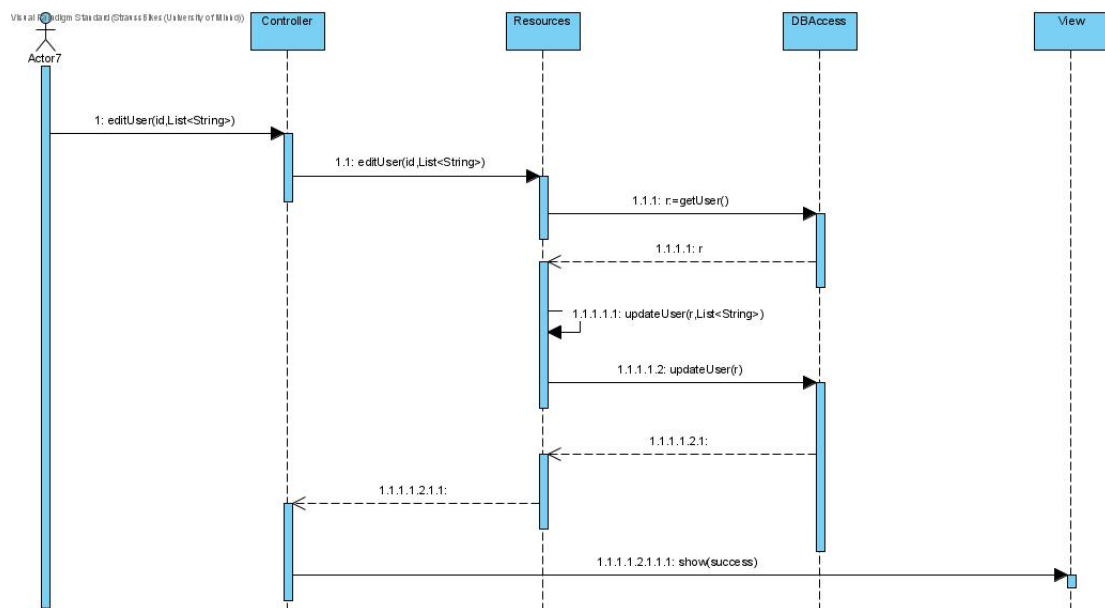


Figura 3.32: DSS Editar Configurações pessoais

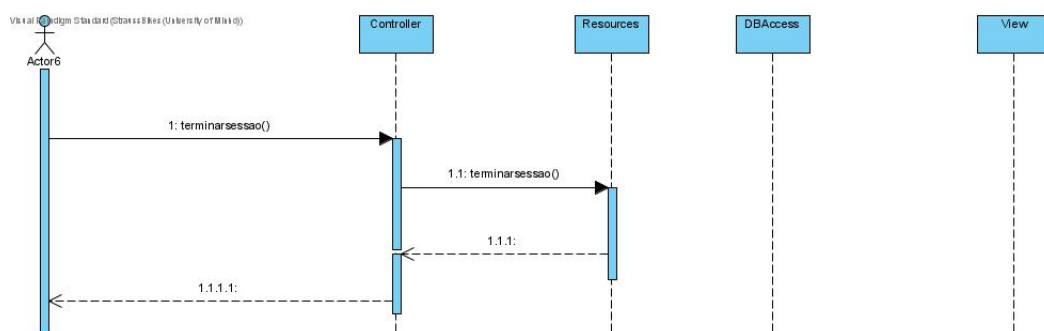


Figura 3.33: DSS Terminar sessão

## 3.3 Arquitetura da Camada de negócios

### 3.3.1 Camada de Negócios

Terminada a parte do levantamento de requisitos, da sua análise e do desenvolvimento do modelo de domínio, é agora necessário desenvolver o diagrama de classes, para isso começamos por fazer uma generalização das classes principais do nosso sistema de forma a projetar melhor o modelo final.

### 3.3.2 Dicionário das principais Classes

1. *WebGoal* será a classe principal responsável pelo agrupamento de toda informação e por sua vez, comunicação com o cliente;
2. *Utilizador* representa os intervenientes do sistema, guardando informações sobre os mesmos;
3. *GestãoConteudo* será responsável pelo gerenciamento da informação relativa as equipas jogadores e ligas;
4. *GestãoUtilizador* Responsável pela administração da informação relativa ao utilizador, nomeadamente favoritos;
5. *GestãoJogos* Responsável pela manutenção de jogos, nomeadamente relativo ao estado do jogo, isto é, *live*, *agendado* ou *terminado*;
6. *Equipa* classe que representa a informação relativa as equipas;
7. *Jogador* classe que representa a informação relativa aos jogadores;
8. *Liga* classe que representa a informação relacionada com as ligas, nomeadamente classificações e marcadores;

### 3.3.3 Descrição da arquitectura

Primeiramente o utilizador tem dois estados possíveis, isto é, pode estar autenticado, adquirindo assim alguns privilégios, ou pode ser um utilizador normal. O utilizador com sessão iniciada no sistema tem a possibilidade de criar a sua lista de equipas favoritas, podendo segui-las de uma forma mais fácil, recebendo inclusive notificações acerca de eventos desportivos relacionados com as mesmas.

Posteriormente, temos duas classes responsáveis pela manipulação de dados. (*GestaoJogos*, *GestaoConteudo*).

A classe *GestaoJogos* tem como objetivo controlar as diferentes coleções, jogos agendados, jogos terminados e jogos em ao vivo.

A classe *GestaoConteudo* tem como intuito controlar a partilha de informação relativa às equipas, aos jogadores e às ligas.

### 3.3.4 Diagrama de Classes

Segue abaixo o diagrama de classes desenvolvido.

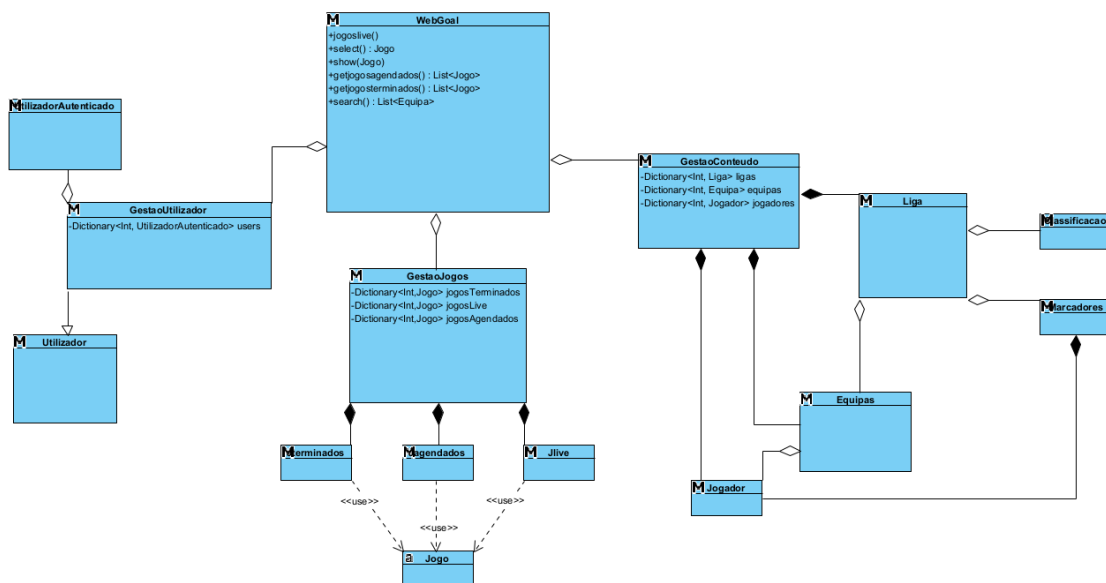


Figura 3.34: Diagrama de Classes

### 3.3.5 Diagrama de ORM

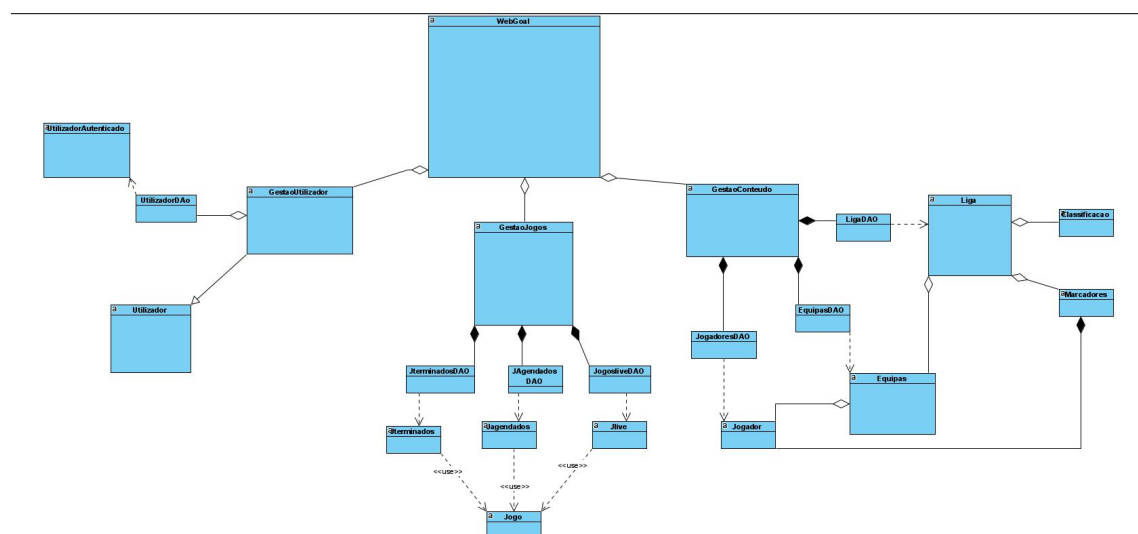


Figura 3.35: Diagrama de Classes com DAO

### 3.4 Diagrama de Pacotes

O diagrama de pacotes permite-nos agrupar as classes em *packages* de acordo com o seu papel na arquitectura da aplicação final, identificando assim as relações de dependência entre os vários componentes do sistema. Este processo torna-se necessário de forma a manter a organização entre as classes e para que seja possível trabalhar cada parte de forma separada.

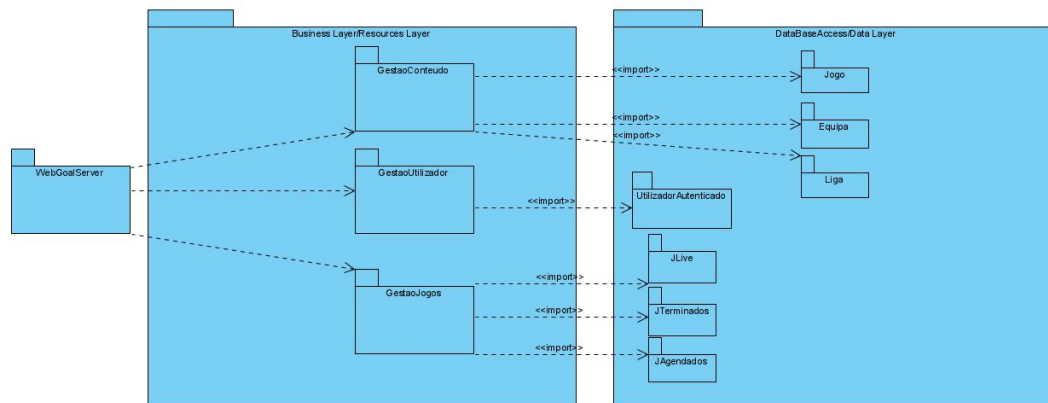


Figura 3.36: Diagrama de Package

Como se pode ver no diagrama a cima, decidimos manter toda a informação da base de dados num único Package, sendo este acedido pela Business Layer, de forma a conseguir responder às várias funcionalidade do sistema.

## 3.5 Projeto da base de dados

O modelo lógico apresentado de seguida foi desenvolvido a partir da estruturação da informação importante a guardar, de forma a que esta informação respeitasse todas as regras associadas ao desenvolvimento de modelos lógicos.

Com isto, foram utilizadas classes como Jogo, Resultados, Clube, Jogador, Liga, Utilizador, Favoritos, assim como todas as relações associadas a estas. Para respeitar os princípios da normalização foi adicionada uma nova tabela, *Favoritos\_has\_Equipa*, que será uma lista com *ids* de favoritos e *id* de equipas de forma a criar a lista de favoritos de um dado utilizador.

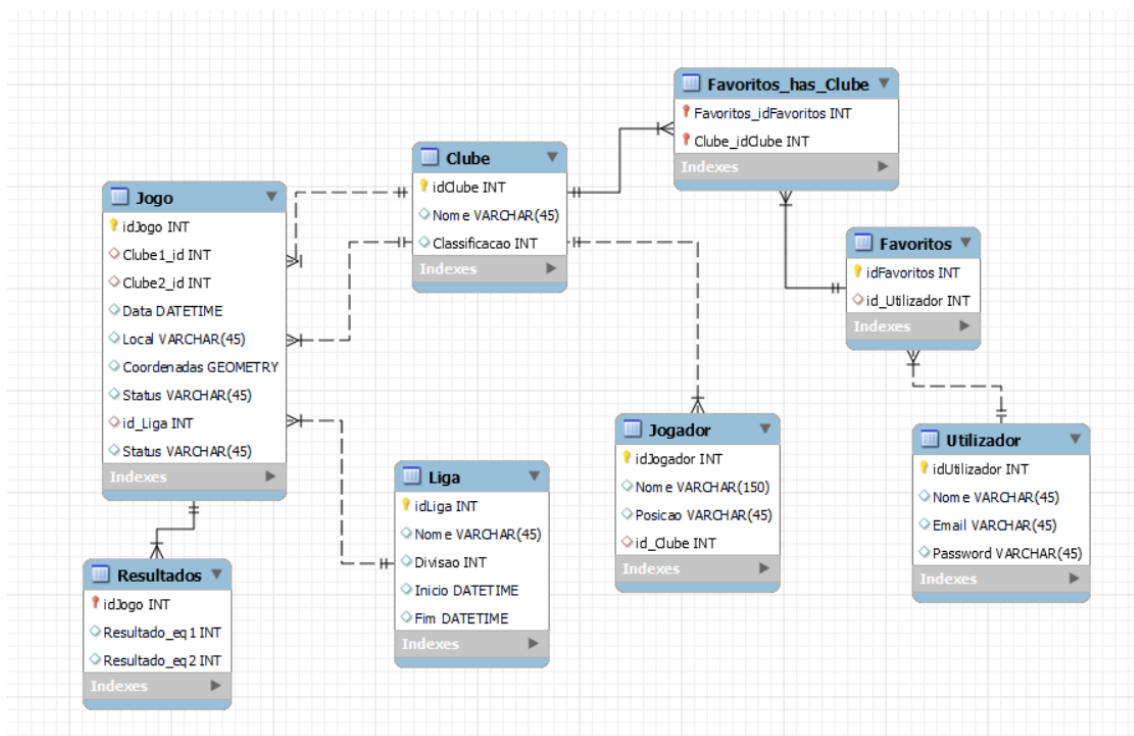



Figura 3.37: Modelo Lógico

### 3.6 Protótipo da interface



Jogos

Ligas

Equipas

Mapa de jogos

Entrar

Registrar

Entrar

Nome de utilizador ou E-mail

utilizador1@hotmail.com


Password

Mais de 8 caractères

Entrar

Cancelar

Figura 3.38: Interface de Login



Jogos

Ligas

Equipas

Mapa de jogos

Entrar

Registrar

Registrar

Nome de Utilizador

utilizador1

✓

E-mail

utilizador1@hotmail.com

✓

Password

Mais de 8 caractères

✗

Registrar

Cancelar

Figura 3.39: Interface de registo de utilizador



Jogos
Ligas
Equipas
Mapa de jogos
Favoritos

Utilizador1  
[Editar dados](#)  
[Terminar sessão](#)

Voltar

3 BARCELONA
87 minutos
REAL MADRID 2

ACONTECIMENTOS

1ª PARTE

10' (GOLO) Messi

15' (GOLO) Messi

23' (Amarelo) Modric

46' (Vermelho) Modric

2ª PARTE

62' (GOLO) Pedri

73' (Substituição) IN-Puig OUT-Pedri

46' (GOLO) Vinicius Jr.

56' (GOLO) Casemiro

ESTATÍSTICAS

20	REMATES	6
8	REMATES À BALIZA	3
68%	POSSE DE BOLA	32%
11	FALTAS	7
9	CANTOS	5
0	CARTÕES VERMELHOS	1
0	CARTÕES AMARELOS	0

EQUIPAS

Real Madrid 4-3-3

Barcelona 4-4-2

Figura 3.40: Interface de terminar sessão

Jogos
Ligas
Equipas
Mapa de jogos
Favoritos


Utilizador1

Todos os jogos
Jogos terminados
Jogos ao vivo
Jogos agendados

05/12/2021

JOGO	LIGA	HORA	RESULTADO	LOCALIZAÇÃO
Porto-Benfica	Liga NOS	20:00H	3-1	Estádio do Dragão , Porto

Figura 3.41: Interface da consulta de jogos por data



Jogos
Ligas
Equipas
Mapa de Jogos
Favoritos

Utilizador1

Jogos Terminados
Jogos Agendados
Equipas

Manchester City

JOGO	LIGA	HORA	RESULTADO	LOCALIZAÇÃO
Manchester City - PSG	Champions League	21:15H	1-1	Ethiad Stadium , Manchester

Figura 3.42: Interface da consulta de jogos por nome de equipa

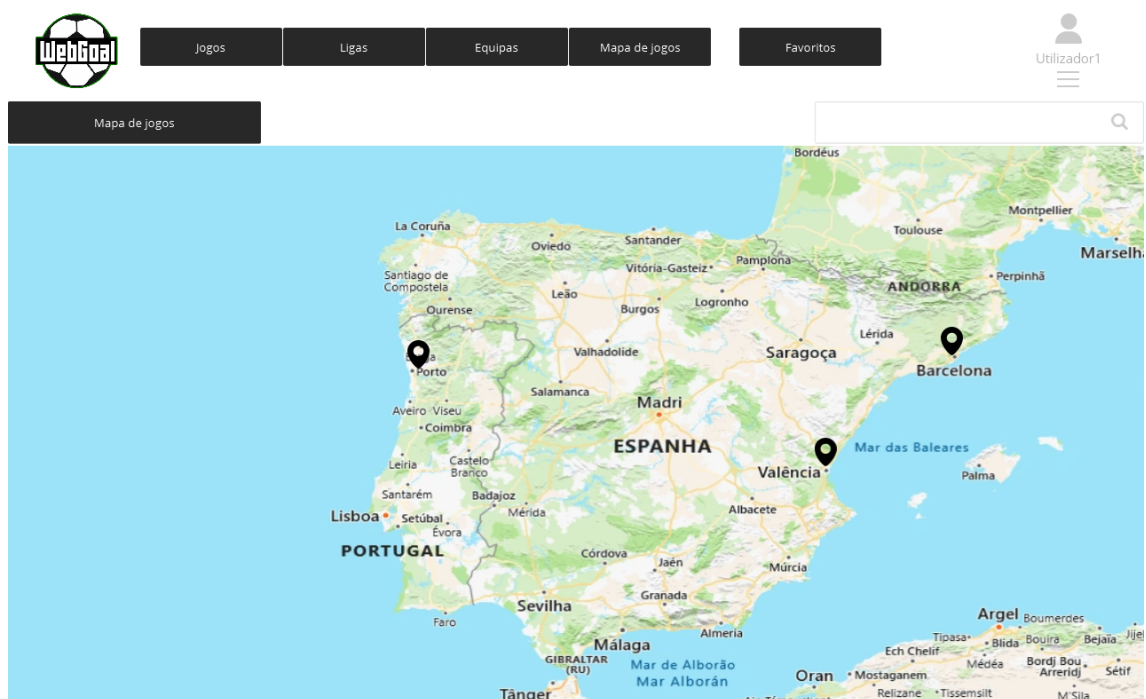


Figura 3.43: Interface do mapa com os jogos marcados

Jogos
Ligas
Equipas
Mapa de jogos
Favoritos

utilizador1

Voltar

BARCELONA

20:15H

REAL MADRID

DATA E LOCALIZAÇÃO

Camp Nou , Barcelona , Espanha

22 de dezembro de 2021

Confrontos

Resultado	Data
BARCELONA 2 - 1 REAL MADRID	22 de novembro de 2021
REAL MADRID 3 - 1 BARCELONA	4 de outubro de 2021
BARCELONA 2 - 2 REAL MADRID	3 de maio de 2021
BARCELONA 0 - 1 REAL MADRID	2 de janeiro de 2021

EQUIPAS

BARCELONA

1	GK	Marc-André ter Stegen
2	DF	Sergiño Dest
3	DF	Gerard Piqué
4	DF	Ronald Araújo
5	MF	Sergio Busquets
7	FW	Antoine Griezmann
8	MF	Miralem Pjanić
9	FW	Martin Braithwaite
10	FW	Lionel Messi
11	FW	Ousmane Dembélé
12	MF	Riqui Puig
13	GK	Neto

REAL MADRID

1	GK	Thibaut Courtois
2	DF	Dani Carvajal
3	DF	Éder Militão
4	DF	Sergio Ramos
5	DF	Raphaël Varane
6	DF	Nacho
7	FW	Eden Hazard
8	MF	Toni Kroos
9	FW	Karim Benzema
10	MF	Luka Modrić
11	FW	Marco Asensio

Figura 3.44: Interface da consulta de jogos agendados

Jogos
Ligas
Equipas
Mapa de jogos
Favoritos

utilizador1

Voltar

3

BARCELONA

87 minutos

REAL MADRID

2

ACONTECIMENTOS

1ª PARTE

10' (GOLO) Messi

15' (GOLO) Messi

23' (Amarelo) Modric

46' (Vermelho) Modric

2ª PARTE

62' (GOLO) Pedri

73' (Substituição) IN-Puig OUT-Pedri

46' (GOLO) Vinicius Jr.

56' (GOLO) Casemiro

ESTATÍSTICAS

20	REMATES	6
8	REMATES A BALIZA	3
68%	POSSE DE BOLA	32%
11	FALTAS	7
9	CANTOS	5
0	CARTÕES VERMELHOS	1
0	CARTÕES AMARELOS	0

EQUIPAS

Real Madrid


4-3-3

Barcelona


4-4-2

Figura 3.45: Interface da consulta de jogos a decorrer

44



Jogos
Ligas
Equipas
Mapa de jogos
Favoritos


Utilizador1

Manchester City










Jogos Terminados		Jogos Agendados		Equipa	
NOME ^	POS	IDADE	ALT	PS	
 Sergio Agüero	A	32	1,73 M	70 kgs	
 Nathan Aké	E	26	1,80 M	75 kgs	
 Adrián Bernabé	M	19			
 Scott Carson	G	35	1,90 M	76 kgs	
 João Pedro Cavaco Cancelo	E	26	1,82 M	74 kgs	
 Kevin De Bruyne	M	29	1,81 M	68 kgs	
 Gabriel Fernando de Jesus	A	24	1,75 M	73 kgs	
 Liam Delap	A	18			
 Rúben dos Santos Gato Alves Dias	E	23	1,87 M	70 kgs	

Figura 3.46: Interface da consulta da informação das equipas

WebGoal

Jogos

Ligas

Equipas

Mapa de Jogos

Favoritos













Utilizador1

Classificações

Melhores Marcadores

Época

2020-21

Clube	PD	V	E	D	GM	GS	DG	Pts	Últimos 5
1  Atlético Madrid	35	23	8	4	61	22	39	77	<div><div></div><div></div><div></div><div></div><div></div></div>
2  Real Madrid	35	22	9	4	60	26	34	75	<div><div></div><div></div><div></div><div></div><div></div></div>
3  Barcelona	35	23	6	6	80	33	47	75	<div><div></div><div></div><div></div><div></div><div></div></div>
4  Sevilla	35	22	5	8	51	29	22	71	<div><div></div><div></div><div></div><div></div><div></div></div>
5  Real Sociedad	35	15	11	9	53	35	18	56	<div><div></div><div></div><div></div><div></div><div></div></div>
6  Villarreal	35	13	13	9	53	42	11	52	<div><div></div><div></div><div></div><div></div><div></div></div>
7  Betis	34	14	9	11	43	46	-3	51	<div><div></div><div></div><div></div><div></div><div></div></div>
8  Celta de Vigo	35	12	11	12	50	53	-3	47	<div><div></div><div></div><div></div><div></div><div></div></div>
9  Ath. Bilbao	35	11	13	11	46	38	8	46	<div><div></div><div></div><div></div><div></div><div></div></div>
10  Granada	34	13	6	15	43	55	-12	45	<div><div></div><div></div><div></div><div></div><div></div></div>
11  Cádiz	35	11	10	14	31	50	-19	43	<div><div></div><div></div><div></div><div></div><div></div></div>
12  Osasuna	35	10	11	14	33	43	-10	41	<div><div></div><div></div><div></div><div></div><div></div></div>

La Liga

Premier League

Ligue 1


Serie A

Bundesliga


Liga NOS

Champions League

Figura 3.47: Interface da consulta das classificações de ligas



Jogos
Ligas
Equipas
Mapa de jogos
Favoritos


Utilizador1

Adicionar Favoritos


La Liga
Premier league
Ligue 1
Serie A
Bundesliga
Liga NOS
Champions League

Bayern Munich
RB Leipzig
Wolfsburg
Dortmund
Eintracht Frankfurt
Bayer Leverkusen
B. Monchengladbach
Union Berlin
Freiburg
Stuttgart
Hoffenheim
Mainz
Augsburg
Hertha Berlin
Werder Bremen
Arminia Bielefeld
FC Koln
Schalke


Guardar Alterações

Cancelar

Figura 3.48: Interface de alterar a lista de favoritos



Jogos
Ligas
Equipas
Mapa de jogos
Favoritos


utilizador1

Editar dados pessoais

Ativar Notificações

Nome de utilizador	utilizador1
E-Mail	utilizador1@hotmail.com
Password	••••• <input type="button" value="Q"/>

Guardar Alterações

Cancelar

Figura 3.49: Interface de alterar os dados pessoais

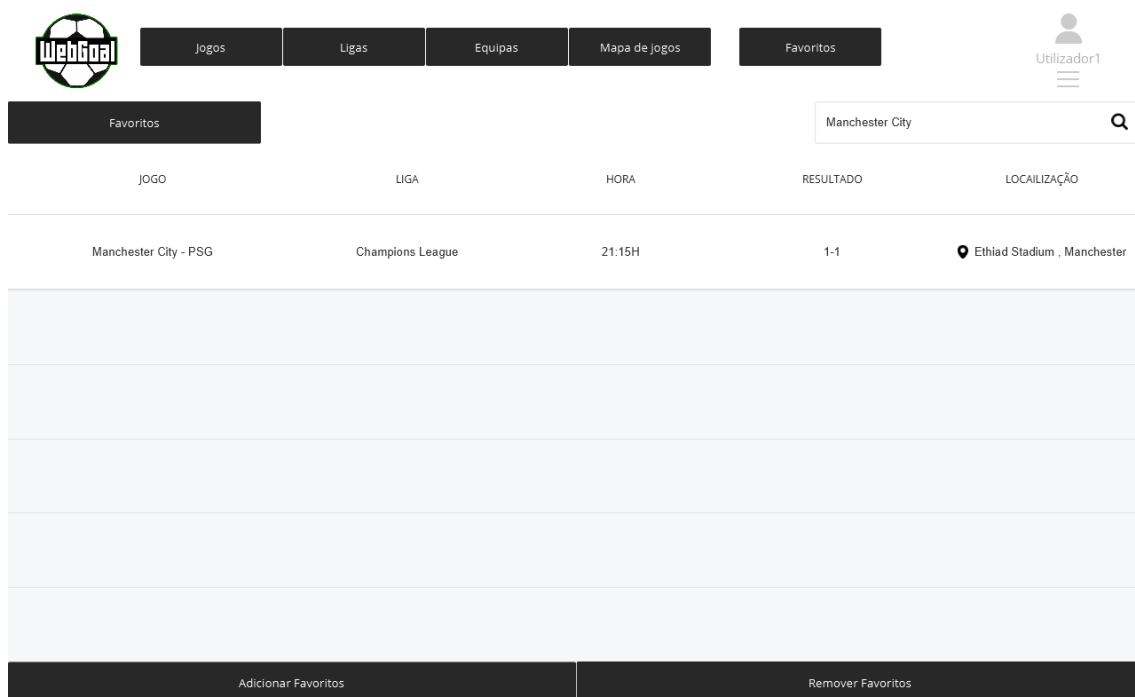


Figura 3.50: Interface de consulta de jogos das equipas favoritas

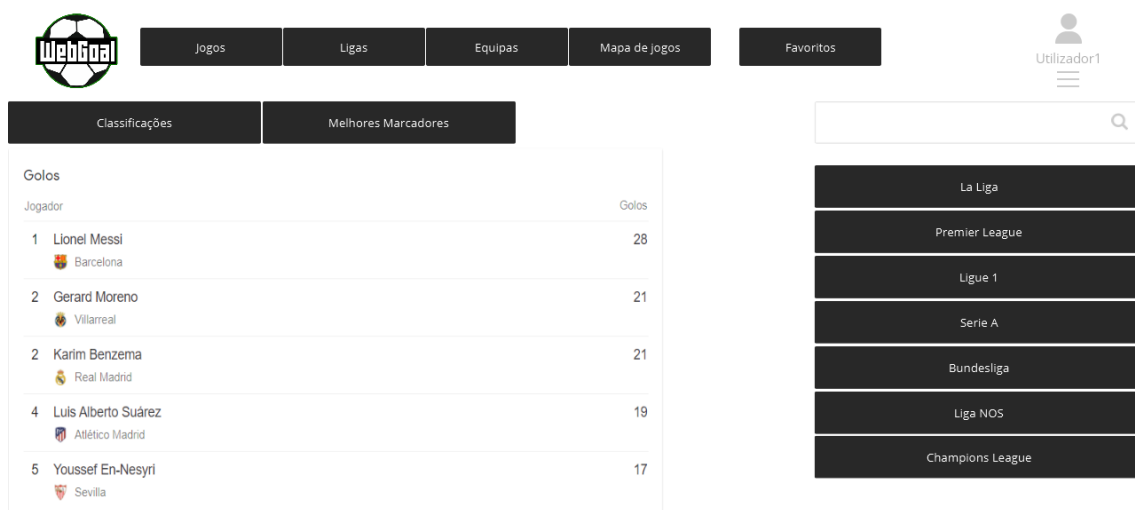


Figura 3.51: Interface da consulta dos melhores marcadores de uma liga

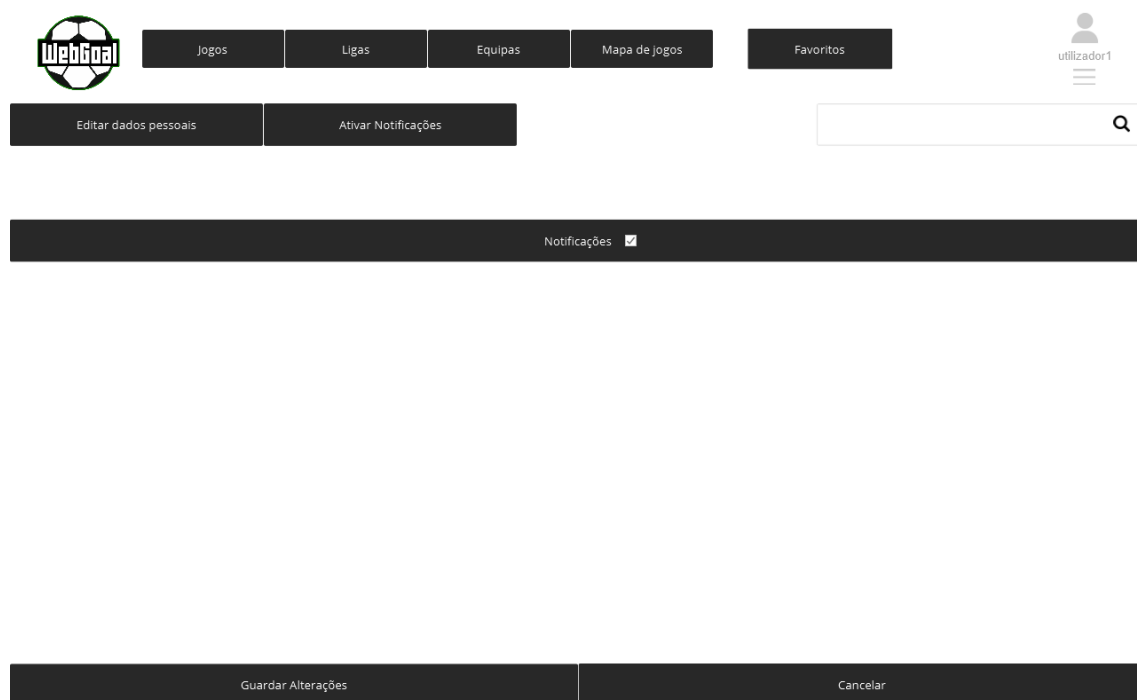


Figura 3.52: Interface para aceitar o envio de notificações

## 4 Fase de Construção

Finalmente temos a fase de Construção. Nesta iremos detalhar o desenvolvimento específico de cada funcionalidade idealizada nas fases anteriores. Assim sendo, podemos dividir esta em duas partes principais: primeiro, a criação da base de dados, bem como o seu povoamento com os dados necessários para o bom funcionamento do sistema e o alojamento desta num servidor, e segundo o desenvolvimento de um *front-end* acessível e *user-friendly*, sendo este capaz de representar e resolver todos os *use cases* por nós definidos anteriormente.

Sendo assim, a nossa base de terá de ser criada tendo em conta a informação presente na API por nós escolhida e que consiga responder às *queries* pretendidas.

De seguida iremos explicar como será desenvolvida a interface do Sistema, bem quais as ferramentas por nós utilizadas para a desenvolver. Iremos também abordar como funciona a implementação da mesma para cada *use case* definido anteriormente.

Finalmente iremos apresentar e analisar o nosso trabalho final, apresentado possíveis alternativas e soluções para problemas que não conseguimos resolver.

Uma vez que não conseguimos desenvolver a maioria do pretendido para esta fase do trabalho, decidimos que esta parte do relatório passaria a ser mais explicativa sobre as ideias e planos que nós tínhamos para a mesma, uma vez que isto é também um importante passo para a criação de Sistemas como este.



## 4.1 Base de Dados

Após alguma pesquisa, decidimos utilizar uma API chamada **football-data**. Nesta conseguimos obter informação de 12 ligas diferentes, tendo acesso às informações referentes aos jogos, às equipas, às classificações das mesmas dentro de cada liga e até ao símbolo associado a cada clube. Contudo, a versão grátis desta API apenas permite realizar 10 chamadas por minuto, o que limita a capacidade de realização de diferentes *queries* e/ou atualização da informação.

Posto isto, alteramos a nossa implementação da base de dados criada na segunda fase, uma vez que esta tinha demasiada informação que não era fornecida pela API.

Uma possível alternativa a esta implementação passaria por apenas utilizar a base de dados para guardar a informação referente aos utilizadores. Assim, sempre que fosse necessário informação para responder às *queries*, por exemplo sobre jogos de uma dada equipa, seria feito um novo *request* referente ao pretendido, de forma a retirar diretamente a informação da API. Mais uma vez, como a versão grátis da API tem um limite de 10 requests por minuto, esta implementação teria o problema de caso o utilizador fizesse mais *requests* do que o permitido este teria de esperar para realizar mais pedidos, sendo que a Base de Dados iria funcionar quase como uma *cache* a informação dada pela API. No entanto, seria provável que mesmo assim o limite de 10 chamadas fosse atingido constantemente, pelo que seria mesmo necessário investir num pacote com mais chamadas à API para fazer *deployment* do sistema real.

Mais uma vez, o nosso grupo não conseguiu realizar a parte da camada de dados, sendo assim iremos de seguida apresentar e discutir algumas das conclusões e ideias às quais chegamos enquanto tentávamos resolver o problema.

### 4.1.1 Alojamento da Base de Dados

Após a criação de um novo modelo lógico será necessário alojar a base de dados de forma a que esta possa ser acedida pela aplicação *web* que iremos desenvolver futuramente. Para isto temos 3 alternativas principais: a informação pode ser guardada localmente na máquina que está a correr a aplicação, ou pode ser guardada na *cloud* utilizando quer o *Microsoft Azure* ou o *SQL Server*. A principal diferença entre local ou na *cloud* é basicamente que, caso o utilizador queira usar o sistema com a base de dados local a informação terá que estar toda guardada na mesma máquina. Para além disto, este tipo de implementação impossibilita a utilidade de ter uma conta pessoal no sistema, uma vez que cada utilizador terá acesso à toda a informação que está guardada localmente, e não faria sentido um utilizador ter acesso à informação de outro.

Assim sendo, a solução mais lógica é então realizar o alojamento da Base de Dados diretamente na *Cloud*, onde, para além de ocupar menos espaço na máquina do utilizador, consegue também proteger melhor a informação nela presente.

### 4.1.2 Atualização da Informação da Base de Dados

Uma vez que o objetivo principal do trabalho é a criação de um sistema de monitorização de eventos em tempo (semi-)real é necessário que a informação apresentada seja atualizada regularmente, de forma a ser possível representar novos eventos ou informações novas sobre os eventos que se encontram ao vivo.

Uma abordagem possível passaria por utilizar os 10 *requests* por minuto disponíveis. Assim, através de *requests* espaçados ao longo do minuto poderíamos ter a informação que se encontra na API com um *delay* de no máximo 6 segundos (considerando que não existe *delay* entre a comunicação Evento-API, e que não existe latência de pedidos). Apesar da informação estar bastante atualizada, esta talvez não seja a solução ideal devido a uma possível sobrecarga da API. Para além disto, utilizar esta solução implica que toda a informação esteja permanentemente acessível na base de dados, uma vez que não será possível ao utilizador realizar *requests* à API, caso estes sejam necessários para algum *use case*, como iremos explicar mais à frente.

Por outro lado, e uma vez que não é completamente necessário apresentar a informação num tempo muito próximo ao real, tal como a solução anterior consegue, poderemos optar por simplesmente realizar um *request* em intervalos fixos de, por exemplo, 1 em 1 minuto. Desta forma, para além de termos a informação maioritariamente atualizada no sistema inteiro conseguimos também permitir ao utilizador realizar possíveis *requests*, para além de tornar todo o sistema menos "pesado".

## 4.2 Interface

De forma a conseguir apresentar o sistema e a informação nele presente aos utilizadores criamos uma interface utilizando o *Blazor*. Esta ferramenta permite-nos construir aplicações *web* de uma forma mais simples, utilizando *C#*. Para além disso, é bastante fácil realizar a ligação à base de dados, quer esta esteja alojada no *Azure* ou no *SQL Server*, o que permite juntar duas componentes principais do Sistema.

Uma vez que não conseguimos alojar a base de dados na *cloud* optamos por uma abordagem um pouco diferente, mas que já temos vindo a referir ao longo desta parte do relatório. Quando o utilizador pretender realizar um *query*, será feito um *request* específico à API, não havendo qualquer tipo de persistência de dados no sistema. Isto apenas foi utilizado como uma espécie de último recurso, uma vez que a não existência de dados permanentes no sistema faz com que, após o décimo pedido no sistema, não seja possível apresentar mais informação durante o minuto.

Assim, criamos um ficheiro *Classes.cs* com as estruturas necessárias para conseguirmos guardar a informação referente aos vários tipos de *requests* diferentes. Para além deste ficheiro, temos também dentro da diretoria *Shared* o *NavMenu.razor* e o *MainLayout.razor*, que serão utilizados para chamar as opções do menu lateral e o próprio menu lateral, respetivamente.

Dentro da diretoria *Pages* temos então as nossas respostas às *queries*. Cada uma tem um ficheiro *Razor* e um *C#*, sendo o primeiro responsável pela apresentação da informação e o segundo por efetuar o *request* necessário.

## 4.3 Funcionalidades Finalizadas/Pretendidas

Finalmente iremos falar sobre os *use cases* que conseguimos implementar e como estaríamos a pensar implementar aqueles que ficaram por fazer. Mais uma vez, a maioria das limitações quanto aos *use cases* que tínhamos definido anteriormente foram criadas por não conseguirmos ter uma base de dados funcional e devido à informação presente na API, o que nos obrigou a alterar alguns dos *use cases*.

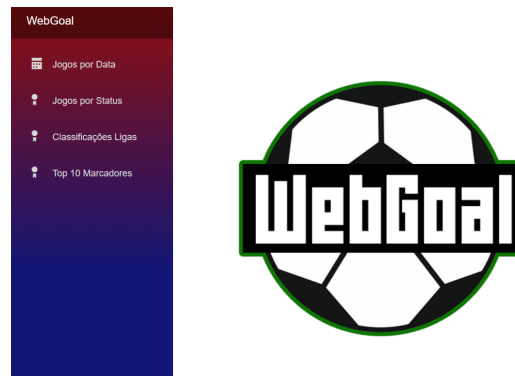


Figura 4.1: Interface da Página Inicial

### Utilizadores

Uma vez que não conseguimos realizar a ligação do sistema com uma Base de Dados não implementamos nenhum dos *use cases* relacionados com os utilizadores, quer sejam estes iniciar sessão, criar conta, ou qualquer um dos privilégios a que os utilizadores registados teriam acesso. Contudo, chegamos a algumas conclusões relativas a possíveis alternativas para as funcionalidades desejadas.

Por exemplo, de forma a facilitar a implementação, seria útil considerar o utilizador iria receber notificações de todos os Clubes que tem nos favoritos, em vez de decidir, numa lista à parte, os Clubes dos quais pretende receber notificações. Para além disto existem duas formas de guardar esta lista. A lista pode estar ou guardada no utilizador, como tínhamos imaginado inicialmente, ou então cada clube pode manter uma lista dos utilizadores que os marcaram como favoritos. Faz sentido a lista de favoritos de cada utilizador estar centralizada na sua informação, mas caso parte desta esteja diretamente no clube, será muito mais fácil descobrir quais os utilizadores que devem receber as notificações. Assim que existe alguma mudança num determinado Clube, o Sistema só terá de percorrer a lista de utilizadores que o tem nos favoritos (estando esta na "posse" do Clube) e enviar a notificação pretendida. Caso não fosse assim, então seria necessário percorrer todos os utilizador e verificar quais é que têm o Clube na lista de favoritos, e só aí é que se enviava a notificação.

Ainda sobre as notificações, uma vez que ao criar conta o utilizador teria de fornecer a sua conta de e-mail faria sentido as notificações serem enviadas diretamente para o e-mail pessoal

do utilizador. Também seria possível enviar estas através de *Push-Notifications* diretamente pelo *Browser*.

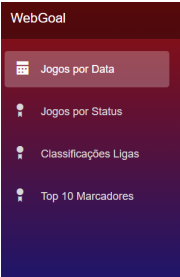
## Pesquisas

Considerando os *use cases* que tínhamos especificado na fase anterior do trabalho no que toca à pesquisa e apresentação da informação referente aos jogos, chegamos à conclusão que estes ainda poderiam ser alterados de forma a, não só diminuir o número total de *use-cases*, o que torna a interface em si mais simples, mas também de forma a tornar a codificação do próprio programa mais simples. Assim, a consulta de jogos agendados, ao vivo ou terminados passa a ser tratada como um único *use case*, a consulta por *status*. Nesta o utilizador insere o tipo de jogo que pretende analisar e o Sistema irá apresentar a informação da mesma forma, seja qual for o *status* do jogo pretendido.

A Consulta por equipa foi algo que não conseguimos implementar, uma vez que o *request* apresentado na documentação da API utilizada parecia não retornar a informação necessária para a conseguirmos resolver, ou então não retorna a informação de forma a que consigamos realizar as *queries* tendo em conta a informação recebida.

Com isto, conseguimos desenvolver uma interface para pesquisar jogos por data e por *status*, a consulta de classificação das ligas e a consulta dos melhores marcadores das ligas.

Assim sendo apresentamos em seguida como ficou a interface para os *use cases* mencionados em cima.



The screenshot displays the 'WebGoal' application interface. On the left is a dark sidebar with a white header 'WebGoal' and four menu items: 'Jogos por Data' (highlighted), 'Jogos por Status', 'Classificações Ligas', and 'Top 10 Marcadores'. The main content area has a white background. At the top, it says 'Escolha a Liga que pretende analisar:' followed by a dropdown menu showing 'Primeira Liga Portuguesa'. Below this is a label 'Data do Jogo que pretende observar:' followed by a date input field containing '07-05-2021' and a calendar icon. The main part of the interface is a table with the following data:

Hora	Status	Equipa1	Resultado	Equipa2
18:00:00	FINISHED	FC Famalicão	1 - 0	CD Santa Clara
20:15:00	FINISHED	Boavista FC	1 - 1	CD Tondela

Figura 4.2: Interface principal e Consulta de jogos por Data/Liga



WebGoal

Jogos por Data

Jogos por Status

Classificações Ligas

Top 10 Marcadores

Escolha a Liga que pretende analisar:

Primeira Liga Portuguesa

Escolha o estado do jogo que pretende analisar:

Terminado

Data	Hora	Equipa 1	Resultado	Equipa 2
2020 / 9 / 18	18:00:00	FC Famalicão	1 : 5	Sport Lisboa e Benfica
2020 / 9 / 18	20:15:00	Vitória SC	0 : 1	Os Belenenses Futebol
2020 / 9 / 19	15:00:00	CD Nacional	3 : 3	Boavista FC

Figura 4.3: Interface de consulta de jogos por Status e Liga



WebGoal

Jogos por Data

Jogos por Status

Classificações Ligas

Top 10 Marcadores

Escolha a Liga que pretende analisar:

Primeira Liga Portuguesa

Posição	Clube	Nº. de Jogos	Vitórias	Empates	Derrotas	Golos Marcados	Golos Sofridos	Pontos	Últimos 5
1	 Sporting Clube de Portugal	34	26	7	1	63	19	85	W,L,W,W,W
2	 FC Porto	34	24	8	2	74	29	80	W,W,W,D,W

Figura 4.4: Interface de Consulta da Classificação das Ligas



WebGoal

Jogos por Data

Jogos por Status

Classificações Ligas

Top 10 Marcadores

Escolha a Liga que pretende analisar:

Primeira Liga Portuguesa

Nome	Clube	Posição	Golos Marcados
Pedro Gonçalves	Sporting Clube de Portugal	Midfielder	24
Haris Seferović	Sport Lisboa e Benfica	Attacker	23
Mehdi Taremi	FC Porto	Attacker	17
Mario González	CD Tondela	Attacker	15

Figura 4.5: Interface de Consulta dos melhores marcadores das Ligas

## Consultar eventos no Mapa

Acabamos por não conseguir desenvolver este *use case* mais por falta de tempo. Contudo, a própria API não possuía a informação referente à localização dos jogos nem dos estádios, o que nos iria dificultar bastante a tarefa. Caso tivéssemos conseguido organizar melhor o nosso tempo poderíamos ter criado uma *page* onde iríamos utilizar a ferramenta fornecida pela API *Bing Maps* para representar o mapa, e utilizando localizações inseridas por nós, representar a localização de alguns dos possíveis lugares onde os jogos poderiam decorrer, mesmo que não fosse possível associar os jogos ao locais em específico.

## 5 Conclusão

Com este Sistema queríamos proporcionar aos utilizadores um método fácil e acessível de monitorizar eventos futebolísticos de várias Ligas e Clubes em tempo real. Apesar de conseguir apresentar alguma informação interessante, ainda existem vários pontos que não foram tratados totalmente, para além das funcionalidades que não conseguimos implementar. Por exemplo, uma vez que a ideia principal do Sistema seria apresentar a informação de um jogo em específico, podemos considerar que aquilo que temos agora é mais semelhante a um método de procura, e não de consulta. Algo interessante que poderia ser implementado seria a possibilidade de clicar nos jogos demonstrados nos *use cases* concluídos, apresentando a informação detalhada dos mesmos. Esta era inicialmente a nossa ideia principal, mas com o desenvolver do trabalho acabamos por optar por algo um pouco diferente.

Para além disto, sentimos que o trabalho poderia ter sido bastante mais desenvolvido caso tivéssemos mais tempo disponível. Esta última fase coincidiu com vários trabalhos de várias cadeiras do grupo, o que não facilitou a nossa coordenação e organização no que toca às datas e objetivos definidos.

Como trabalho futuro, para além de implementar os *use cases* que não conseguimos para esta fase, e a criação da base de Dados, temos também a análise mais propriamente dita dos jogos, tal como referimos anteriormente. Uma vez que a API também tem informação relativa às *odds*, talvez também fosse interessante implementar funcionalidades que utilizassem estes dados.

Finalmente, apesar de nunca termos desenvolvido algo na área de *web-development*, sentimos que ficámos mais à vontade com a sintaxe utilizada e que conseguimos desenvolver as nossas competências no que toca a gestão de projetos de média escala, especialmente, no que toca ao planeamento dos mesmos.