



Universidade do Minho
Escola de Engenharia
Mestrado Integrado em Engenharia Informática

Unidade Curricular de Laboratórios de Informática IV

Ano Lectivo de 2020/2021

LightsOut – Sistema de Monitorização de Eventos de F1

António Santos (A83700)

Carlos Preto (A89587)

Jorge Vieira (A84240)

Simão Monteiro (A85489)

junho 2021

Data de Recepção	
Responsável	
Avaliação	
Observações	

LightsOut – Sistema de Monitorização de Eventos de F1

António Santos (A83700)

Carlos Preto (A89587)

Jorge Vieira (A84240)

Simão Monteiro (A85489)

junho 2021

Resumo

No âmbito da unidade curricular de Laboratórios de Informática IV, foi proposto aos alunos que desenvolvessem um Sistema de Monitorização de Eventos, podendo este ser referente a qualquer evento pretendido.

O projeto tem como objetivo acompanhar e reportar em tempo (próximo de) real um conjunto de eventos previamente definidos, fornecendo a sua caracterização, localização e explicação, permitindo também configurar as diversas fontes de informação. Também se pretende que seja possível visualizar num conjunto de painéis os eventos e a sua caracterização, incorporando mecanismos de alerta configuráveis que avisem dos utilizadores.

Ficou definido que o projeto seria dividido em 3 etapas distintas, sendo estas a Fundamentação, Especificação e Construção. Na primeira fase, denominada de Introdução, será abordada a etapa de Fundamentação, que consistiu em identificar e caracterizar de forma geral a aplicação a desenvolver, bem como fundamentar o seu desenvolvimento e justificar em termos de modelo de negócio o projeto. Também foi necessário definir um conjunto de medidas de sucesso e desenvolver um diagrama de desenvolvimento do projeto, de maneira a facilitar a organização do grupo no desenvolvimento do projeto. Na segunda fase será abordada a etapa de Especificação. Nesta foi feito o levantamento e análise de requisitos, a especificação em UML do sistema a desenvolver, onde serão definidos o diagrama de domínio, os diagramas de sequência, o diagrama de Use Case, o diagrama de Classes, a máquina de estado e o modelo lógico. Também será apresentado o dicionário de dados da base de dados e as respetivas *mockups* do sistema. Por fim, na terceira e última fase, denominada de Construção, serão referidas quais as ferramentas utilizadas, a estrutura do projeto desenvolvido, acompanhada de exemplos visuais, e as considerações finais associadas ao desenvolvimento do projeto.

No final do relatório será apresentada uma breve conclusão/reflexão do trabalho realizado, e também será referido o trabalho que poderá vir a ser realizado no futuro de maneira a consolidar ainda mais o projeto desenvolvido.

Área de Aplicação: Desenvolvimento de um Sistema de Monitorização de Eventos

Palavras-Chave: *Microsoft SQL Server, Microsoft Word, Microsoft Excel, Microsoft PowerPoint, Diagramas de Classe, Diagramas de Sequência, Diagramas de Use Case, Diagrama de Gantt, Monitorização de Eventos, Interface Visual.*

Índice

Resumo	iii
Índice	iv
Índice de Figuras	vi
1. Introdução	1
1.1. Contextualização	1
1.2. Motivação e Objetivos	2
1.3. Justificação do Sistema	2
1.4. Utilidade do Sistema	3
1.5. Estabelecimento da Identidade do Projeto	3
1.6. Identificação dos Recursos necessários	4
1.7. Maqueta do Sistema	5
1.8. Definição de Medidas de Sucesso	6
1.9. Plano de desenvolvimento	7
2. Levantamento e Análise de Requisitos	8
2.1. Base de Dados	8
2.2. Dashboard's	9
2.3. Notificações	11
2.4. API	12
2.5. Autenticação	12
3. Modelação do Sistema	13
3.1. Modelo de Domínio	13
3.2. Use Cases	14
3.2.1 Consultar Mapa de Provas	15
3.2.2 Consultar Prova	16
3.2.3 Notificar Prova	17
3.2.4 Consultar Classificações Gerais	18
3.2.5 Consultar Equipas e Pilotos	21
3.3. Diagrama de Classes	22
3.4. Máquina de Estado	24
4. Base de Dados Relacional	25
4.1. Esquema Lógico	25
4.2. Dicionário de Dados e Estimativa de Espaço	26

5. Definição do sistema de Interface (<i>Mockups</i>)	29
6. Implementação	33
6.1. Ferramentas Utilizadas	33
6.2. Apresentação do Sistema	34
6.2.1 Consultar Mapa de Provas	34
6.2.2 Consultar Prova	35
6.2.3 Login	36
6.2.4 Notificar Prova	36
6.2.5 Consultar Classificações Gerais	39
6.2.6 Consultar Equipas e Pilotos	40
6.2.7 Página Inicial	41
6.3. Validação do Sistema	42
6.4. Considerações Finais	43
7. Conclusões e Trabalho Futuro	45
Referências	46
Lista de Siglas e Acrónimos	47

Índice de Figuras

Figura 1 – Maqueta do Sistema	5
Figura 2 - Diagrama de Gantt	7
Figura 3 - Modelo de Domínio	13
Figura 4 - Diagrama Use Cases	14
Figura 5 - Especificação Mapa de Provas	15
Figura 6 - Diagrama Sequência Mapa Provas	16
Figura 7 - Especificação Consultar Prova	16
Figura 8 - Diagrama Sequência Consultar Prova	17
Figura 9 - Especificação Notificar Prova	17
Figura 10 - Diagrama Sequência Notificar Prova	18
Figura 11 - Especificação Classificações Gerais	19
Figura 12 - Diagrama Sequência Classificação Pilotos	20
Figura 13 - Diagrama Sequência Classificação Equipas	20
Figura 14 - Especificação Equipas e Pilotos	21
Figura 15 - Diagrama Sequência Equipas e Pilotos	22
Figura 16: Diagrama de Classes DataAccessLibrary	23
Figura 17: Diagrama Classes Backend	23
Figura 18: Diagrama Classes Frontend	24
Figura 19 - Máquina de Estado	24
Figura 20 - Modelo Lógico	25
Figura 21 - Dicionário de Dados	28
Figura 22 - Mockup Página Inicial	29
Figura 23 - Mockup Mapa de Provas	30
Figura 24 - Mockup Prova	30
Figura 25 - Mockup Classificações Gerais	31
Figura 26 - Mockup Equipas e Pilotos	31
Figura 27 - Mockup Login	32
Figura 28: Distinção das Provas	34
Figura 29: Exemplo Resultado Prova	35
Figura 30: Login	36
Figura 31: Criar Notificação	37
Figura 32: Autenticação Necessária	37

Figura 33: Notificação já Existe	38
Figura 34: Notificação Adicionada	38
Figura 35: Notificação Começo de Prova	39
Figura 36: Classificações Gerais	40
Figura 37: Equipas e Pilotos	40
Figura 38: Página Inicial Normal	41
Figura 39: Página Inicial Autenticado	41

1. Introdução

1.1. Contextualização

O desporto está bastante presente na vida das pessoas, sendo que é através da prática deste que se consegue obter um estilo de vida mais ativo e saudável. Com o surgimento da pandemia e os sucessivos confinamentos, a prática de desporto passou para um plano secundário, ou seja, ao invés de se praticar desporto, passou-se a observar a prática de desporto em televisões, telemóveis e outros aparelhos.

Esta mudança de contexto levou ao aumento do número de adeptos dos diferentes desportos por toda a parte. Com este aumento, notou-se o surgimento de diferentes aplicações que monitorizam os mais diversos desportos, de maneira a informar as pessoas dos eventos desportivos mais atuais e que lhes possam ser de interesse.

Um dos eventos desportivos que despertou um grande interesse do público nos últimos tempos foi o desporto motorizado, sendo que destes, o que mais se destacou em Portugal foi a F1. No ano de 2020 verificou-se o regresso do Campeonato do Mundo de F1 a Portugal, tendo este tido lugar no Autódromo Internacional do Algarve, em Portimão. Com este regresso, o foco da imprensa desportiva portuguesa passou a ser a F1, que levou à introdução deste desporto num maior número de pessoas.

Neste contexto, a *LightsOut* pretende afirmar-se como fonte informativa dos eventos de F1, oferecendo as informações mais recentes e importantes ao seu público alvo, o mais detalhadamente possível.

1.2. Motivação e Objetivos

Segundo a página oficial da F1, no ano de 2020, o número de espectadores associado a cada Grande Prémio deste desporto rondou os 87.4 milhões. Uma vez que a temporada teve 17 Grandes Prémios, pode-se concluir que houve um total de 1,49 mil milhões de espectadores, o que levou o grupo a considerar o tema de F1 de bastante interesse.

Além das nossas preferências, também se teve em consideração os meios de obtenção de informação e de que forma a informação disponibilizada online poderia beneficiar-nos. Após algumas pesquisas, encontraram-se diferentes sites capazes de fornecer a informação necessária para o desenvolvimento do projeto.

Desse modo, a *LightsOut* pretende fornecer uma plataforma inovadora, atrativa e informativa, que cativa os apoiantes deste desporto motorizado. Para além das funcionalidades habituais, também se pretende que a plataforma apresente funcionalidades únicas que atraiam todo o tipo de utilizadores, nunca deixando de ter em mente que esta seja de fácil compreensão e uso, tentando sempre fornecer dados fiáveis e atualizados o mais rápido possível, pois só dessa maneira será possível manter os utilizadores da nossa plataforma.

1.3. Justificação do Sistema

Existem várias plataformas informativas que colocam ao dispor do utilizador um leque cheio de informações acerca dos diferentes eventos desportivos, porém o maior foco destas plataformas centra-se em desportos como futebol e basquetebol.

Destas plataformas, a que mais se destaca é a aplicação *FlashScore*. Esta plataforma, apesar de também abordar o tópico da F1, não apresenta o devido detalhe e informação que seria pretendido, em comparação com os desportos reis referidos acima. Nota-se também, várias vezes, a desatualização de resultados, sendo que essa desatualização de resultados leva a que muitos utilizadores fiquem insatisfeitos e procurem outra plataforma.

Também é importante salientar que a falta de concorrência nesta área poderá fazer com que a nossa plataforma se destaque e seja preferida pelos utilizadores. O próprio site da F1 será possivelmente o único concorrente do projeto, porém, é necessário ao utilizador subscrever-se para obter todo o tipo de informação.

Tendo em conta estes fatores, o grupo decidiu que o projeto é viável para desenvolvimento.

1.4. Utilidade do Sistema

O quotidiano de uma pessoa não lhe permite acompanhar, em tempo real, todos os eventos que lhe despertam interesse, sendo por isso necessário arranjar uma solução que permita a monitorização dessa informação, de uma forma simples e concisa.

Apesar da maioria das pessoas ter acesso à internet, a informação pretendida encontra-se muitas vezes dispersa, sendo complicado obter uma fonte fidedigna e atualizada.

Tendo estes fatores em consideração, a *LightsOut* pretende disponibilizar uma ferramenta de monitorização de eventos de F1 que reúna a informação necessária, de maneira a que um adepto de F1 se consiga manter a par das últimas novidades.

Como foi dito anteriormente, em algumas aplicações são utilizados métodos de subscrição, permitindo aos utilizadores que utilizem esse método ter acesso a informação restrita. O nosso projeto será totalmente independente desses métodos, podendo qualquer pessoa ter acesso à informação que bem pretenda, sendo que por essa mesma razão, o grupo espera atrair mais utilizadores.

As funcionalidades implementadas no sistema irão permitir ao utilizador navegar e obter informação com o menor grau de dificuldade possível, de uma maneira organizada e apelativa, podendo sempre que pretender filtrar apenas a informação que lhe interessa.

1.5. Estabelecimento da Identidade do Projeto

O sistema *LightsOut* surgiu da proposta de desenvolvimento de um Sistema de Monitorização de Eventos, sendo que o nosso projeto será enquadrado na categoria Desporto.

Uma frase muito reconhecida pelos fãs de F1 é proferida pelo atual comentador deste desporto na *SkySports*, o senhor David Croft, onde no instante em que as luzes se apagam e se dá o início da prova este diz “It’s lights out, and away we go!”. Dito isto, o grupo achou que *LightsOut* seria um nome apelativo e adequado para o nosso sistema.

O grupo é identificado pelo número 2 e é constituído por quatro elementos, todos estudantes de Engenharia Informática na Universidade do Minho, sendo eles o Carlos Preto, o Simão Monteiro, o Jorge Vieira e o António Santos.

O projeto terá de ser desenvolvido até ao dia 7 de junho de 2021, e no final, o grupo espera ter um sistema de monitorização único, permitindo acompanhar os últimos resultados da F1, bem como a sua evolução e descrição.

1.6. Identificação dos Recursos necessários

Apesar de ainda nos estarmos numa fase muito recente do projeto, o grupo já tem uma ideia geral dos recursos que serão necessários para atingir as funcionalidades desejadas.

Antes de se procurar por informações e notícias que pudessem contribuir para o sucesso do projeto, foi necessário definir diferentes funções a alguns membros do grupo. Ficou definido que iria haver um coordenador e um analista de dados, responsáveis por controlar a gestão e progresso do projeto, sendo importante referir que todos os elementos do grupo irão contribuir para a programação do projeto.

Posteriormente, será necessário arranjar maneira de obter os dados mais recentes relacionados com a F1. Para isso, teremos ao nosso dispor três *API's*:

- ✓ *Ergast F1*;
- ✓ *API-Formula-1*;
- ✓ *Formula One API*;

Em cada uma das *API's* irá estar disponível informação acerca das corridas, horários, classificações, pilotos e muito mais relacionado com a F1, o que permitirá enriquecer o conteúdo disponível na plataforma. Além das *API's* relacionadas com a F1, também será usada a *Bing Maps Platform API*, para que assim se obtenha um mapa para representar as diferentes provas.

De maneira a se conseguir controlar o estado em que se encontra o projeto, será necessário realizar reuniões semanais, onde cada elemento irá apresentar o trabalho realizado, podendo também sugerir novas funcionalidades para o projeto. Nestas reuniões serão feitas ou não correções a esse mesmo, sendo que será sempre importante que cada elemento esteja presente e cumpra a sua parte, pois só assim se terá um desenvolvimento correto do sistema.

Por fim, e se ainda houver tempo, o grupo gostaria de ter a opinião de algumas pessoas acerca do nosso projeto, onde estas pudessem apresentar o seu *feedback*. Posteriormente, o grupo, com base nas críticas obtidas, iria tentar melhorar o projeto.

1.7. Maqueta do Sistema

Tendo por base exemplos de sistemas de monitorização, conclui-se que o nosso sistema irá ser constituído pelas componentes *Front-End* e *Back-End*.

Comecemos o explicar o que é cada uma delas. A *Front-End* é referente à etapa inicial de um processo, ou seja, é a interface com a qual o utilizador irá interagir. Esta tratará de processar a interação do utilizador e adequá-la a uma especificação em que o *Back-End* possa utilizar. O *Back-End* tratará das regras de negócio do nosso sistema, isto é, será o servidor que possibilita o funcionamento do nosso sistema e onde se faz a gestão das funcionalidades internas do sistema.

A *Front-End* irá consistir num *website*, que irá colocar ao dispor do utilizador diferentes tipos de informações, associados a diferentes vistas, relacionados com os tópicos de F1 que estes pretenderem. De maneira a ter um funcionamento correto do sistema, o *Front-End* e *Back-End* terão de estar a funcionar simultaneamente e corretamente, para assim apresentar dados corretos e atualizados.

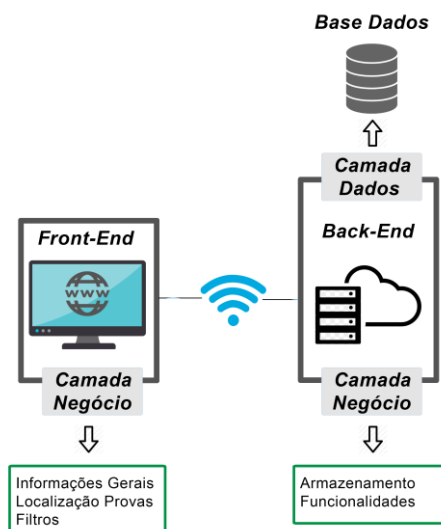


Figura 1 – Maqueta do Sistema

1.8. Definição de Medidas de Sucesso

Numa fase inicial do projeto, é necessário traçar um conjunto de metas a atingir, de maneira a que posteriormente se possa averiguar se o projeto teve êxito. Estas metas terão por base não só comparações com produtos concorrentes, como também com a satisfação obtida pelo grupo no final do projeto.

A primeira meta a atingir é ter um projeto 100% funcional, ou seja, que cumpra todos os requisitos propostos e que esteja disponível para qualquer pessoa. Só após tal ser atingido é que se podem definir as restantes metas.

Posteriormente, pretende-se criar um sistema com uma interface visualmente atrativa, rápida e de fácil compreensão/uso. Uma vez que os gostos variam de pessoa para pessoa, sabe-se que poderá haver algum utilizador que não concorde com as nossas abordagens, de tal modo que o nosso objetivo será agradar o máximo de pessoas possível.

Porém, de maneira a ter a opinião das pessoas em relação ao nosso projeto, será necessário arranjar maneira de as atrair. Uma vez que estamos em tempo de pandemia, a divulgação do projeto terá de ser feita maioritariamente nas redes sociais, onde poderá ser publicada publicidade ao projeto, ou até mesmo cada indivíduo do grupo convidar amigos e conhecidos a utilizar o nosso sistema.

Por fim e não menos importante, espera-se que o projeto seja um bom concorrente aos seus rivais, e que se diferencie destes de maneira positiva.

Caso estas metas sejam todas atingidas, pode-se concluir que o projeto teve êxito.

1.9. Plano de desenvolvimento

É importante delinear um plano de desenvolvimento do projeto, pois só assim será possível ao grupo organizar-se e ter uma ideia geral do estado de desenvolvimento do sistema.

Este plano consistirá na definição do tempo de execução das diferentes tarefas, de maneira a saber se alguma fase do projeto cumpriu os prazos delimitados, ou se o grupo terá de avançar o ritmo de desenvolvimento do projeto.

O projeto será dividido em 3 fases, tendo cada uma delas uma data limite de conclusão. A primeira fase consiste na “Fundamentação” do projeto, onde é necessário escolher o tema do projeto, fundamentar o desenvolvimento deste em termos de modelo de negócio, definir as medidas de sucesso, entre outras.

A próxima fase será a de “Especificação”, onde se fará uma análise dos requisitos, especificação geral do *software* a desenvolver, construir os diagramas de *Use Case*, Sequência e Classe.

Numa terceira fase, ter-se-á a “Construção” do projeto. Nesta fase, será feita uma apresentação e caracterização geral da arquitetura do sistema, uma descrição dos diversos módulos e ferramentas utilizadas, bem como a validação do software de desenvolvimento.

De maneira a ter um bom ritmo de trabalho, ficou definido que a cada etapa do projeto irão estar associadas, no mínimo, 2 pessoas, especialmente na fase de desenvolvimento da *BackEnd* e *FrontEnd*, para que assim se possam desenvolver várias etapas em simultâneo.

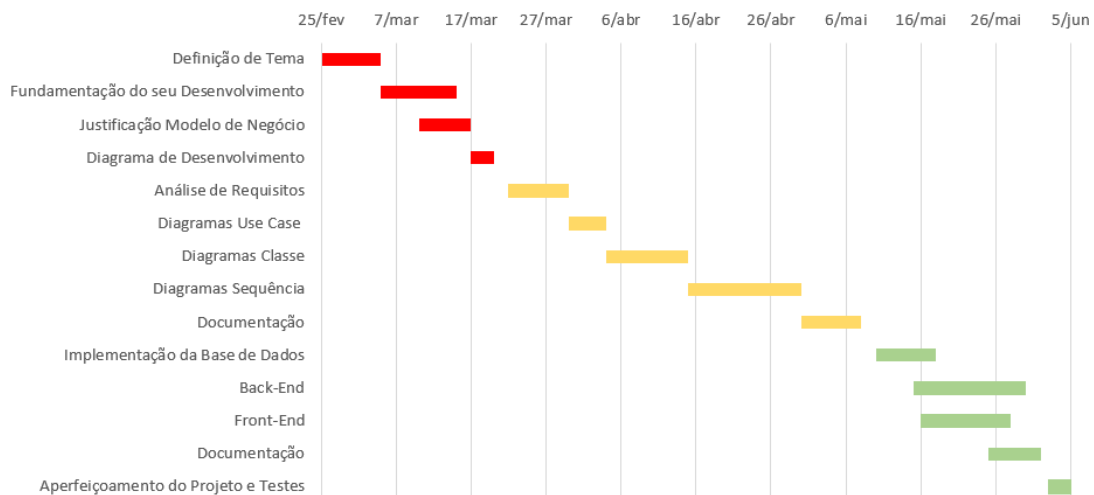


Figura 2 - Diagrama de Gantt

2. Levantamento e Análise de Requisitos

De maneira a ter uma ideia das funcionalidades do projeto, o grupo decidiu reunir-se com alguns adeptos de F1, sendo esses o Carlos Preto, o Simão Monteiro, o Jorge Vieira e o António Santos.

Foi inquirido a cada um destes adeptos quais seriam as diversas fontes de informação que gostariam de ver implementadas num sistema de monitorização de eventos de F1, de tal forma que fosse apresentada a informação fundamental para que qualquer adepto deste desporto monitorizado conseguisse visualizar a informação necessária.

Após alguns dias de discussão e análise dos diferentes requisitos propostos, o grupo conseguiu chegar a um consenso em relação aos requisitos finais do Sistema.

2.1. Base de Dados

Requisito Funcional

- ✓ O Sistema deverá ser capaz de suportar uma base de dados, à qual deverá aceder para que seja possível mostrar as diferentes informações pretendidas ao Utilizador.
 - De maneira a manter as informações atualizadas, foi necessário definir com que frequência teria de se atualizar a base de dados.
 - Após consulta do calendário de F1, observou-se que todas as Qualificações decorrem ao Sábado à tarde e terminam, no máximo, às 16h. Todas as corridas decorrem ao Domingo à tarde e terminam, no máximo, às 17h.
 - Desta forma, convém atualizar a base de dados todos os dias, de maneira a que caso se aceda ao sistema num dia da semana, a base de dados já esteja atualizada com a informação mais recente.

2.2. Dashboard's

O Carlos Preto referiu que gostava que lhe fosse apresentado um mapa com as provas que ocorreram, que estão a ocorrer e que irão ocorrer. Acrescentou também que gostaria de ser ele mesmo a definir quais os intervalos temporais pretendidos.

Requisito Utilizador

1. Sistema deverá apresentar uma *dashboard* com um mapa das provas a ocorrer num determinado intervalo de tempo.

Requisitos Sistema

- i. O intervalo de tempo predefinido será o da data da consulta do site - **Requisito Funcional.**
- ii. Deverá ser possível navegar pelo mapa, de maneira a poder visitar diferentes áreas do planeta - **Requisito Funcional.**
- iii. O Sistema deverá fornecer a possibilidade de incrementar e decrementar o intervalo de tempo pretendido - **Requisito Funcional.**
- iv. As provas que já ocorreram irão aparecer com cor verde, as provas que estão a ocorrer irão aparecer com cor vermelha, e as provas que irão ocorrer irão aparecer com cor preta - **Requisito Funcional.**

O Simão Monteiro adicionou também que gostaria de, para cada prova terminada que aparece no mapa, ver qual foi a Qualificação e a Classificação Final, onde também se apresentariam informações associadas a cada uma delas.

Requisito Utilizador

2. O Sistema deverá apresentar uma *dashboard* com a Qualificação e Classificação final de uma dada prova.

Requisitos Sistema

- i. Na dashboard poderá ser visualizado, para cada piloto:
 - Posição Final;
 - Veículo;
 - Nome;
 - Tempo de Prova;
 - Melhor Tempo de Qualificação;
 - Posição Inicial;
 - Pontos obtidos;

O António Santos abordou o grupo acerca da possibilidade de apresentar, para uma dada época, qual a Classificação Geral dos Pilotos e das Equipas, até uma determinada prova, referindo também que gostaria que fosse ele a definir qual a época e prova pretendida, de maneira a poder ver a evolução temporal de cada uma das Classificações.

Requisito Utilizador

3. O Sistema deverá apresentar uma dashboard com a Classificação Geral dos Pilotos e das Equipas, numa determinada época, até uma determinada prova.

Requisitos Sistema

- i. A época predefinida deverá ser a época atual - **Requisito Funcional**.
- ii. A prova predefinida deverá ser a primeira prova - **Requisito Funcional**.
- iii. O Sistema deverá oferecer a possibilidade de incrementar e decrementar quer a época pretendida, quer a prova pretendida - **Requisito Funcional**.
- iv. Na Classificação Geral, para cada Piloto, poderá ser consultada:
 - Posição no Campeonato;
 - Nome;
 - Total de Pontos obtidos;
- v. Na Classificação Geral, para cada Equipa, poderá ser consultada:
 - Posição no Campeonato;
 - Nome;
 - Total de Pontos obtidos;

De seguida, o Jorge Vieira sugeriu uma ideia bastante atrativa. Este adepto referiu que acompanha a F1 há bastante tempo, e que algumas das sua equipas favoritas já deixaram de existir. Por isso, admitiu que gostaria de poder consultar, para uma dada época, quais as equipas pertencentes ao Campeonato do Mundo de Construtores.

Requisito Utilizador

4. O Sistema deverá apresentar uma dashboard com as Equipas pertencentes ao Campeonato do Mundo de Construtores, numa determinada época.

Requisitos Sistema

- i. A época predefinida deverá ser a atual - **Requisito Funcional..**
- ii. O Sistema deverá oferecer a possibilidade de incrementar e decrementar a época pretendida - **Requisito Funcional..**
- iii. Para cada Equipa, poderá ser consultado:
 - Nome;
 - Veículo/Nacionalidade;
 - Piloto Principal;
 - Piloto Secundário;

Por fim, ambos os adeptos abordados referiram que, de modo a se ter uma projeto mais apelativo, o grupo deveria ter uma página principal com um breve resumo do nosso sistema de monitorização.

Requisito Utilizador

5. O Sistema deverá apresentar uma página inicial.

Requisitos Sistema

- i. A página inicial deverá ter pouco texto, limitando-se ao essencial – **Requisito Funcional**
- ii. Poderá ser apresentado, como fundo, um vídeo ou uma imagem ilustrativa da F1 – **Requisito Funcional.**

2.3. Notificações

Requisito Utilizador

6. O Sistema deverá ser capaz de suportar mecanismos de alerta associados ao início de uma determinada prova.

Requisitos Sistema.

- i. Utilizador deverá estar autenticado – **Requisito Funcional.**
- ii. Utilizador poderá selecionar uma prova que ainda não tenha ocorrido e selecionar a opção de “Notificar Prova” - **Requisito Funcional.**
- iii. A notificação pretendida ficará associada ao perfil do utilizador, para que assim lhe possa chegar a notificação em qualquer *browser* – **Requisito Funcional.**
- iv. Quando a prova pretendida começar, deverá aparecer uma mensagem no ecrã a informar acerca do evento - **Requisito Funcional.**

2.4. API

Requisito Funcional

- ✓ O Sistema deverá utilizar uma API, de modo a poder obter informação necessária povoar a base de dados.
 - i. A API selecionada foi a *Ergast F1*.
 - ii. Esta deverá providenciar as seguintes informações, para determinadas datas:
 - Pilotos;
 - Equipas;
 - Provas;
 - Resultados;
 - Qualificações;

2.5. Autenticação

Requisito Utilizador

- ✓ O Sistema deverá implementar um mecanismo de autenticação, sendo este apenas necessário caso um dado Utilizador pretenda receber notificações.

Requisitos Sistema

- i. O Utilizador terá a opção de se autenticar no site - **Requisito Funcional**.
- ii. Para cada uma das opções escolhidas, será necessário que o Utilizador introduza o seu email e a sua *password* - **Requisito Funcional**.
- iii. O Sistema deverá guardar as informações referentes a cada Utilizador na sua base de dados, não podendo haver dados repetidos - **Requisito Funcional**.

3. Modelação do Sistema

Uma vez definidos os requisitos, o grupo avançou para a especificação em *UML* do sistema de *software* a desenvolver. Para tal, elaborou-se alguns diagramas que irão facilitar a compreensão da estrutura e contexto do sistema.

Após alguma ponderação, o grupo optou por desenvolver o Modelo de Domínio, o Diagrama de *Use Cases*, bem como a sua especificação, os Diagramas de Sequência associados a cada um dos *Use Cases* e a máquina de estado do sistema.

3.1. Modelo de Domínio

O nosso modelo de domínio corresponde à conceção inicial das classes candidatas do nosso sistema, ou seja, inclui as diferentes classes relacionadas com o tema da F1, bem como qual a relação entre deles. Este modelo contribuirá para que, numa fase posterior, se possa definir o diagrama de classes do sistema, onde algumas das classes definidas no modelo de domínio serão as finais, e as restantes passarão a ser atributos.

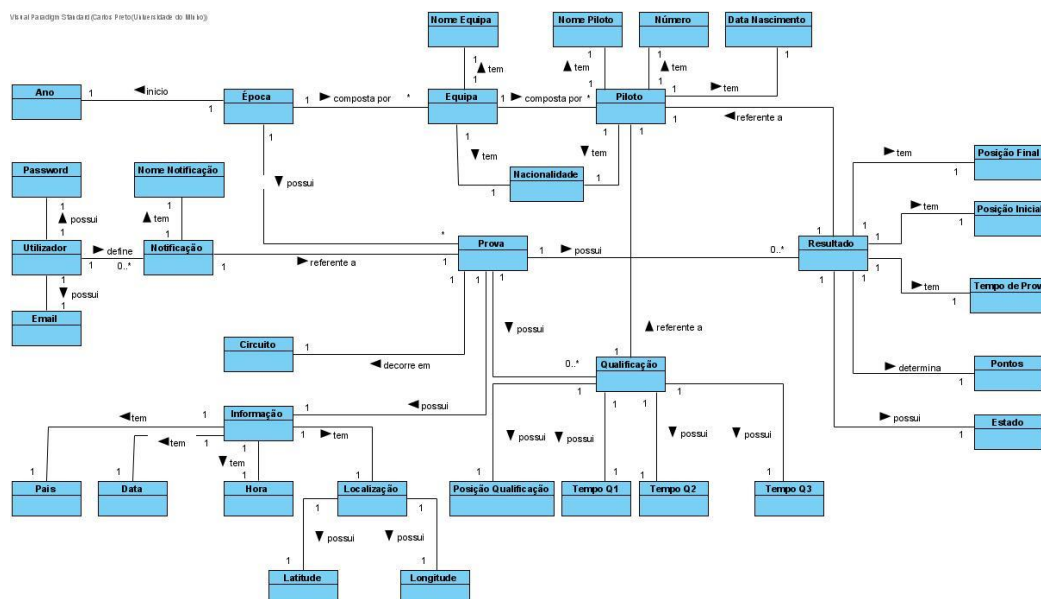


Figura 3 - Modelo de Domínio

Analisando a Figura 3, conseguimos perceber quais serão as nossas componentes principais, sendo estas o Utilizador, a Prova, o Resultado, a Qualificação, a Equipa, o Piloto e a Época, sendo que para cada uma destas componentes é também possível observar quais serão os seus atributos.

O Utilizador é caracterizado pelo seu email e pela sua password, sendo que um Utilizador também poderá ter notificações associadas ao seu perfil. A notificação será definida pelo nome do Utilizador que a criou e a que prova está associada. A prova será definida por um circuito e por um conjunto de informações que, no futuro, permitirão localizar esta no mapa do mundo. Uma prova terá várias Qualificações e Resultados, sendo que cada uma destas é referente a um dado piloto.

Um resultado é definido por uma posição final e inicial, um tempo de prova, os pontos associados à posição final e por um estado. A qualificação será composta por quatro atributos que a caracterizam, sendo esses: posição de qualificação e os tempos na qualificação 1, 2 e 3.

Uma equipa é caracterizada pelo seu nome, nacionalidade e pelos pilotos que a compõe. Um piloto, por sua vez, é caracterizado pelo seu nome, número, data de nascimento e nacionalidade. Quanto à época, definiu-se que esta seria composta pelo ano de início da época, e que uma época pode ter várias equipas e também várias provas.

3.2. Use Cases

Com base nos requisitos de Utilizador abordados no capítulo 2 é possível definir os Use Cases do nosso sistema e representa-los num diagrama, como se encontra na Figura 4. Os casos de utilização especificados irão simbolizar as diferentes interações possíveis de um Utilizador no nosso Sistema.

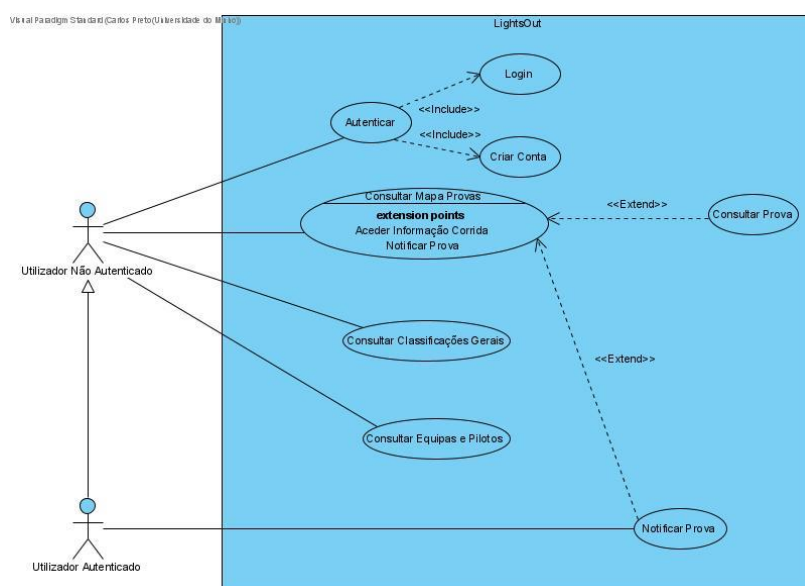


Figura 4 - Diagrama Use Cases

Após definidos os Use Cases, foi necessário elaborar a especificação para cada um destes, onde será abordado como será a interação entre o utilizador e o sistema, bem como procederá o sistema no caso do Utilizador realizar alguma ação que não corresponda ao funcionamento normal deste.

Uma vez que o processo de autenticação é bastante simples e perceptível, o grupo optou por não incluir a sua especificação.

3.2.1 Consultar Mapa de Provas

Use Case:	Consultar Mapa de Provas	
Ator:	Utilizador	
Pré-Condição:	-	
Pós-Condição:	True	
Fluxo Normal	Ator	Resposta Sistema
	1. Utilizador pretende aceder ao Mapa das Provas.	
		2. Sistema acede à latitude e longitude de cada prova a ocorrer no momento.
		3. Sistema apresenta mapa com as provas nas suas devidas localizações.
	4. Utilizador não pretende alterar o intervalo temporal	
	5. Utilizador não pretende receber notificação de nenhuma prova	
	6. Utilizador não pretende consultar nenhuma prova.	
Fluxo Alternativo 1[Utilizador altera intervalo temporal] (passo 4)		4.1. Sistema procura na base de dados provas que decorram nesse intervalo temporal.
		4.2. Repetir passo 3
Fluxo Alternativo 2[Utilizador pretende receber notificação de uma prova] (passo 5)	5.1. <<include>> Notificar Prova	
Fluxo Alternativo 3[Utilizador consulta uma prova] (passo 6)	6.1. <<Extend>> Consultar Prova	

Figura 5 - Especificação Mapa de Provas

Ao consultar o mapa de provas, serão apresentados três filtros, um para definir qual a data inicial pretendida, outro para definir qual a data final pretendida, e um último para variar a escala de progresso de cada filtro. Por defeito, quando se entra nesta View, a data Inicial definida será a atual, sendo esta igual à data final.

Sempre que os filtros forem alterados, o sistema terá de aceder à base de dados e apresentar novos resultados. Para melhor compreender o dinamismo deste processo, foi concebido um Digrama de Sequências, como se pode ver na Figura 6.

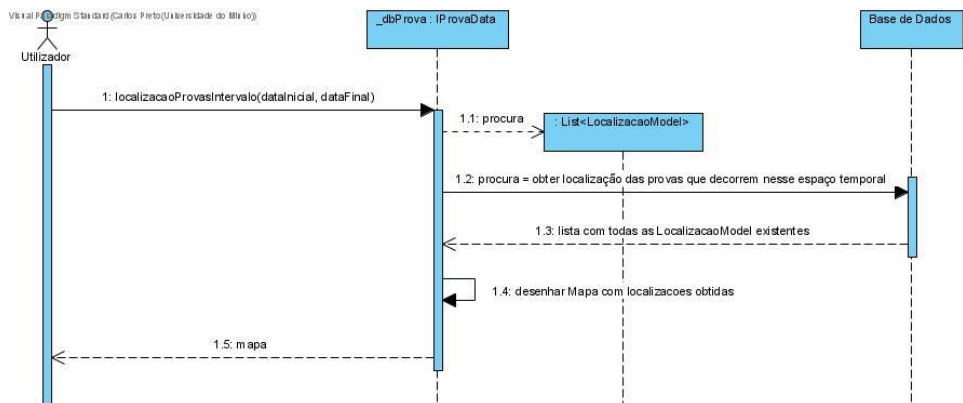


Figura 6 - Diagrama Sequência Mapa Provas

3.2.2 Consultar Prova

Use Case:	Consultar Prova	
Ator:	Utilizador	
Pré-Condição:	Utilizador tem de estar no mapa de provas e Prova ter terminado	
Pós-Condição:	True	
Fluxo Normal	Ator	Resposta Sistema
	1. Utilizador pretende aceder a informação de uma prova.	
		2. Sistema obtém resultados da prova e qualificação.
		3. Sistema apresenta tabelas com resultados e qualificações

Figura 7 - Especificação Consultar Prova

Sempre que uma prova for concluída, o Utilizador terá a possibilidade de consultar quer o resultado da prova, quer os tempos de qualificação que definiram as posições de arranque dos pilotos. Para tal, os sistema terá de ter conhecimento quer do nome da prova, quer da época na qual decorreu, uma vez que todos os anos as provas decorrem nos mesmos circuitos, e portanto é necessário distinguir as épocas.

Após obter a época e a prova pretendida, será necessário aceder a todos os resultados e qualificações associados a essa prova. No final, será devolvido uma lista de listas de *string*, onde cada lista de *string* terá a informação referente ao resultado de cada piloto que participou na prova, bem como o melhor tempo de qualificação de cada piloto.

Com base na especificação do Use Case, foi possível construir o respetivo Diagrama de Sequência, como se pode verificar na Figura 8.

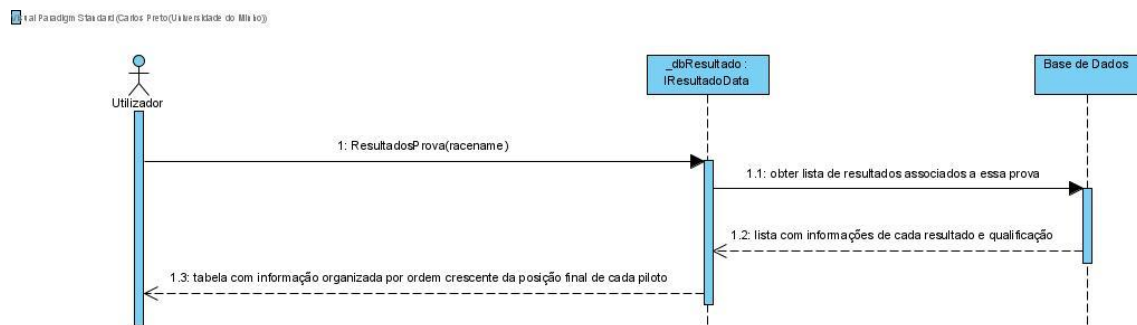


Figura 8 - Diagrama Sequência Consultar Prova

3.2.3 Notificar Prova

Use Case:	Notificar Prova	
Ator:	Utilizador Autenticado	
Pré-Condição:	Utilizador tem de estar autenticado	
Pós-Condição:	Nova notificação registada no Utilizador	
Fluxo Normal	Ator	Resposta Sistema
	1. Utilizador pretende aceder a notificações de uma prova.	
		2. Sistema verifica que a notificação da prova ainda não existe.
		3. Sistema regista notificação no Utilizador.
Fluxo Exceção 1[Notificação já existe] (passo 2)		2.1. Sistema informa que utilizador que notificação já existe.

Figura 9 - Especificação Notificar Prova

Sempre que o Utilizador se encontrar no mapa de provas e pretender receber notificação de uma dada prova, há certos requisitos que têm de ser satisfeitos. Primeiro, o sistema tem de verificar se o Utilizador está autenticado, de maneira a ser possível ao Sistema enviar notificações ao Utilizador. Só depois de verificar a autenticação é que se pode verificar se o Utilizador ainda não tem a notificação dessa prova, de maneira a não haver repetição de notificações já existentes.

Com base nestas informações, e tendo como base a especificação do Use Case, foi possível construir o respetivo Diagrama de Sequência, como se pode verificar na Figura 10.

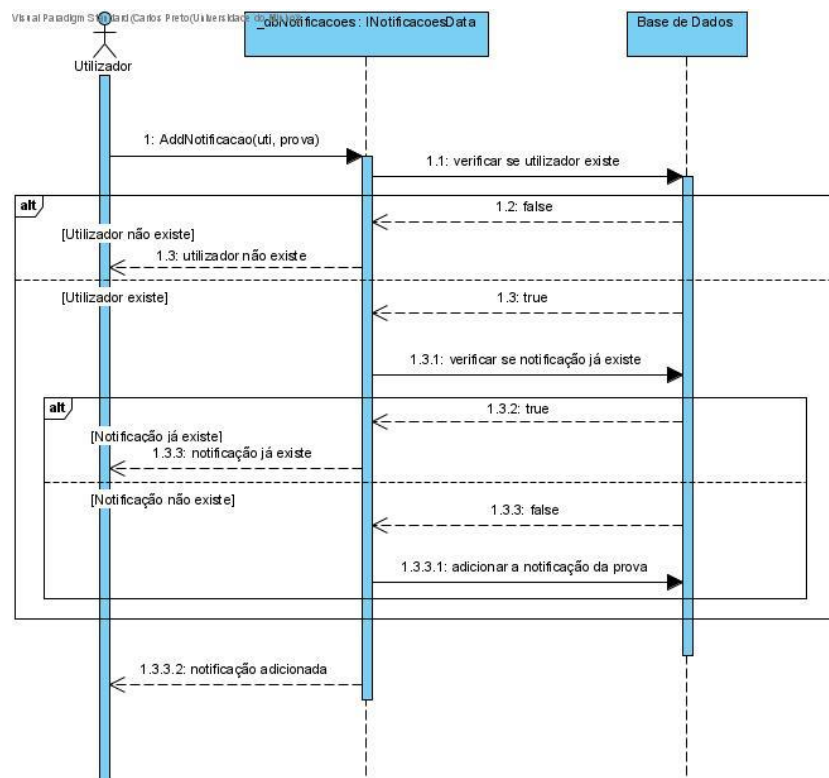


Figura 10 - Diagrama Sequência Notificar Prova

3.2.4 Consultar Classificações Gerais

Use Case:	Consultar Classificações Gerais	
Ator:	Utilizador	
Pré-Condição:	-	
Pós-Condição:	True	
Fluxo Normal	Ator	Resposta Sistema
	1. Utilizador pretende consultar Classificações Gerais.	
		2. Sistema procura, para a época atual, todos os resultados até á

		prova mais recente.
		3. Sistema calcula pontos de todos os pilotos.
		4. Sistema calcula pontos de todas as equipas.
		5. Sistema apresenta dois gráficos de barras com o pontos dos pilotos e pontos das equipas, ordenado decrescentemente.
	6. Utilizador não pretende alterar a prova	
	7. Utilizador não pretende alterar a época	
Fluxo Alternativo 1[Utilizador pretende alterar prova] (passo 6)	6.1. Utilizador altera prova.	
		6.2. Sistema procura, para a época atual, todos os resultados até à prova pretendida.
		6.3. Regressar passo 3.
Fluxo Alternativo 2[Utilizador pretende alterar época] (passo 7)	7.1. Utilizador altera época.	
		7.2. Sistema procura, para a época pretendida, todos os resultados até à última prova.
		7.3. Regressar passo 3.

Figura 11 - Especificação Classificações Gerais

Quando o Utilizador pretender consultar as Classificações Gerais, terá a possibilidade de alterar dois tipos de filtros, um referente à época pretendida e outro referente à prova até à qual pretender que sejam contabilizados os pontos, ou seja, se pretender a época 2020 e a prova 3, então será apresentado o somatório de pontos dos pilotos e das equipas até essa mesma prova.

Para melhor especificar este Use Case, recorreu-se ao desenvolvimento de dois diagramas de sequência distintos, sendo o primeiro referente à Classificação Geral dos Pilotos, e o segundo referente à Classificação Geral das Equipas.

Em ambos será necessário obter a época e todas as provas que decorreram nessa época até uma determinada a prova pretendida pelo utilizador. Posteriormente, para cada prova obtida, saber todos os pontos obtidos por cada piloto. No caso da classificação geral de pilotos, devolve-se um mapa com o Piloto e os seus pontos, estando este mapa ordenado por ordem decrescente de pontos. No caso da classificação geral de equipas, será necessário obter a equipa associado a cada piloto e, posteriormente, devolver um mapa com a Equipa e os seus pontos, ordenado, mais uma vez, por ordem decrescente.

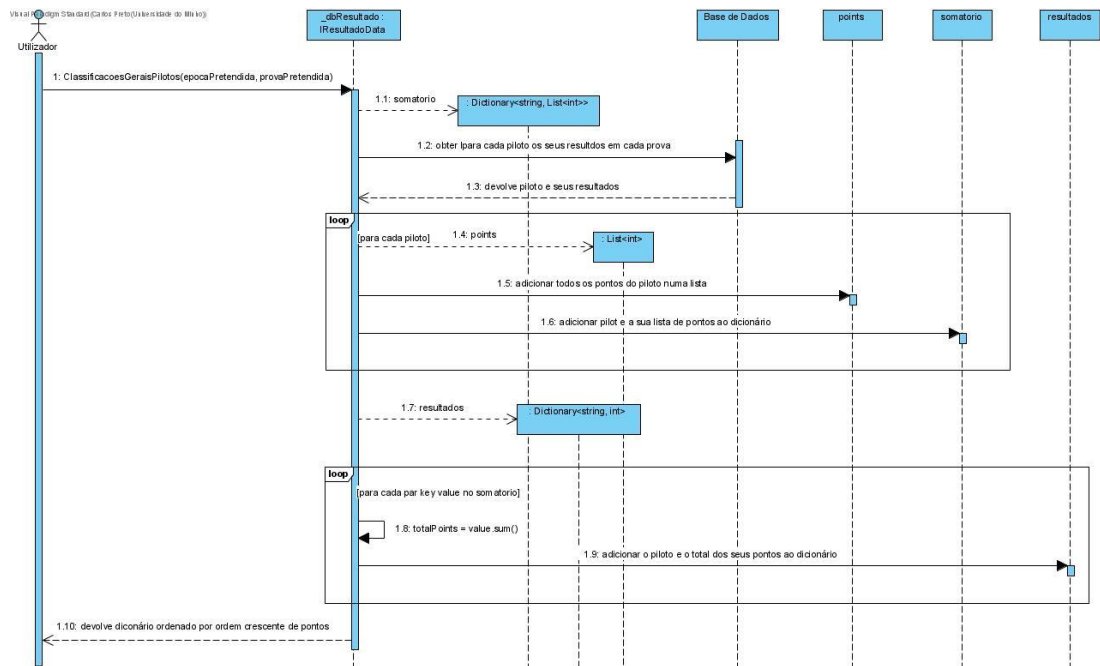


Figura 12 - Diagrama Sequência Classificação Pilotos

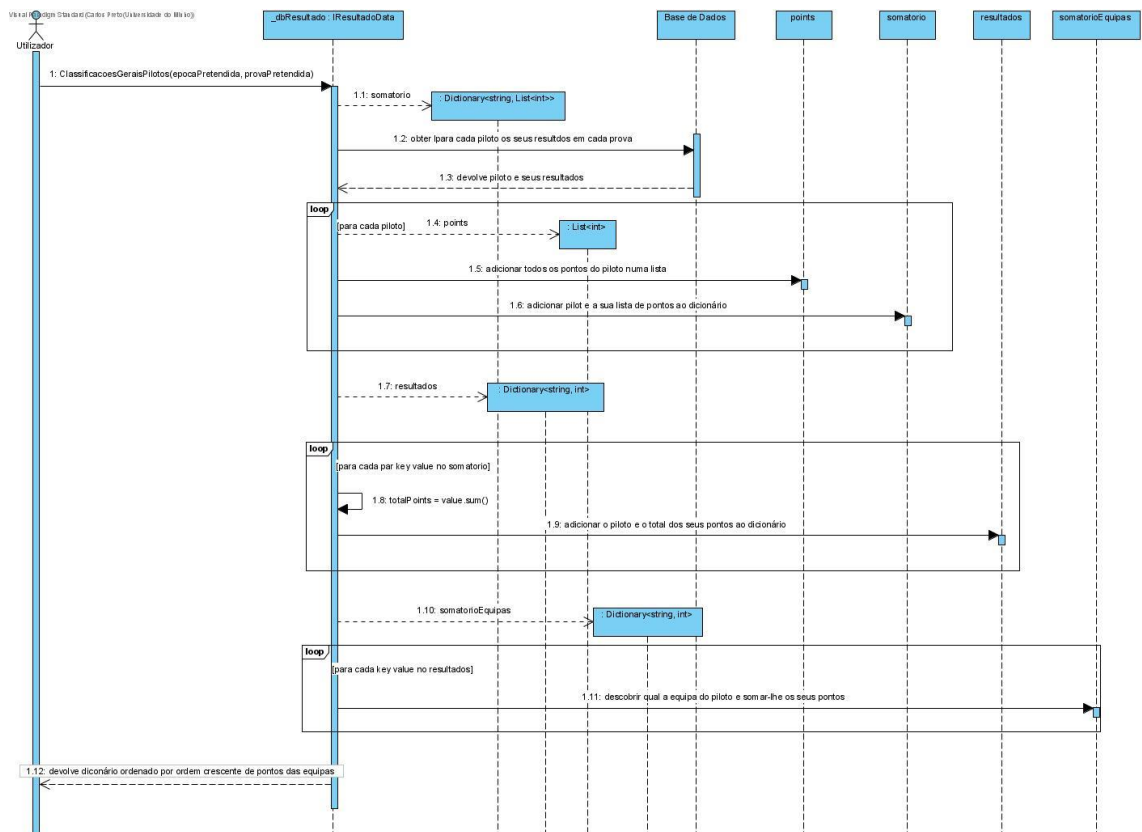


Figura 13 - Diagrama Sequência Classificação Equipas

3.2.5 Consultar Equipas e Pilotos

Use Case:	Consultar Equipas e Pilotos	
Ator:	Utilizador	
Pré-Condição:	-	
Pós-Condição:	True	
Fluxo Normal	Ator	Resposta Sistema
	1. Utilizador pretende consultar equipas e pilotos.	
		2. Sistema procura, para a época atual, todas as equipas.
		3. Sistema apresenta, numa tabela, equipas e os seus pilotos.
	4. Utilizador não pretende alterar Época.	
Fluxo Exceção 1[Utilizador altera Época] (passo 4)	4.1. Utilizador altera época.	
		4.2. Sistema procura, para a época pretendida, todas as equipas.
		4.4. Regressar passo 3.

Figura 14 - Especificação Equipas e Pilotos

Quando o Utilizador pretender consultar as equipas e pilotos, terá, mais uma vez, a possibilidade de alterar a época pretendida. Será necessário obter a época pretendida e, posteriormente, aceder a todas as equipas que participaram no campeonato do mundo de construtores de F1 nessa mesma época.

Apesar de ser um Use Case bastante simples, em comparação com os restantes, o grupo decidiu construir o seu Diagrama de Sequência, como se pode ver na Figura 15.

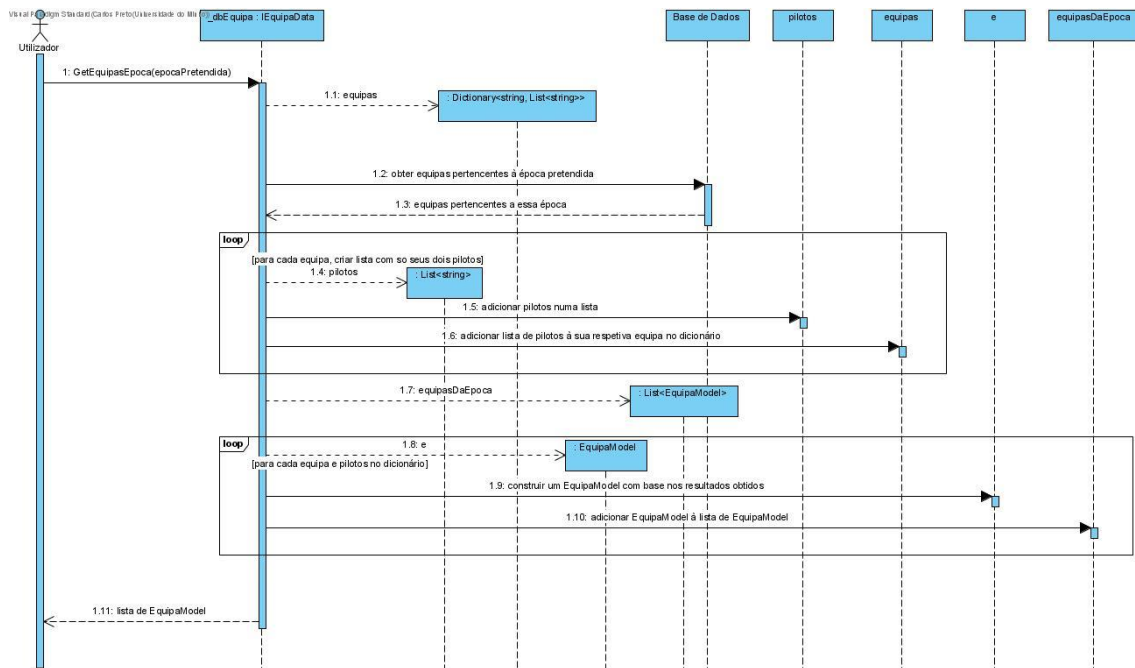


Figura 15 - Diagrama Sequência Equipas e Pilotos

3.3. Diagrama de Classes

O Diagrama de Classes definido pelo grupo contém os futuros *models* do nosso sistema. A cada uma das classes foram atribuídos diferentes atributos que, numa fase inicial, identificarão cada uma destas. As classes definidas são resultado de uma reinterpretação feita após concluída a fase de implementação do projeto, visto que foi necessário modificar as classes previamente definidas, bem como os seus atributos.

O projeto foi dividido em três componentes principais, cada uma com as suas classes, sendo estas a *Frontend*, a *DataAccessLibrary* e a *Backend*, sendo que é possível à *Frontend* comunicar com a *DataAccessLibrary* através da injeção de uma conexão da *DataAccessLibrary* na *Frontend*.

A *DataAccessLibrary* contém todos os modelos necessários para se obter dados da base de dados. Todas as classes terminadas em “Data” representam as classes que contém os métodos de acesso à base de dados, sendo que as classes terminadas com “Model” apenas contém os atributos de uma dada classe. A cada classe “Model” está associada uma classe “Data”, o que permite uma melhor gestão das entidades do Sistema. Também se definiu uma interface para cada uma das classes “Data”, para que assim, caso seja necessário alterar o tipo de alguma função, tal não implique modificar o código todo do projeto.

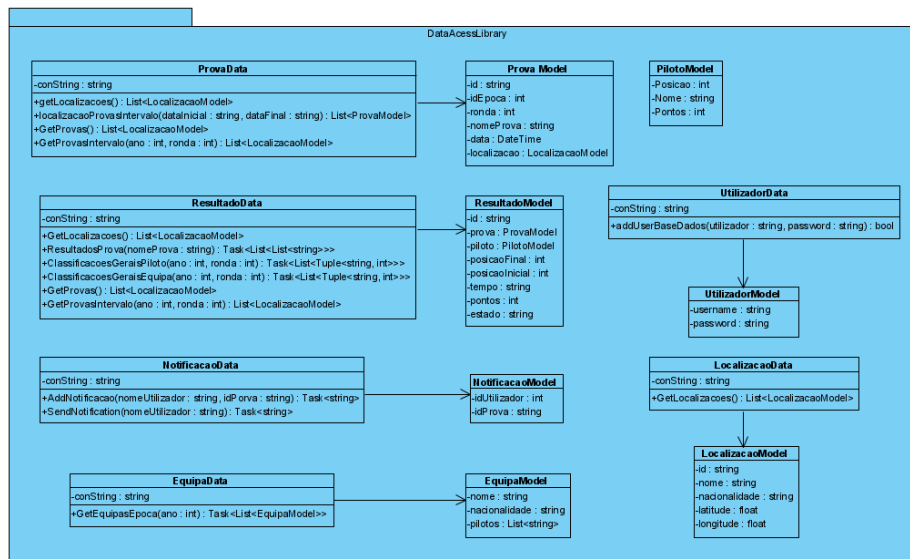


Figura 16: Diagrama de Classes DataAccessLibrary

A *Backend* é apenas composta por uma classe denominada “*Program*”. Esta classe é responsável por aceder à *API* do *Ergast* e armazenar toda a informação na Base de Dados. Do *Ergast* pode ser obtida informação acerca das Epocas, circuitos, equipas, pilotos, provas, resultados e qualificações.

De maneira a otimizar os acessos à base de dados, verifica-se se a base de dados contém informação, e caso já tenha informação, apenas se adiciona as informações referentes à época de 2021. Caso contrário, serão inseridas todas as informações relativas à F1, desde a época de 2000.

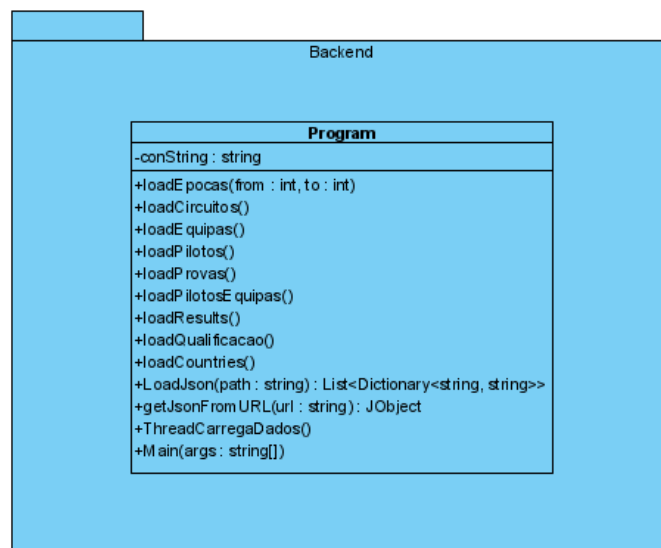


Figura 17: Diagrama Classes Backend

Por fim, na *Frontend*, foi também necessário criar três classes. A classe *CustomAuthenticationStateProvider* é responsável por obter os dados do utilizador autenticado, e tratar de fazer o *logout* deste. Por sua vez, a classe *PageHistoryState* permite guardar um histórico de todas as Views acedidas pelo Utilizador. Para que assim, quando por exemplo, o Utilizador for da página inicial para a View Login, o Sistema sabe que uma vez concluído o *Login* o Utilizador será reencaminhado para a página inicial. A classe *TimerService* é responsável por definir a taxa de refrescamento do Sistema, de maneira a se poder obter informação em tempo próximo de real.

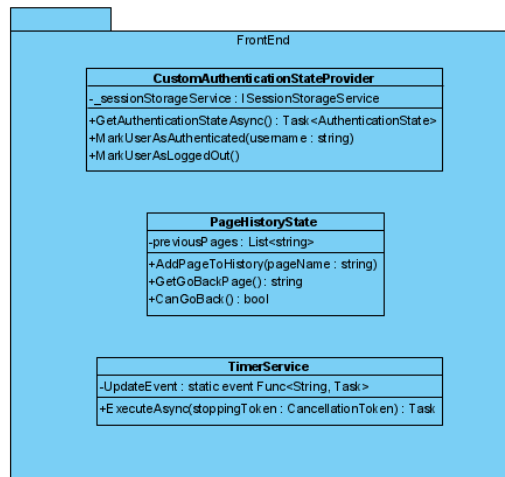


Figura 18: Diagrama Classes Frontend

3.4. Máquina de Estado

Para melhor organizar as ideias relativamente às interações entre o Utilizador e o Sistema, o grupo desenvolveu uma Máquina de Estado. Como se pode verificar na Figura 19, através da página inicial será possível aceder às restantes Views que, posteriormente, poderão ser navegadas e modificadas conforme as preferências do Utilizador, através da alteração de filtros.

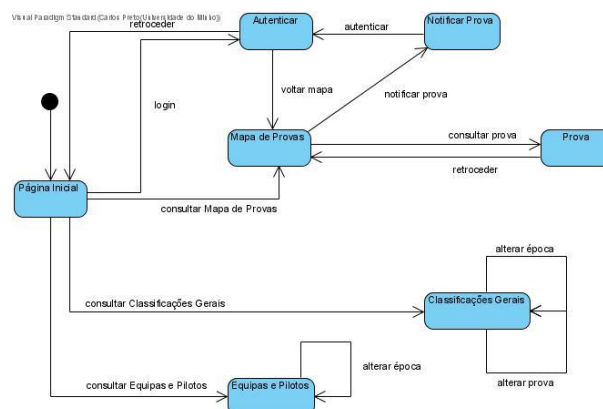


Figura 19 - Máquina de Estado

4. Base de Dados Relacional

4.1. Esquema Lógico

Uma vez realizada a modelação do sistema, passou-se para a planificação da base de dados relacional. Com base no Diagrama de Classe definido pelo grupo, foi possível elaborar o esquema lógico do nosso sistema. Na terceira fase, foi necessário alterar o esquema lógico previamente definido na segunda fase, uma vez que faltava considerar algumas dependências e acrescentar alguns atributos necessários para caracterizar a informação disponível. Assim, o modelo lógico apresentado na Figura 20 é representativo da estrutura final da nossa base de dados.

No modelo lógico, há que destacar a criação de uma tabela nova, denominada “*PilotoEquipa*”, uma vez que, durante a fase de implementação, o grupo verificou que um piloto pode não estar associado a apenas uma equipa, sendo então importante associar um piloto e a sua equipa, consoante a época pretendida.

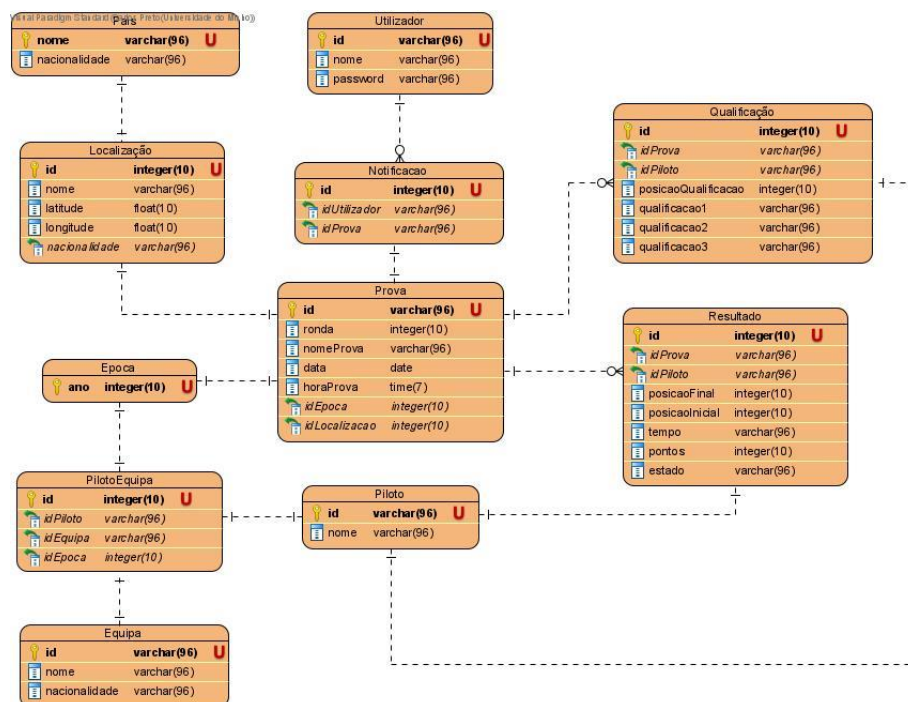


Figura 20 - Modelo Lógico

4.2. Dicionário de Dados e Estimativa de Espaço

De maneira a explicitar cada uma das tabelas e descrever cada atributo, foi necessário elaborar um dicionário de dados. Este dicionário de dados permitirá um nível de análise mais detalhado do nosso esquema lógico, para que assim seja perfeitamente perceptível a razão pela qual um atributo existe.

Além disso, o grupo optou por acrescentar no dicionário de dados o espaço que cada entidade e atributo irão ocupar, para ter uma ideia da gestão futura que estará associada à manutenção do projeto.

Tabela	Atributo	Tipo Dados	Espaço Ocupado	Descrição	Exemplo
Utilizador	id	int	10 bytes	Identificador único de um Utilizador	1
	Nome	Varchar(96)	96 bytes	Nome Utilizador	Carlos Preto
	Password	Varchar(96)	96 bytes	Password Utilizador	cj1234
Notificacao	id	Int	4 bytes	Identificador único de uma Notificacao	3
	idUtilizador	Varchar(96)	96 bytes	Identificador do Utilizador da Notificação	Carlos Preto
	idProva	Varchar(96)	96 bytes	Identificador da Prova que corresponde a Notificação	PortugueseGrandPrix
Prova	id	Varchar(96)	96 bytes	Identificador da Prova	PortugueseGrandPrix
	ronda	Int	10 bytes	Identificador da Época em que decorre a prova	12
	nomeProva	Varchar(96)	96 bytes	Nome Prova	Portuguese Grand Prix
	data	Date	3 bytes	Data da Prova	2021-05-02
	horaProva	Time	5 bytes	Hora da Prova	14:00:00Z
	idEpoca	Int	10 bytes	Identificador da época	2021
	idLocalizacao	Int	10 bytes	Identificador da Localização	10
Epoca	ano	Int	10 bytes	Identificador da Época	2021
Localizacao	id	Int	10 bytes	Identificador Localizacao	4
	nome	Varchar(96)	96 bytes	Pais onde se situa uma localizacao	Portugal

	latitude	Float(10)	10 bytes	Coordenada de Latitude	-37.8497
	longitude	Float(10)	10 bytes	Coordenada de Longitude	144.968
	nacionalidade	Varchar(96)	96 bytes	Nacionalidade associada ao País	Portuguese
Resultado	id	Int	10 bytes	Identificador do resultado	40
	idPiloto	Varchar(96 bytes)	96 bytes	Identificador do Piloto do Resultado	lando
	idProva	Varchar(96 bytes)	96 bytes	Identificador da Prova associada ao Resultado	PortugueseGrandPrix
	posicaoFinal	Int	10 bytes	Posição final do Piloto	1
	posicaoInicial	Int	10 bytes	Posição Inicial do Piloto	5
	Tempo	Varchar(96 bytes)	96 bytes	Tempo de Prova	2:02:34.598
	Pontos	Int	10 bytes	Pontos obtidos	20
	Estado	Varchar(96)	96 bytes	Como piloto concluiu a prova	Collision
Qualificacao	id	int	10 bytes	Identificador Qualificação	31
	idProva	Varchar(96 bytes)	96 bytes	Identificador da Prova associada ao Resultado	PortugueseGrandPrix
	idPiloto	Varchar(96 bytes)	96 bytes	Identificador do Piloto associada ao Resultado	lando
	posicaoQualificacao	Int	10 bytes	Posição na qual Piloto se qualificou	5
	qualificacao1	Varchar(96 bytes)	96 bytes	Tempo na Qualificacao1	1:25.567
	qualificacao2	Varchar(96 bytes)	96 bytes	Tempo na Qualificacao2	1:25.009
	qualificacao3	Varchar(96 bytes)	96 bytes	Tempo na Qualificacao3	1:24.679
Piloto	id	Varchar(96)	96 bytes	Identificador Piloto	lando
	Nome	Varchar(96)	96 bytes	Nome do Piloto	Lando Norris
Equipa	id	Varchar(96)	96 bytes	Identificador da Equipa	red_bull
	nome	Varchar(96)	96 bytes	Nome da Equipa	Red Bull
	nacionalidade	Varchar(96)	96 bytes	Nacionalidade Equipa	British

PilotoEquipa	id	int	10 bytes	Identificador PilotoEquipa	2
	idPiloto	Varchar(96)	96 bytes	Identificador Piloto	lando
	idEquipa	Varchar(96)	96 bytes	Identificador Equipa	mclaren
	idEpoca	int	10 bytes	Identificador Época	2021

Figura 21 - Dicionário de Dados

Como se pode verificar na Figura 21, cada atributo irá ocupado um determinado tamanho na nossa base de dados, sendo que, com base no tamanho de cada atributo, será possível saber qual o tamanho ocupado por cada entidade da Base de Dados:

- ✓ Utilizador: 202 bytes
- ✓ Notificação: 196 bytes
- ✓ Prova: 230 bytes
- ✓ Época: 10 bytes
- ✓ Localização: 222 bytes
- ✓ Resultado: 424 bytes
- ✓ Qualificacao: 500 bytes
- ✓ Piloto: 192 bytes
- ✓ Equipa: 288 bytes
- ✓ PilotoEquipa : 212 bytes

De maneira a estimar o espaço que toda a informação referente a uma dada época ocupa na base dados, tomou-se como exemplo a época de 2020. Durante essa época, houve 17 provas ($17 \times 230 = 3\,910$ bytes), cada prova tem uma localização ($17 \times 222 = 3\,774$ bytes), 20 resultados ($17 \times 20 \times 424 = 144\,160$ bytes), 20 qualificações ($17 \times 20 \times 500 = 170\,000$ bytes), 20 pilotos ($20 \times 192 = 3\,840$ bytes), 10 equipas ($10 \times 288 = 2\,880$ bytes) e 20 PilotoEquipa ($20 \times 212 = 4\,240$ bytes).

Além das épocas, também é necessário calcular o espaço ocupado pelos Utilizadores. Imaginando que, por época, o grupo espera receber 20 Utilizadores ($20 \times 202 = 4\,040$ bytes), e que em média, cada Utilizador terá 4 Notificações ($4 \times 196 = 784$ bytes) associadas ao seu perfil, pode-se concluir que teremos uma taxa de crescimento anual de, possivelmente, 337 628 bytes (332 804 bytes das épocas mais 4 824 bytes de Utilizadores e Notificações).

Relativamente ao espaço inicial da base de dados, o grupo pretende guardar dados referentes às últimas 20 épocas, ou seja, inicialmente, serão ocupados 6 656 080 bytes na base de Dados.

5. Definição do sistema de Interface (*Mockups*)

Após definida a Base de Dados Relacional, e de maneira a ajudar a definir a interface visual do nosso sistema, o grupo desenvolveu um conjunto de *Mockups*. Cada *Mockup* corresponderá a uma *View* no nosso sistema final, estando porém, cada *View*, suscetível a sofrer alterações até à versão final do projeto.

Também é importante referir que o grupo optou por apresentar as *Views* segundo a perspetiva de um browser de um computador, não obstante o facto de, no final, o nosso sistema poder ser usado em qualquer tipo de *browser*, independentemente do aparelho em questão.

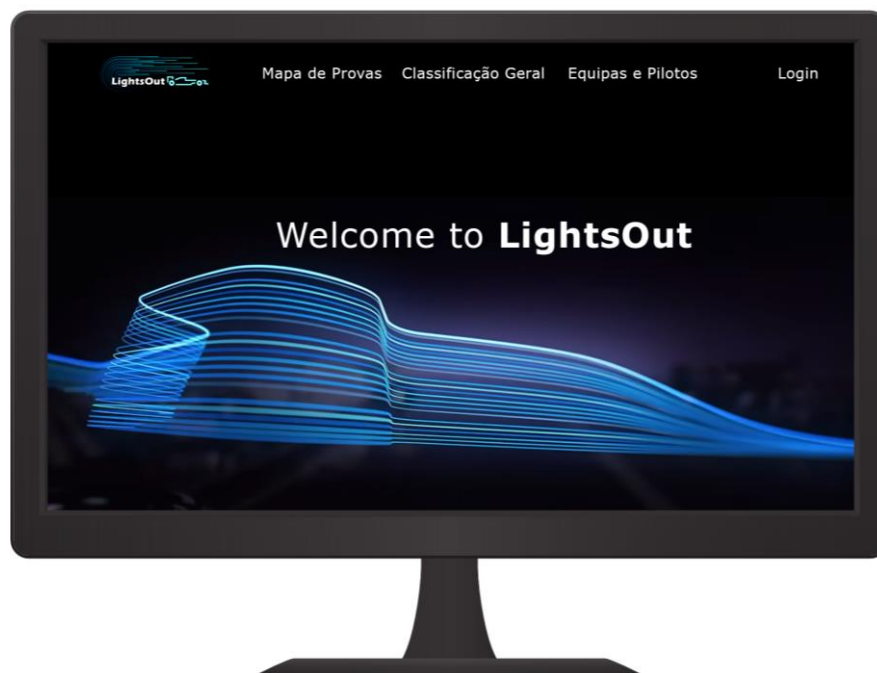


Figura 22 - Mockup Página Inicial

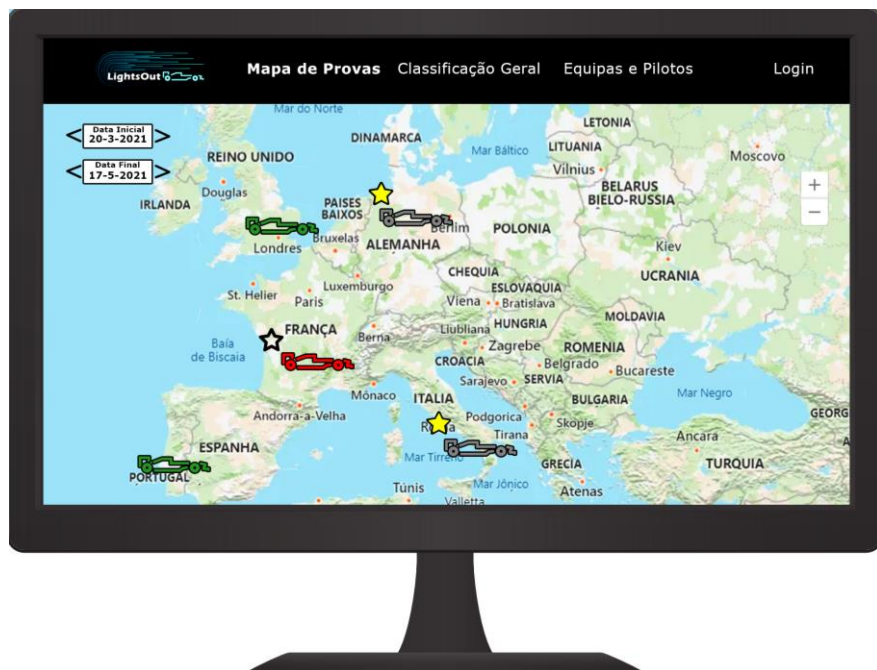


Figura 23 - Mockup Mapa de Provas

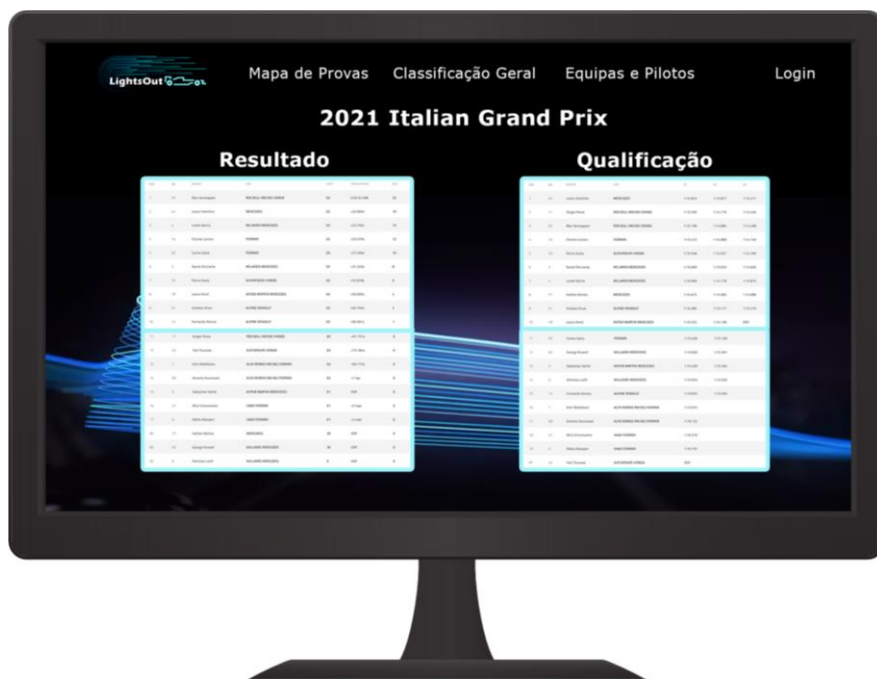


Figura 24 - Mockup Prova



Figura 25 - Mockup Classificações Gerais

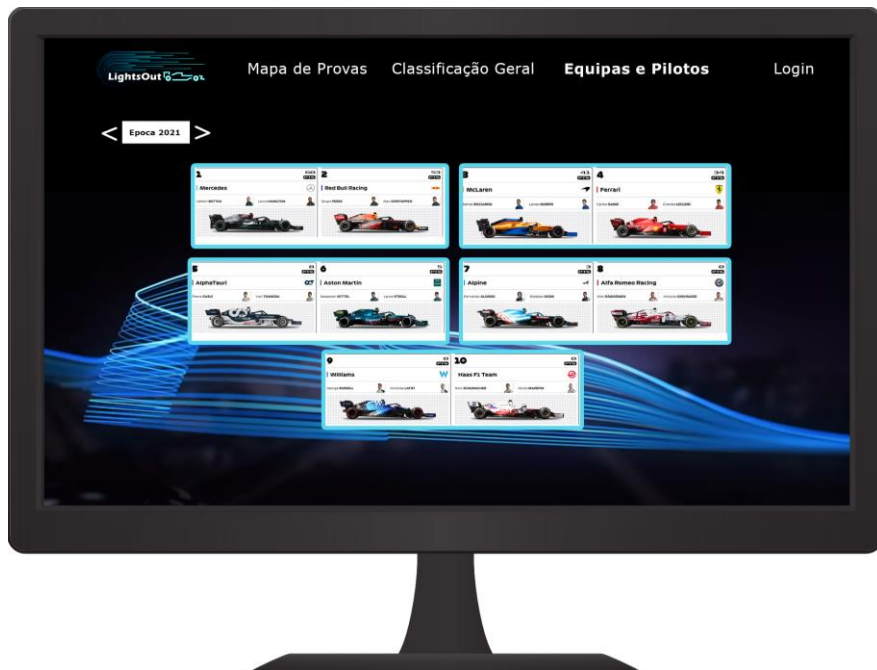


Figura 26 - Mockup Equipas e Pilotos

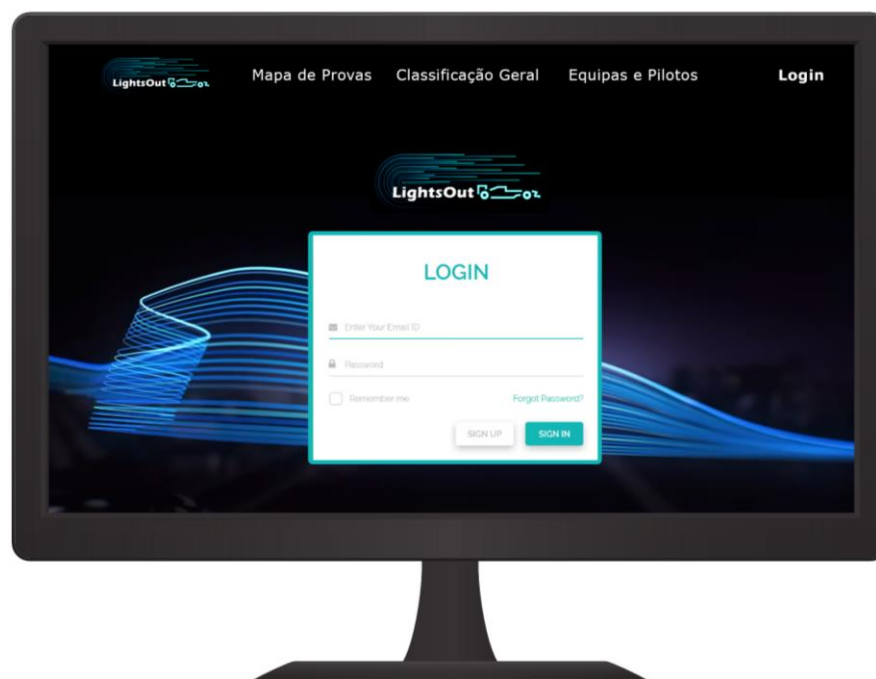


Figura 27 - Mockup Login

6. Implementação

Uma vez organizadas as ideias relativas às funcionalidades do projeto, foi possível avançar para a última parte do desenvolvimento do projeto, referente à implementação das funcionalidades previamente descritas.

6.1. Ferramentas Utilizadas

Antes de se iniciar a fase de implementação, foi feita uma pequena pesquisa acerca das melhores abordagens para desenvolvimento de uma aplicação *.NET*.

Uma vez que o grupo, até ao momento, nunca tinha trabalho com *.NET*, optou-se por seguir a estrutura *Blazor Server*. Apesar de ser relativamente recente, esta tipo de estrutura, tal como referido nos seminários da cadeira de Laboratórios de Informática, acaba por ser o ideal para um primeiro contacto com *.NET*. Nesta estrutura, é possível criar *Razor Pages*, que por sua vez permitem criar páginas *HTML* onde se pode invocar funções através de *C#*.

De maneira a poder armazenar, aceder e manipular a informação pretendida na Base de Dados, utilizou-se o *Microsoft SQL Server*. Como plataformas de desenvolvimento do projeto, utilizou-se o *Microsoft Visual Studio* e *Visual Studio Code*. Relativamente às linguagens utilizadas, o *C#* foi a linguagem predominante, juntamente com o *HTML*. Além destas duas, e de maneira a poder ter elementos dinâmicos, tais como o *Bing Maps*, recorreu-se ao *JavaScript*. Também se recorreu ao uso das *API's* do *Bing Maps* e do *Ergast F1* para se obter o mapa global e as informações de *F1* necessárias, respetivamente.

6.2. Apresentação do Sistema

6.2.1 Consultar Mapa de Provas

O nosso projeto permite acompanhar e reportar em tempo (próximo) de real os eventos de F1. Acedendo à categoria “Mapa de Provas”, é apresentado um mapa e três filtros distintos. Cada um destes filtros permitirá ao Utilizador configurar a quantidade de informação apresentada no ecrã.

O primeiro filtro define a data inicial pretendida, o segundo define a data final pretendida, e o terceiro define qual a escala de progresso e retrocesso pretendida. Quando se acede ao mapa, a data inicial e final é a mesma, isto porque se pretende que sejam apresentadas as provas a ocorrer nesse mesmo dia. Caso o Utilizador pretenda visualizar as provas que já ocorreram ou as que irão ocorrer, basta alterar os filtros das datas. Sempre que se carregar no botão para avançar ou recuar numa data, esse valor varia conforme o valor no filtro “Range”, ou seja, caso “Range” esteja a 1, significa que vamos avançar ou recuar um dia.

No local onde ocorrer um prova, será apresentado um *Icon* sobre o local associado a uma prova. Para que se possam distinguir as provas que já decorreram, que estão a decorrer, ou que irão decorrer, optou-se por utilizar o seguinte esquema de cores:

- ✓ **Verde:** Prova já Terminou
- ✓ **Vermelho:** Prova está a decorrer
- ✓ **Preto:** Prova ainda não começou

O grupo considerou que, uma prova já terminou quando a data da prova é inferior à atual, uma prova está a decorrer quando a hora da prova coincide com a atual ou é maior em 2 horas (visto que um prova de F1 demora cerca de 2 horas), e uma prova ainda não começou quando a sua data é superior à atual.

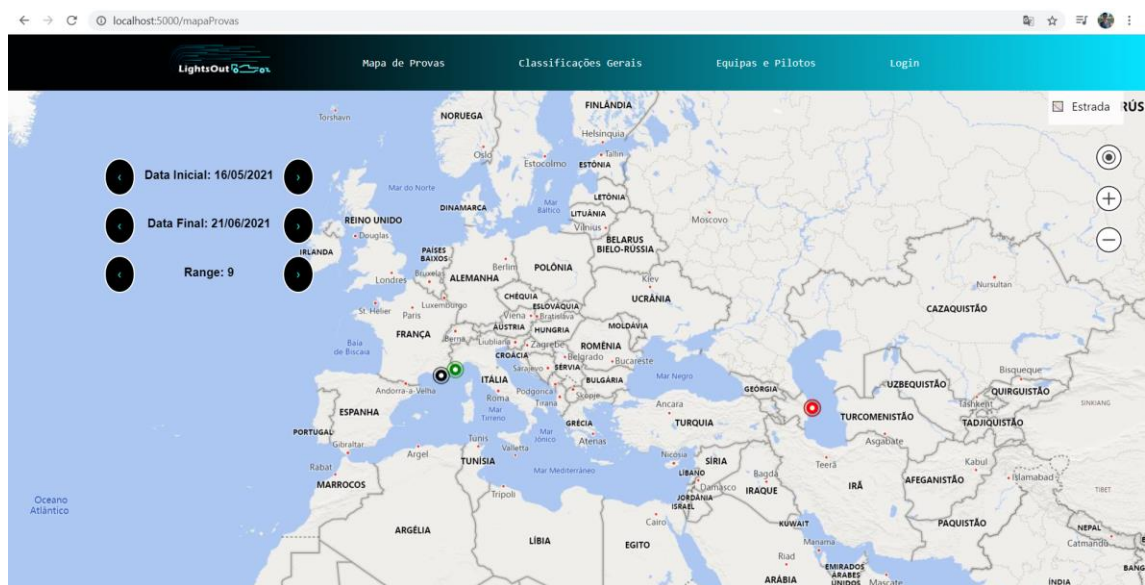


Figura 28: Distinção das Provas

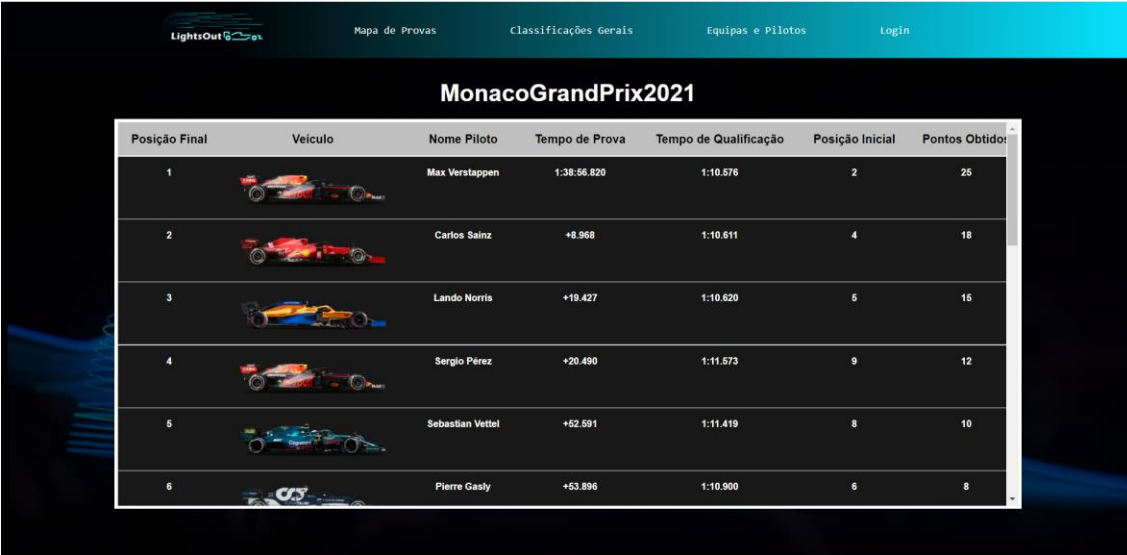
De maneira a manter a informação no mapa atualizada, considerou-se que a cada minuto, o nosso mapa vai automaticamente aceder à base de dados e apresentar a informação mais recente. Esta estratégia permite que, caso o Utilizador esteja no mapa sem realizar nenhuma ação e uma prova comece, então essa prova irá aparecer a vermelho, demonstrando assim ao Utilizador que a prova começou. Caso a prova termine, então a prova passa de vermelho para verde.

6.2.2 Consultar Prova

O resultado de uma prova em específico apenas poderá ser consultado através do mapa. Quando um *Icon* verde aparecer em cima de um país, se o Utilizador pressionar esse mesmo *Icon*, será levado para uma página com o resultado dessa mesma prova.

O resultado irá apresentar, por ordem crescente de posição final, a posição final de cada piloto, o veículo no qual participou na prova, o nome de cada piloto, o tempo total de prova, ou seja, quanto tempo demorou a concluir a prova, e os pontos obtidos.

Em relação à qualificação, de maneira a conseguir apresentar toda a informação numa única tabela, o grupo optou por, para cada piloto, apresentar qual o seu melhor tempo de qualificação e qual a posição inicial na grelha, associado a esse mesmo tempo.









Posição Final	Veículo	Nome Piloto	Tempo de Prova	Tempo de Qualificação	Posição Inicial	Pontos Obtidos
1		Max Verstappen	1:38:56.820	1:10.576	2	25
2		Carlos Sainz	+8.968	1:10.611	4	18
3		Lando Norris	+18.427	1:10.620	5	15
4		Sergio Pérez	+20.490	1:11.573	9	12
5		Sebastian Vettel	+52.591	1:11.419	8	10
6		Pierre Gasly	+63.896	1:10.900	6	8

Figura 29: Exemplo Resultado Prova

6.2.3 Login

De maneira a controlar os Utilizadores no nosso sistema e as notificações associadas a cada um deles, implementou-se um mecanismo de *Login*. Acedendo à categoria “*Login*”, será apresentada uma secção onde o Utilizador terá de preencher dois campos, um referente ao nome do Utilizador e outro referente à sua password. Uma vez concluídos esses passos, ao seleccionar o botão “Entrar”, o sistema irá verificar se as credenciais já existem no sistema. Caso existam, não se insere essas credenciais na base de dados, caso contrário, insere-se.

Sempre que um Utilizador estiver autenticado, na página inicial aparecerá o seu nome, e quando se quiser desconectar, poderá seleccionar a opção “*Logout*”, na barra de navegação.

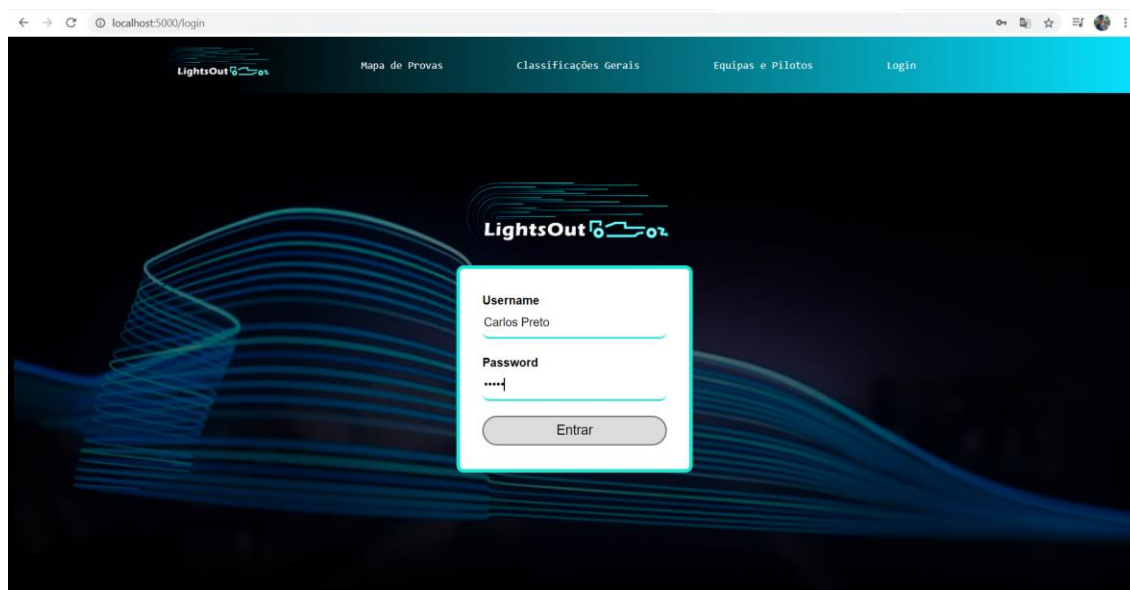


Figura 30: Login

6.2.4 Notificar Prova

Uma vez mais, caso um Utilizador pretenda receber notificações acerca de uma prova, terá de aceder a essa mesma prova através do mapa de provas. De maneira a gerir as notificações, optou-se por apenas disponibilizar este mecanismo ao Utilizador, caso este se encontre autenticado. As notificações apenas estão disponíveis para as provas que ainda não começaram, ou seja, as provas que permitem receber notificações são aquelas onde, no mapa, aparece um *Icon* preto sobre a sua localização.

Dito isto, quando um Utilizador carregar sobre essas provas, será reencaminhado para uma página onde se encontra uma imagem relativa ao circuito onde essa prova decorrerá, e um botão que, uma vez pressionado, será responsável por associar a notificação da prova ao Utilizador.

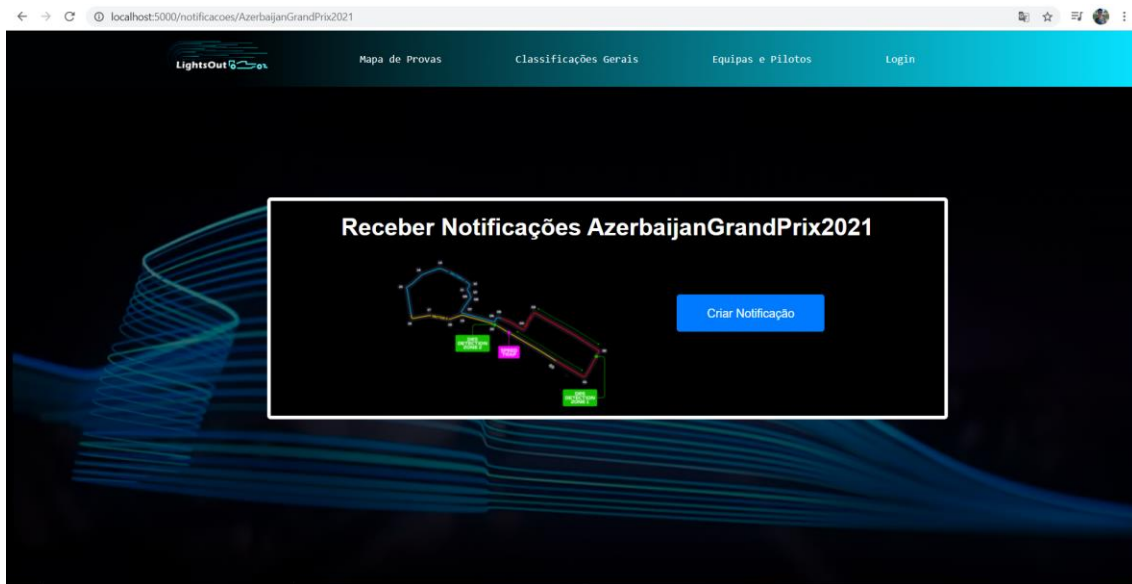


Figura 31: Criar Notificação

Porém, é importante referir que existem três casos possíveis de acontecer, aquando da criação da notificação da prova:

Utilizador não está Autenticado

Caso o Utilizador não esteja autenticado, o Sistema irá enviar uma notificação a informar da necessidade do Utilizador estar autenticado, caso queira receber a notificação da prova, e irá reencaminhar este para a página de login.

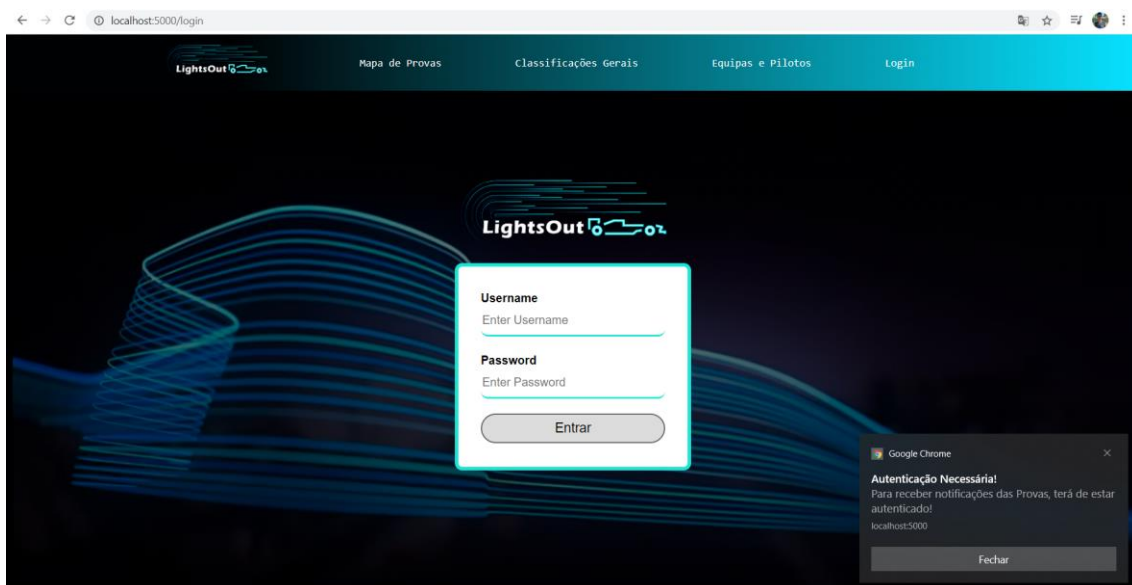


Figura 32: Autenticação Necessária

Notificação já Existe

Pode acontecer que o Utilizador, no passado, tenha acionado as notificações para uma prova e por isso não é necessário guardar a notificação na base de dados, visto que esta já se encontra presente nessa. Quando tal acontece, é enviada uma notificação a informar da prévia existência da notificação, sendo posteriormente o Utilizador reencaminhado para o mapa de provas.

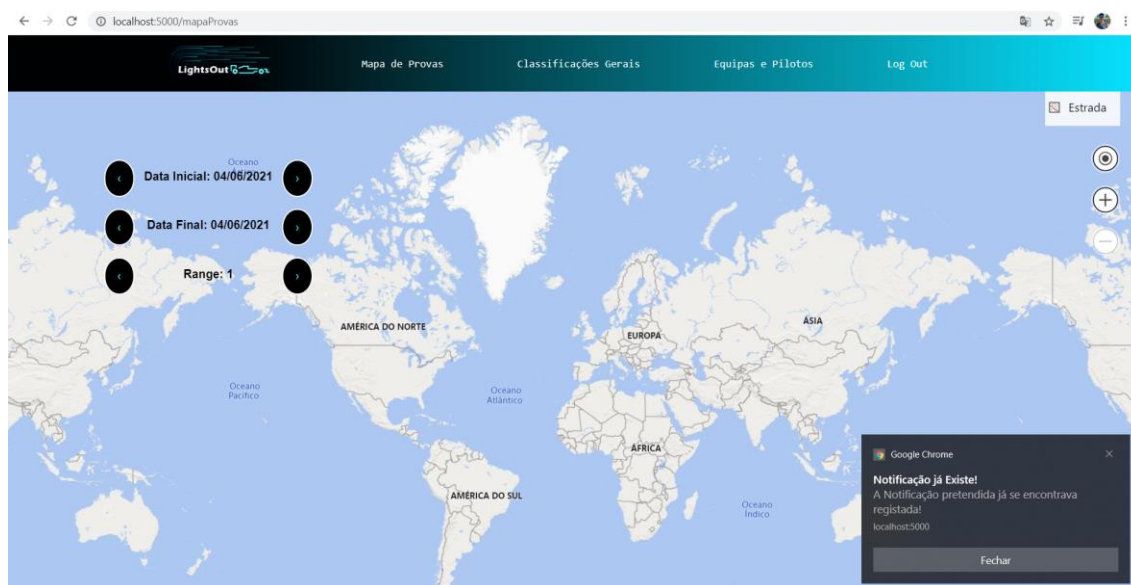


Figura 33: Notificação já Existe

Notificação ainda não Existe e Utilizador Autenticado

Quando são cumpridos todos os requisitos para se poder criar uma notificação, a notificação é adicionada à base de dados e é enviada uma notificação ao Utilizador a informar do registo da notificação com sucesso.

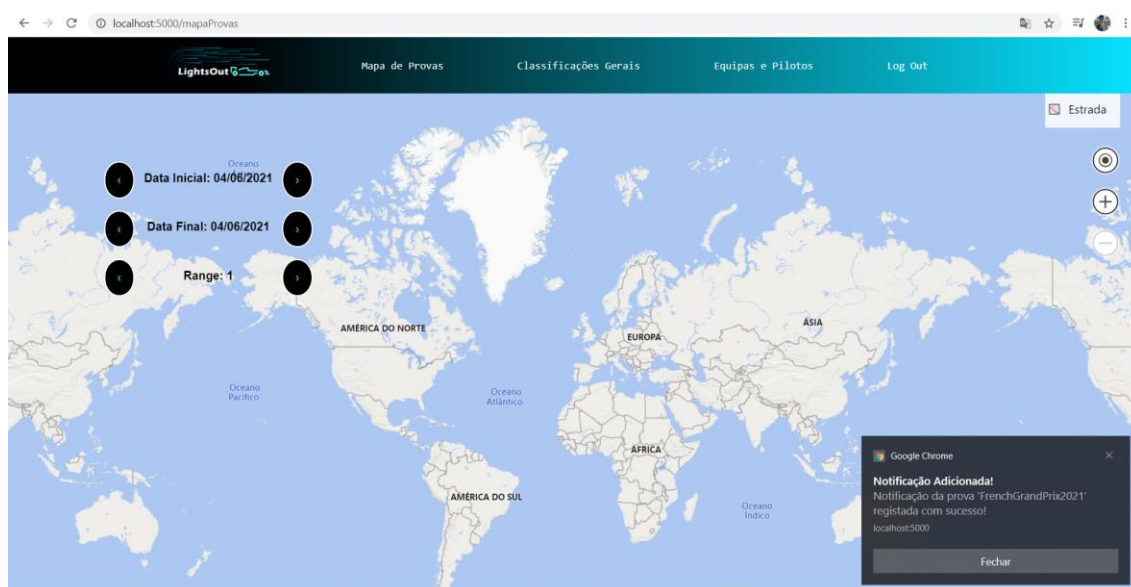


Figura 34: Notificação Adicionada

Após se adicionar uma notificação à base de dados, é importante controlar quando é que o Utilizador deve ser notificado do início da prova associada a essa notificação. Para tal, de minuto em minuto, o sistema atualiza automaticamente, e caso a hora do sistema coincida com a data de início da prova, então é enviada a notificação ao Utilizador a informar do início da prova e posteriormente, remove-se a notificação da base de dados, porque esta já foi invocada. Também é importante referir que o Utilizador só receberá a notificação se, na hora de início da prova, este estiver autenticado no sistema.

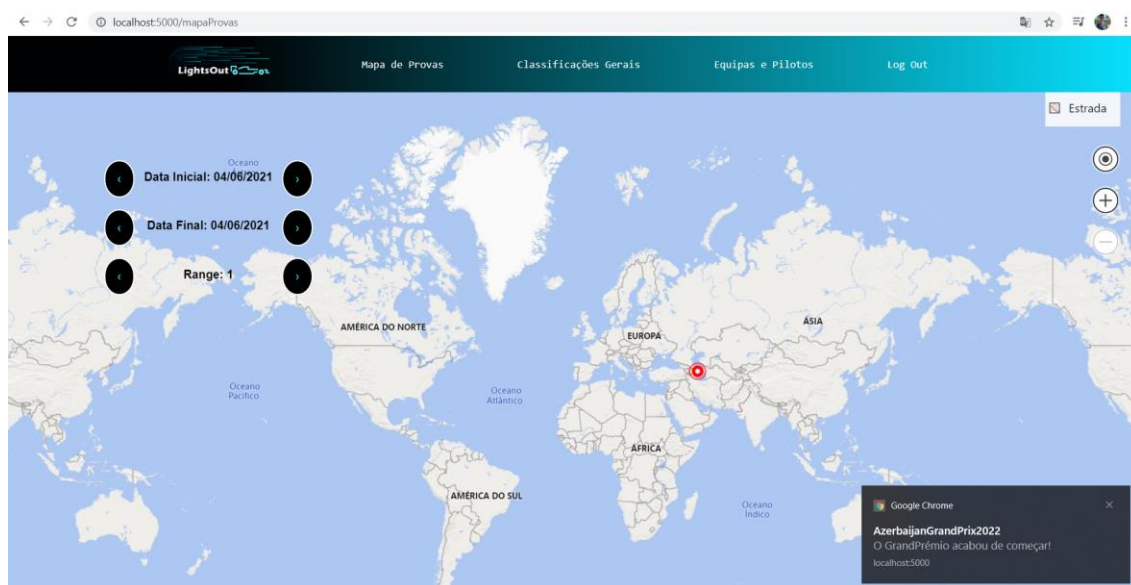


Figura 35: Notificação Começo de Prova

6.2.5 Consultar Classificações Gerais

Relativamente às Classificações Gerais, serão apresentados dois filtros e dois gráficos de barras. O gráfico de barras evolui conforme os valores presentes no filtro, e permite ao Utilizador verificar a evolução temporal dos pontos, quer dos pilotos, quer das equipas. Quando um Utilizador entra na categoria “Classificações Gerais”, o filtro referente à época está predefinido como a época atual e o filtro referente à prova está predefinido para a primeira prova da época de 2021.

O grupo optou também por associar o veículo de cada piloto ao seu respetivo piloto, passando-se o mesmo para a equipa. Apenas serão apresentados veículos para as últimas 3 épocas, visto que não foi possível arranjar os veículos das épocas anteriores, ficando por isso para trabalho futuro tentar arranjar os veículos das épocas anteriores.

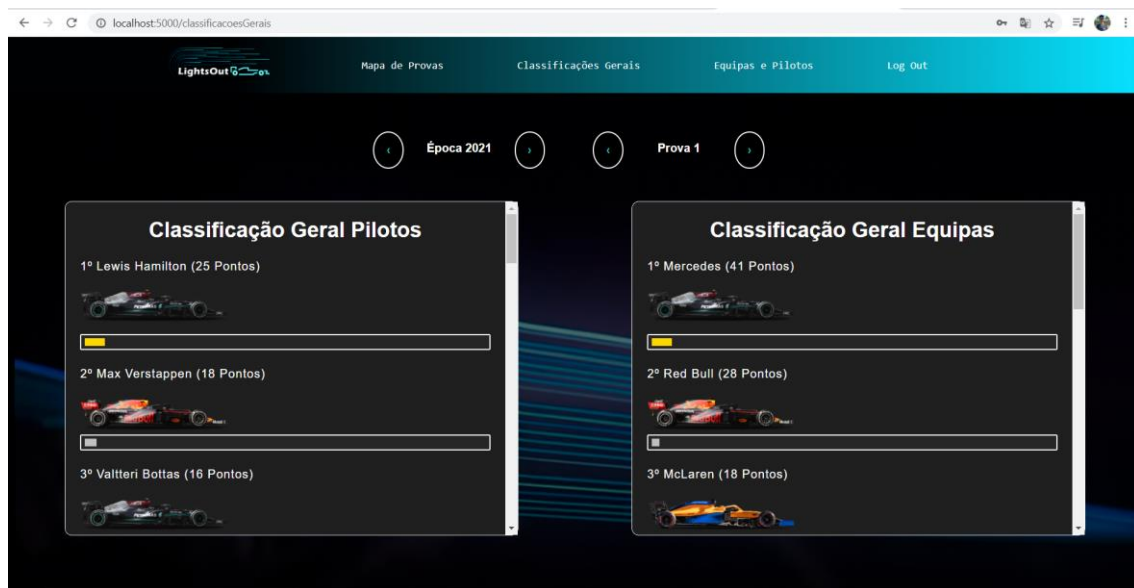


Figura 36: Classificações Gerais

6.2.6 Consultar Equipas e Pilotos

A informação das equipas encontra-se presente na categoria “Equipas e Pilotos”. Nesta será apresentada, numa tabela, informação acerca do nome da equipa, nacionalidade ou veículo, qual o piloto principal e qual o piloto secundário.

The screenshot shows the 'Equipas e Pilotos' page with a navigation bar and a table listing the teams and their drivers for 'Época 2020/2021'. The table has four columns: 'Equipa', 'Veículo', 'Piloto Principal', and 'Piloto Secundário'. The data is as follows:

Equipa	Veículo	Piloto Principal	Piloto Secundário
Mercedes		Lewis Hamilton	Valtteri Bottas
Red Bull		Max Verstappen	Sergio Pérez
McLaren		Lando Norris	Daniel Ricciardo
Ferrari		Charles Leclerc	Carlos Sainz

Figura 37: Equipas e Pilotos

6.2.7 Página Inicial

Como foi referido no levantamento e análise de requisitos, optou-se por apresentar uma página inicial, com um vídeo associado à F1 e um texto simples, apenas com informação do nome do projeto e a que tipo de sistema se refere. Quando um Utilizador está autenticado no Sistema, em vez de aparecer esta informação, irá ser apresentada uma frase de boas-vindas ao Utilizador, tal como se pode verificar na Figura 39.

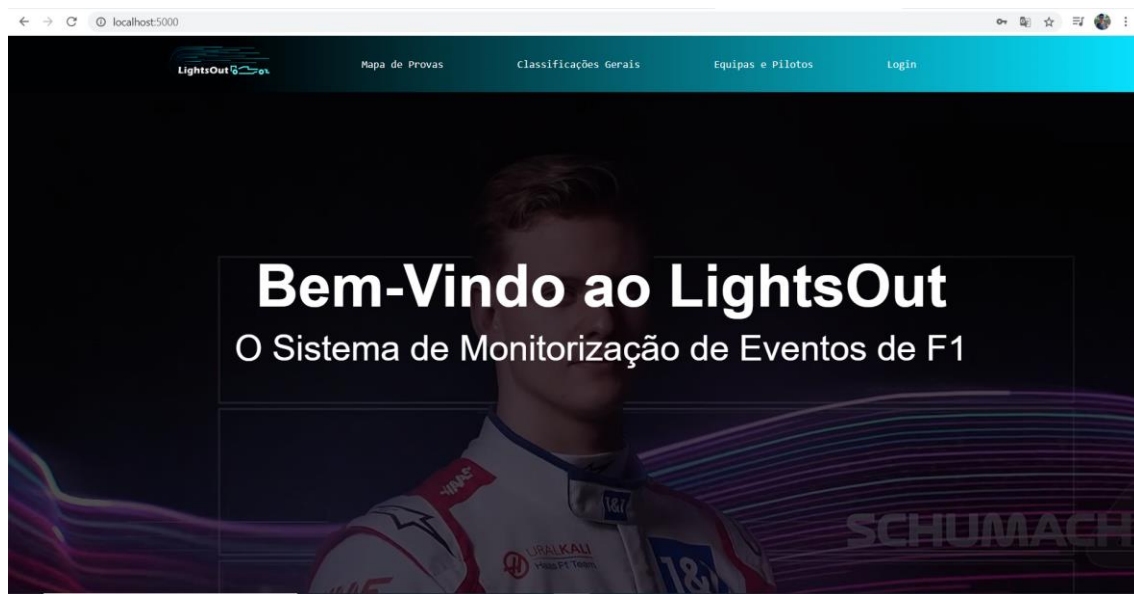


Figura 38: Página Inicial Normal



Figura 39: Página Inicial Autenticado

6.3. Validação do Sistema

Uma vez terminada a fase a fase de implementação, foi necessário validar se o produto final foi de encontro com o idealizado durante a fase de Especificação do projeto. Para tal, foi necessário reunir novamente com os adeptos de F1 que previamente tinham definido os requisitos do projeto.

Relativamente ao mapa de provas, o Carlos Preto encontrou-se bastante satisfeito com o mapa de provas apresentado, bem como com o tipo de filtros desenvolvidos. Por sua vez, o Simão Monteiro apreciou bastante a maneira como a informação referente ao resultado de uma prova é apresentado no ecrã. Posteriormente, o António Santos aprovou as tabelas de classificações gerais e o Jorge Vieira admitiu que a informação das equipas e dos seus pilotos ia de encontro ao esperado, deixando no entanto a sugestão de no futuro, procurar encontrar informações de todos os veículos da F1, bem como incluir os logos de cada equipa.

Uma vez validados todos os requisitos do sistema com os adeptos de F1, foi também necessário verificar se o projeto contemplava todas as funcionalidades desejadas pelos docentes. O nosso mapa de provas permite acompanhar e reportar em tempo (próximo de real) os eventos de F1, fornecendo também o resultado de cada prova e a sua localização no mapa com base em coordenadas de latitude e longitude. Em cada uma das Views do sistema, é possível configurar as diversas fontes de informações através de filtros. Os filtros permitem definir uma agenda específica, ou seja, num determinado espaço temporal, as diferentes informações do sistema.

Nas Classificações Gerais, é possível visualizar, num gráfico de barras, a evolução temporal dos pontos de cada piloto e de cada equipa, de acordo com os valores presentes nos filtros, confirmando assim a inclusão de gráficos no nosso sistema.

Para cada prova que ainda não decorreu, estará disponível a opção de receber uma notificação aquando do início do prova. As notificações são armazenadas na base de dados e quando uma prova começar, é apresentada uma mensagem no ecrã do utilizador, a avisá-lo do início da prova pretendida. Desde que o utilizador tenha o site aberto e esteja autenticado, mesmo que se encontre, por exemplo, no ambiente de trabalho, irá receber na mesma a notificação.

O sistema também acede a uma *API* para obter a informação necessária, e armazena essa mesma informação numa base de dados, de maneira a não ter de se estar sempre a aceder à *API* quando se pretende obter informação.

Por fim, é também necessário verificar se as medidas de sucesso delimitadas para o projeto foram alcançadas. Dito isto, o grupo considera ter desenvolvido um projeto 100% funcional, com uma interface apelativa. Em relação à opinião de outras pessoas relativamente ao nosso projeto, pode-se concluir que a maioria das pessoas a quem foi inquirido acerca do sistema o apreciou bastante, significando então que o nosso projeto é um bom concorrente aos seus rivais, tal como pretendido. Dito isto, o grupo considera então ter alcançado todas as metas delimitadas, concluindo então que se desenvolveu um sistema válido.

6.4. Considerações Finais

O projeto *LightsOut* revelou ser bastante desafiador, não só em termos de gestão de tempo, como também em termos de domínio da *framework .NET Core*.

Apesar de na fase inicial de desenvolvimento de projeto se ter desenvolvido um plano de desenvolvimento, o grupo percebeu rapidamente que algum dos espaços temporais definidos para desenvolvimento das diferentes etapas do projeto estavam longe da realidade. Um exemplo notório deste caso é o desenvolvimento da *Backend* e *Frontend*. Estas tarefas decorreram de 10 de maio até 4 de junho, ou seja, praticamente um mês, enquanto que no plano de desenvolvimento se tinha considerado que estas seriam desenvolvidas em cerca de 10 dias. Alguns dos métodos apresentados nos diagramas de sequência revelaram ser inúteis, porém, a maioria dos métodos implementados foi de encontro com os métodos pretendidos.

De maneira a poder implementar as funcionalidades pretendidas, foi necessário dispendir bastante tempo a ver vídeos e navegar sites que abordassem a *framework .NET Core*. Os seminários de Laboratórios de Informática também se mostraram bastante úteis, na medida que nos providenciaram exemplos práticos.

Tal como tinha sido definido no levantamento de requisitos, foi necessário usar uma *API* para aceder aos dados referentes aos eventos de F1, sendo essa a *Ergast F1*. Uma vez que nunca se tinha trabalhado com *API's*, o processo de aprendizagem, obtenção e filtração de dados demonstrou ser bastante difícil. A criação e conexão à base de dados demonstrou ser das partes mais acessíveis do projeto, visto que previamente se tinha tido contado com a linguagem *SQL*.

O sistema encontra-se dividido em três componentes principais, sendo elas a *FrontEnd*, a *DataAccessLibrary* e a *BackEnd*. A *FrontEnd* apresenta as *Views* do nosso sistema, e utiliza predominantemente *HTML* e *CSS*, invocando métodos que se encontram presentes na *DataAccessLibrary* sempre que seja necessário, de maneira a poder obter dados da base de dados. Sempre que se pretender ir buscar dados à *API* e coloca-los na base de dados, terá de se aceder à *BackEnd*. Nesta também se foi de encontro ao referido nos requisitos do sistema e colocou-se que todos os dias se iriam obter os dados das Qualificações e Resultados mais recentes. Para poder hospedar o projeto num sistema, terá de se previamente ter criado a base de dados. Tal pode ser obtido através da execução do código *SQL* presente no ficheiro *DBStart*, sendo que será posteriormente necessário carregar os dados iniciais para a base de dados.

Apesar de se terem implementado todas as funcionalidades pretendidas, o grupo tem consciência que a maneira como algumas destas funcionalidades foram implementadas poderá não corresponder à solução mais otimizada e eficiente, visto que previamente, nenhum elemento do grupo tinha tido contacto com *.NET*.

Dito isto, e tendo em conta o trabalho realizado em cada uma das fases de desenvolvimento do projeto, o grupo considera ter conseguido desenvolver um sistema relativamente complexo e sem falhas no seu normal funcionamento, considerando também ter desenvolvido um sistema visualmente apelativo e agradável.

7. Conclusões e Trabalho Futuro

Uma vez terminada cada uma das fases de desenvolvimento do projeto, é possível retirar algumas conclusões acerca de cada uma delas.

A primeira fase, apesar de ter sido a menos trabalhosa, revelou ser bastante útil, na medida em que permitiu de limiar uma ideia geral de como seria o nosso sistema de monitorização, bem como organizar o tempo disponível para o desenvolvimento de cada fase. Da primeira fase, também há de destacar a escolha do tema. O facto do tema da *F1* ser bastante apreciado por cada elemento do grupo fez com que a motivação e vontade de suceder no projeto fossem relativamente altas.

A segunda fase incidiu, principalmente, no levantamento e análise de requisitos, bem como no desenvolvimento de diagramas e esquemas que permitiram ao grupo ter uma ideia mais concreta de como teria de se proceder na fase de implementação do projeto.

A especificação em UML do sistema a desenvolver permitiu que o grupo discutisse algumas ideias e rejeita-se outras de maneira a ter um sistema simples mas com as informações úteis necessárias.

Por fim, na terceira fase, após concluída a escrita de código propriamente dita, foi necessário fazer algumas alterações aos diagramas desenvolvidos na segunda fase. Estas alterações foram realizadas, não com o intuito de mudar todos os diagramas previamente desenvolvidos, mas sim adaptar estes ao fluxo e composição do projeto final.

Apesar do grupo considerar ter alcançado todos os objetivos com os quais se propôs na primeira e segunda fase, há alguns aspetos que podem ser melhorados. O sistema de autenticação poderia ser mais complexo e seguro, devendo colocar restrições aos caracteres referentes ao *username* e da *password* de um dado Utilizador. No mapa de provas, sempre que se cria uma notificação e se volta ao mapa, este limpa os filtros previamente definidos pelo Utilizador, ficando apenas com as provas a decorrer no dia atual, sendo por isso necessário ao utilizador redefinir os filtros. Estes dois aspetos referidos ficam então como trabalho futuro.

Dito isto, o grupo encontra-se bastante satisfeito com o resultado final do projeto, considerando até mesmo ter superado todas as expectativas delimitadas nas fases anteriores. O tempo despendido fez com que o grupo ficasse a perceber um pouco mais da *framework .NET*, bem como consolidar ainda mais linguagens como o *HTML* e o *C#*, que certamente trará vantagens no futuro de cada um dos elementos do grupo.

Referências

Formula1, s.d. *TV and Digital Audience figures for 2020*. [Online]
Available at: <https://www.formula1.com/en/latest/article.formula-1-announces-tv-and-digital-audience-figures-for-2020.3sbRmZm4u5Jf8pagvPoPUQ.html>
[Acedido em 15 março 2021]

DriveTribe, s.d. *How David Croft Became the voice of Modern Formula 1*. [Online]
Available at: https://drivetribe.com/p/fast-talker-how-david-croft-became-OC35IcTcRvqNZ0Vsr_3D8g?iid=XmqZuLGxQhSLCQmCTrUYRQ
[Acedido em 16 março 2021]

Syncfusion, s.d. *How do you add Bing Maps to a Blazor application?*. [Online]
Available at: <https://www.syncfusion.com/faq/blazor/general/how-do-you-add-bing-maps-to-a-blazor-application>
[Acedido em 18 maio 2021]

Github, s.d. *brunobmo/Blazor_Course*. [Online]
Available at: https://github.com/brunobmo/Blazor_Course
[Acedido em 22 maio 2021]

Github, s.d. *Append-It/Blazor.Notifications*. [Online]
Available at: <https://github.com/Append-IT/Blazor.Notifications>
[Acedido em 30 maio 2021]

Youtube, s.d. *Blazor Server BackgroundService*. [Online]
Available at: <https://www.youtube.com/watch?v=SfZ64zTQYR8>
[Acedido em 30 maio 2021]

Lista de Siglas e Acrónimos

BD	Base de Dados
DW	Data Warehouse
OLTP	<i>On-Line Analytical Processing</i>
F1	Fórmula 1
API	<i>Application Programming Interface</i>
UML	<i>Unified Modeling Language</i>
.NET	<i>dotNet</i>
HTML	<i>HyperText Markup Language</i>
C#	<i>C Sharp</i>
SQL	<i>Structured Query Language</i>