

Universidade de Trás-os-Montes e Alto Douro

Plataformas interativas de vídeo: uma proposta de implementação

Dissertação de Mestrado em Comunicação e Multimédia



Por:

Miguel Ângelo Ferreira Fonseca

Orientador: Emanuel Peres

Coorientador: Francisco Pereira

Coorientador: António Rio Costa

Vila Real, 2017

Universidade de Trás-os-Montes e Alto Douro

Plataformas interativas de vídeo: uma proposta de implementação

Dissertação de Mestrado em Comunicação e Multimédia

Por:

Miguel Ângelo Ferreira Fonseca

Orientador: Emanuel Peres

Coorientador: Francisco Pereira

Coorientador: António Rio Costa

Dissertação submetida à
UNIVERSIDADE DE TRÁS-OS-MONTES E ALTO DOURO
para obtenção do grau de
MESTRE
em Comunicação e Multimédia, de acordo com o disposto no
DR – I série–A, Decreto-Lei n.º 74/2006 de 24 de Março e no
Regulamento Geral dos Ciclos de Estudo Conducentes ao Grau de Mestre da UTAD,
2.ª série – Regulamento n.º 658/2016

Vila Real, 2017

Orientação Científica:

Emanuel Peres

Professor Auxiliar do
Departamento de Engenharias da Escola de Ciências e Tecnologia
Universidade de Trás-os-Montes e Alto Douro

Francisco Pereira

Professor Auxiliar do
Departamento de Engenharias da Escola de Ciências e Tecnologia
Universidade de Trás-os-Montes e Alto Douro

António Rio Costa

Coordenador Técnico dos
Serviços de Informática e Comunicações
Universidade de Trás-os-Montes e Alto Douro

“Aproveitem a vida e ajudem-se uns aos outros. Apreciem cada momento. Agradeçam e não deixem nada por dizer. Nada por fazer.”

António Feio

À minha família, amigos e colegas

Resumo

A forma como se vê televisão está a mudar. Nos dias de hoje, apenas com uma conexão à Internet é possível aceder a canais televisivos ou eventos, independentemente do dispositivo que se está a utilizar. Contudo, a comunicação não é alterada: é unidirecional, não existe interação/retorno, tornando o utilizador, sobretudo, num mero espetador.

Por outro lado, vive-se num mundo onde cada vez mais a interação e a colaboração são palavras-chave na sociedade. Surge, então, a ideia de modificar a disponibilização de conteúdos multimédia pela *web* tornando-a num meio interativo proporcionando, simultaneamente, uma experiência colaborativa. Nesse sentido, foram utilizadas tecnologias de *stream* de vídeo em 360° e comunicação em tempo real, pretendendo-se, com isso, evitar que o utilizador seja um mero espetador e passe a ser um elemento ativo e interventivo na emissão, se assim o desejar. Nesta dissertação fez-se um estudo dos requisitos de uma plataforma deste género, foi desenvolvido um protótipo com as características pretendidas e foram realizados dois casos de estudo de forma a perceber o tipo de aplicabilidade que estas tecnologias poderiam ter, assim como a sua aceitação por parte dos utilizadores.

A investigação e o protótipo permitiram perceber ser possível interligar e agregar este conjunto de tecnologias numa só plataforma e os casos de uso foram essenciais na determinação das melhores práticas para este tipo de emissão em direto em 360°, assim como para aferir o elevado grau de aceitação por parte dos espectadores que, através desta plataforma, passam a conseguir interagir, em tempo real, com os programas em emissão. Conclui-se, por isto, que esta abordagem pode constituir-se como um caminho seguro a trilhar para o futuro da televisão, enquanto serviço e experiência multimédia.

Palavras-chave: Colaboração, WebRTC, Vídeo Imersivo, Laravel, Jitsi Meet, Youtube, Facebook.

Abstract

The way we perceive television is changing. Nowadays, it is possible to access TV channels and events with just an Internet connection, regardless of the device used. However, communication does not change: it is unidirectional, there is no interaction/return. The user is, therefore, a mere spectator.

On the other hand, we live in a world where interaction and collaboration are crucial to society. The idea of modifying the availability of multimedia content through the Web becomes relevant, creating an interactive medium while providing a collaborative experience. In this respect, video streaming technologies in 360° and real-time communication were used, in order to prevent the user from being a mere viewer and become an active element in the transmission, if he so wishes.

A study of the requirements of such platform was carried out, a prototype of this platform with these characteristics was developed, and two case studies were conducted in order to understand the type of applicability of these technologies, as well as the acceptance by users.

The research and the prototype have shown that it is possible to link and merge this set of technologies in a single platform, and the two study cases were essential in determining the best practices for this type of 360° live streaming, as well as to assess the high level of acceptance by viewers, who are able to interact in real time with the broadcast programs through this platform. It is therefore possible to conclude that this approach can be a safe way forward to define the future of television as a service and multimedia experience.

Keywords: Collaboration, WebRTC, Immersive Video, Laravel, Jitsi Meet, Youtube, Facebook.

Agradecimentos

Com a conclusão da dissertação, chego ao fim de uma etapa da minha vida. Quero aproveitar esta oportunidade para agradecer aos que contribuíram, de uma forma ou de outra, para a concretização da mesma ao longo destes anos.

Os primeiros agradecimentos vão para os meus orientadores Emanuel Peres, António Costa e Francisco Pereira, pela dedicação, preocupação e disponibilidade que demonstraram ao longo da realização deste trabalho. Agradeço a oportunidade que me deram para ser vosso orientando e a confiança que depositaram em mim ao longo deste tempo.

Quero deixar também aqui uma palavra de agradecimento aos Serviços de Informática e Comunicações (SIC) da Universidade de Trás-os-Montes e Alto Douro (UTAD), por me terem acolhido neste último ano, assim como agradecer pelos recursos disponibilizados para a realização da minha dissertação. A todos os professores da licenciatura e do mestrado de Comunicação e Multimédia com quem fui privando ao longo destes 5 anos, agradeço a partilha de conhecimento e a discussão de opiniões.

Ainda relativamente a professores e a funcionários não-docentes dos SIC, com quem contactei, queria agradecer especialmente a três pessoas que me foram ajudando no último ano, de variadas formas, na realização da dissertação: Alberto Vasconcelos e Armandino Rebelo dos SIC e ao professor José Sousa.

Quero, agora, deixar uma palavra a todos os colegas e funcionários com quem tive o prazer de conviver na UTAD e que me ajudaram na minha evolução. Das pessoas com quem privei, queria destacar algumas que, ao longo destes anos, estiveram mais perto de mim e que mais me apoiaram: Ana Cardoso, Ana Lopes, André Ferreira, Aléxis Silva, António Ventura, Daniela Pinto, Diana de Oliveira, Filipe Carvalho, Márcia Oliveira, Margarida Afonso, Marta Resende, Raquel Santos e Rosaly Martins. Para além destas pessoas, quero, também, agradecer ao grupo das “Pausas” da Residência Além-Rio por aqueles “5 minutos” de descontração ao final do dia, durante estes anos.

Ao longo da dissertação, tive a ajuda de algumas pessoas que foram importantes para a sua realização. À Miriam Cabo e à Filipa Almeida agradeço pelo empréstimo dos dispositivos

Apple, à Sofia Gomes e à casa “The Godfather” pelo empréstimo da Internet, à Universidade FM, mais especificamente ao Luís Mendonça, por ter tornado possível a realização da conversa acerca do Circuito Internacional de Vila Real, e às pessoas que estiveram envolvidas, pela sua disponibilidade em fazer parte desta experiência: Rafael Lobato, Pedro Mendes, Filipe Fernandes, Eduardo Ferreira, Eduardo Santelmo, Ricardo Fortuna e Andreia Pinto.

Deixo aqui um pedido de desculpas e, ao mesmo tempo um agradecimento ao meu grupo de Braga “Friends Like a Boss”. Um pedido de desculpas por andar mais distante, principalmente nestes últimos dois anos e um agradecimento porque, mesmo assim, quando podiam davam uma mão e estavam sempre lá para dar uma palavra de apoio. Deste grupo queria deixar um agradecimento especial à Joana Meneses por já me aturar há cerca de 20 anos e por ter aceite fazer uma breve revisão da escrita, mesmo tendo os seus trabalhos para fazer.

Por último, o mais importante: quero agradecer à minha família no geral, mas principalmente, aos meus pais e à minha irmã. Quero agradecer pela educação que me deram e por me possibilitarem a hipótese de ter seguido para o ensino superior e ter chegado a este ponto da minha caminhada. Quero agradecer, também, pelo apoio que me foram dando ao longo do tempo, principalmente quando via as coisas mais complicadas. Tinham sempre uma palavra de ânimo para dar, mesmo quando não compreendiam bem o que se passava nem o que poderiam fazer para ajudar.

Obrigado a todos e "façam o favor de ser felizes"!

UTAD, Vila Real

Miguel Fonseca

Outubro de 2017

Índice

Resumo	i
Abstract	iii
Agradecimentos	v
Índice de Tabelas	xi
Índice de Figuras	xiii
Glossário, Acrónimos e Abreviaturas	xxi
1. Introdução	1
1.1. Enquadramento	1
1.2. Motivação	2
1.3. Objetivos.....	3
1.4. Organização do documento	4
2. Estado da Arte.....	7
2.1. Colaboração	7
2.1.1. Conceito	8
2.1.2. Colaboração online.....	9
2.1.3. Televisão Social	10
2.2. Vídeo Interativo	11
2.2.1. Conceito	12
2.2.2. Visualização	13
2.2.3. Arquitetura da stream de vídeo 360°	14
2.2.4. Plataformas existentes	16
2.3. WebRTC.....	17
2.3.1. Conceito	17
2.3.2. Arquitetura	19
2.3.3. Protocolos utilizados	20
2.3.4. Plataformas existentes	21
2.4. Frameworks de desenvolvimento web	23

2.4.1.	Conceito	23
2.4.2.	Frameworks de PHP	25
2.4.3.	Comparação entre Frameworks PHP	27
3.	Plataforma de vídeo 360° com integração de tecnologias de interatividade e de colaboração em tempo real.....	31
3.1	Conceção e Modelo	31
3.3.1	Análise de requisitos	32
3.3.2	Casos de uso.....	36
3.3.3	Diagramas de atividades	38
3.3.4	Diagramas de estado	39
3.3.5	Base de dados.....	41
3.3.6	Proposta de Layout	42
3.2	Arquitetura proposta	49
3.3	Implementação.....	50
3.3.1	Configuração dos servidores.....	50
3.3.2	Plataforma Jitsi Meet	53
3.3.3	Framework Laravel.....	54
3.4	Emissão direto 360° de evento	65
3.4.1	Escolha da câmara 360°	65
3.4.2	Encoder	69
3.4.3	Transmissão do direto 360° através do Facebook.....	73
3.4.4	Transmissão do direto 360° através do Youtube.....	75
3.4.5	Problemas.....	78
4.	Casos de estudo	81
4.1	Centro de dados da UTAD.....	81
4.1.1	Equipamento	81
4.1.2	Dados	82
4.1.3	Problemas.....	84
4.1.4	Conclusões do caso de estudo.....	85
4.2	Conversa acerca do 48° Circuito Internacional de Vila Real na Universidade FM	86

4.2.1	Equipamento	87
4.2.2	Dados	88
4.2.3	Problemas.....	93
4.2.4	Conclusões do caso de estudo.....	95
5	Conclusões e trabalhos futuros	97
6.	Referências Bibliográficas.....	99

Índice de Tabelas

Tabela 1. Comparação entre as <i>frameworks</i> PHP Laravel, Symfony e Codeigniter.....	29
Tabela 2. Esta tabela apresenta os cinco pacotes de serviços pensados para a plataforma proposta, bem como uma pequena descrição de cada um deles.	32
Tabela 3. Requisitos funcionais dos Serviços Gerais (P1): funcionalidades a que todos os utilizadores (incluindo os não autenticados) podem aceder.	33
Tabela 4. Requisitos funcionais dos utilizadores registados (P2): funcionalidades a que os utilizadores autenticados podem aceder, além dos do pacote comum P1.....	34
Tabela 5. Requisitos funcionais dos Serviços dos Jornalistas (P3): funcionalidades que os jornalistas podem aceder, além dos do pacote comum (P1).	34
Tabela 6. Requisitos funcionais dos Serviços dos Moderadores (P4): funcionalidades que os moderadores podem aceder, além do pacote comum (P1).	35
Tabela 7. Requisitos funcionais dos Serviços do Administrador (P5): funcionalidades que os administradores podem aceder, além do pacote comum (P1).....	35
Tabela 8. Descrição sumária dos objetivos e dos campos existentes nas tabelas da base de dados.	42
Tabela 9. Comparação dos valores obtidos através da plataforma Facebook após o final do caso de uso “Conversa acerca do 38º Circuito Internacional de Vila Real na Universidade FM” e decorrida uma semana.	92

Índice de Figuras

Figura 1. As diferentes camadas base de uma aplicação onde esteja inserida uma televisão social descritas por Salas e Kalva, 2013.	10
Figura 2. As diferentes formas de ver vídeo 360°. Em cima, a CAVE, em baixo, do lado esquerdo, o HMD e do lado direito, a visualização em browsers. Imagem obtida de Bleumers et al., 2012.	13
Figura 3. Arquitetura apresentada por Ochi et al. para um sistema que permite o stream de vídeo 360°. Imagem obtida de Ochi et al, 2015.	15
Figura 4. Arquitetura de um sistema que permite o stream de vídeo em 360°, proposta por Quax et al, 2013.	16
Figura 5. Arquitetura base da tecnologia WebRTC, tal como apresentada no artigo de Loreto e Romano, 2012.	19
Figura 6. Camadas protocolares do RTCPeerConnection, obtido de Jeengins et al., 2013.	20
Figura 7. As linguagens de programação para a web, que operam do lado do servidor, mais utilizadas no mês de março de 2017. Pode-se constatar que o PHP está presente em 82,6% dos sítios web consultados (W3Techs, 2017).	25
Figura 8. Exemplo do estilo arquitetónico MVC e da respetiva interação entre os seus três elementos, tal como apresentada no trabalho de Pop & Altar, 2014.	26
Figura 9. <i>Frameworks</i> PHP de desenvolvimento para a web mais pesquisadas nos últimos 12 meses. Esta consulta foi realizada no dia 17 de abril de 2017, recorrendo à ferramenta Google Trends.	27
Figura 10. Caso de uso dos serviços gerais, onde é possível ver a relação entre o utilizador comum e as funcionalidades afetas ao pacote P1.	36
Figura 11. Caso de uso para os utilizadores registados, onde é apresentada a relação entre um utilizador autenticado e as respetivas funcionalidades da plataforma.	37
Figura 12. Caso de uso que apresenta a relação entre os utilizadores do tipo Jornalista e as funcionalidades da plataforma com que podem interagir.	37
Figura 13. Caso de uso que apresenta a relação entre os utilizadores do tipo Moderador e as funcionalidades da plataforma com que podem interagir.	37
Figura 14. Caso de uso para o utilizador do tipo Administrador.	38

Figura 15. Diagrama de atividades da do acesso à plataforma, onde é perceptível o procedimento necessário para visualizar a transmissão, em direto, de um evento, bem como a participação nele, caso esteja autenticado.	38
Figura 16. Diagrama de atividades que apresenta as interações necessárias no processo de registo de um utilizador na plataforma, quer em formulário próprio, quer utilizando os dados provenientes de uma rede social.	39
Figura 17. Diagrama de estados de um conteúdo multimédia. Pode assumir o estado público, privado ou em destaque.	39
Figura 18. Diagrama relativo ao tipo e ao estado do utilizador. Um utilizador pode estar ativo ou bloqueado e assumir o papel de Administrador, Jornalista, Moderador ou Utilizador geral.....	40
Figura 19. Diagrama de estado relativo aos comentários dos utilizadores a algum conteúdo multimédia da plataforma. Podem assumir o estado ativo ou bloqueado.....	40
Figura 20. Base de dados desenvolvida para a plataforma de transmissão de vídeo 360° com integração de tecnologias de comunicação para suporte de interação em tempo real.	41
Figura 21. <i>Layout</i> da página principal da plataforma, onde são apresentadas quatro notícias em destaque, as últimas notícias inseridas, as notícias mais lidas e os últimos vídeos inseridos.	43
Figura 22. <i>Layout</i> da página de uma categoria, onde é possível aceder às últimas notícias dessa categoria, às mais lidas e aos últimos conteúdos multimédia inseridos.....	44
Figura 23. <i>Layout</i> da página de uma notícia, onde é possível ver a informação da notícia, os seus comentários, comentar (caso o utilizador esteja autenticado), as notícias mais lidas, as últimas notícias e, ainda, a possibilidade de partilhar a notícia.....	45
Figura 24. <i>Layout</i> da página de um conteúdo multimédia, onde é possível ver informação acerca dele, os comentários já efetuados, submeter um novo comentário (caso o utilizador esteja autenticado), ter acesso aos conteúdos multimédia mais recentes e os mais vistos, bem como a possibilidade de partilhar o conteúdo multimédia.....	46
Figura 25. <i>Layout</i> da página da transmissão em direto, onde pode ser consultada informação relativa à transmissão do evento, é permitida a partilha do evento e, caso o utilizador esteja autenticado, é permitido aceder ao chat, em tempo real, com os restantes utilizadores e com a gestão da emissão.....	47

Figura 26. <i>Layout</i> da página de um jornalista, onde é possível aceder às últimas notícias, às mais lidas e é disponibilizada a hipótese de o utilizador entrar em contacto com o jornalista.....	48
Figura 27. Arquitetura proposta para uma plataforma de emissão de vídeo 360°, com integração de tecnologias de interatividade e de colaboração em tempo real.	49
Figura 28. Ficheiro das interfaces do servidor web do protótipo, com a configuração de rede implementada.	52
Figura 29. Ficheiro de hosts após a alteração, onde é possível ler o endereço e os domínios que estão associados ao servidor web.	52
Figura 30. Página inicial da Jitsi Meet, após a conclusão da sua instalação.	53
Figura 31. Configuração do ficheiro “.env” disponível na pasta raiz do projeto Laravel criado, de forma a ser possível aceder a uma base de dados.	54
Figura 32. Exemplo de um ficheiro de configuração de uma tabela - a relativa aos comentários - realizada através do elemento Migration do Laravel.	55
Figura 33. Algumas das rotas criadas para o protótipo, onde é possível ter alguns exemplos de pares controlador/método previstos para diferentes acessos.	56
Figura 34. Dois tipos de controladores que podem ser criados a partir da consola: à esquerda um controlador já com métodos definidos e à direita um controlador sem métodos definidos.	57
Figura 35. Exemplo de um método desenvolvido para o protótipo. Permite, caso o utilizador esteja autenticado e seja moderador, jornalista ou administrador, acesso às notícias disponíveis no protótipo, assim como a ordená-las por título, categoria ou pelo nome de quem as criou.	57
Figura 36. Exemplo de um modelo - o de utilizadores - desenvolvido para o protótipo.	58
Figura 37. Exemplo de um ficheiro de validação desenvolvido para confirmar a sanidade dos dados, aquando da introdução de uma nova notícia. São, ainda, definidas as mensagens de erro a apresentar ao utilizador, caso estes venham a ocorrer.	59
Figura 38. Associação de uma validação - o relativo às notícias - ao método correspondente.	59
Figura 39. Impressão de uma variável utilizando o “echo” do PHP, em cima, e utilizando a forma que o Laravel utiliza, em baixo. Ambas apresentam o mesmo resultado final.	59
Figura 40. Exemplo de como se declara uma secção numa página-tipo, através da utilização da Blade.	60

Figura 41. Exemplo de um ficheiro de conteúdo, onde é possível ver, na primeira linha, a referência à página-tipo e na restante página, a referência à secção da página-tipo e ao conteúdo a nela inserir.	60
Figura 42. Exemplo do formulário desenvolvido para a criação de uma transmissão, em direto, de um evento, com base no componente Laravel Collective. Poderá, também, ser observada, a integração do formulário no HTML.	61
Figura 43. Modelo referente ao utilizador, após as alterações introduzidas no sentido de combinar os módulos Socialite e Authentication Quickstart. Foi adicionado o campo "facebook_id", sendo que os campos "estado" e "tipo" estão relacionados com os valores acerca do estado e do tipo de utilizador.	62
Figura 44. Migração referente ao utilizador, onde foi necessário adicionar o campo "facebook_id" para guardar o valor de identificação da rede social Facebook do utilizador.	62
Figura 45. Elemento <object> do HTML, em que é possível ver o atributo data com a ligação para a sala da Jitsi criada para o direto. O comando "\$direto->chat" imprime o nome da sala criada para o direto. Caso o utilizador seja, apenas, um visitante, não tem a opção de ter apresentada a sala de conversação.	63
Figura 46. Integração das plataformas de transmissão e de interação, onde é possível, também, ver a opção que existe para esconder a sala de conversação, caso o utilizador assim o entenda.	63
Figura 47. Erro de “syntax error or access violation”, que ocorre ao tentar fazer a migração para a criação das tabelas da base de dados.	64
Figura 48. Alteração necessária ao método "boot" do ficheiro "AppServiceProvider.php" para resolver o problema identificado e apresentado na Figura 47.	65
Figura 49. Configuração necessária para criar uma miniatura a partir de um vídeo colocado no protótipo, utilizando o software "ffmpeg".	65
Figura 50. Imagens adquiridas com a câmara Fly HD, enquanto decorriam os testes para testar a viabilidade de realizar a transmissão 360°, em direto, de um evento.....	66
Figura 51. Imagens onde pode ser verificado que não existe a opção de webcam (1.ª imagem), e em que tanto em Android (2.ª imagem) como em iPhone (3.ª imagem) a versão do firmware é a 1.8.16 e não a necessária 2.1.4. Não foi possível, também, atualizar o firmware da câmara FlyHD.	66

Figura 52. Transmissão realizada com a câmara Fly 4k onde é possível verificar que a imagem é dividida em duas imagens de 180°. Imagem retirada de um comentário de craftyazz, 2017, no fórum da própria empresa, 360fly.	67
Figura 53. Comentário onde é explicado o tipo de saída da câmara Fly HD e o motivo pelo qual não é suportado pelo Facebook e Youtube (potro777, 2016). Tradução: “(...)O atual output não suporta a transmissão em tempo real para o YouTube e o Facebook. (...). É necessário que o ficheiro seja retangular, o que, atualmente, é, apenas, obtido, no pós-processamento.”	67
Figura 54. Fotografia de exterior, com boa luz, captada com a câmara Theta S, no exterior da Residência de Estudantes Além-Rio da UTAD, em Vila Real.	68
Figura 55. Fotografia de interior, com fraca luminosidade, captada com a câmara Theta S, no interior do Pavilhão dos Desportos de Vila Real, durante o XX Sarau de Atividades Gímnicas e Academias.....	68
Figura 56. Fotografia de interior, com boa luminosidade, captada com a câmara Theta S, no Laboratório de Artes Visuais e Multimédia da UTAD, durante uma aula da unidade curricular de Web TV, do Mestrado em Comunicação e Multimédia.	69
Figura 57. À esquerda, uma imagem que apresenta os dois botões que é necessário acionar para colocar a câmara no modo direto e à direita, apresenta-se a imagem da câmara em modo direto.	70
Figura 58. Localização, no OBS, da opção para adicionar um dispositivo de captura de vídeo.	70
Figura 59. Localização, no OBS, onde se seleciona a câmara que se deseja, e, no caso da Theta S, o modo de transmissão: HD ou Full HD.	71
Figura 60. Localização da opção para esticar a imagem da câmara Theta S, de forma que esta ocupe o ecrã todo. Esta transformação não é notada aquando da visualização da transmissão em direto.	72
Figura 61. Ecrã que contém as instruções necessárias para atualizar o firmware da câmara Theta S.	72
Figura 62. Ecrã que surge quando o firmware da Theta S é o mais recente.	73
Figura 63. Página, no Facebook, onde é possível escolher o local de publicação e selecionar que a transmissão é em 360°.	73
Figura 64. Página, no Facebook, onde é possível escolher a descrição e o título da transmissão em direto, assim como aceder à chave de transmissão.	74

Figura 65. Ecrã, no OBS, onde se deve escolher o serviço de transmissão e colocar a respetiva chave, para estabelecer a ligação entre o Facebook e o OBS.	74
Figura 66. Ecrã da transmissão no Facebook, onde é possível ver o direto, o número de utilizadores atuais, quem aderiu, os comentários e responder aos mesmos, assim como terminar o direto.....	75
Figura 67. Página, no Youtube, onde podem ser inserir as informações básicas da transmissão 360°, em direto.	75
Figura 68. Local, nas definições avançadas, da criação da transmissão para o Youtube, onde se deve selecionar que a transmissão será em 360°.	76
Figura 69. Página, no Youtube, onde é possível escolher a taxa de bits e onde é acedida a chave da transmissão para colocar no OBS.	76
Figura 70. Ecrã, no OBS, onde se deve escolher o serviço de transmissão e colocar a chave de transmissão de forma a fazer a ligação entre o Youtube e o OBS.	77
Figura 71. Local onde é possível aceder a pré-visualização da transmissão no Youtube de forma a confirmar se está tudo de acordo com o pretendido.	77
Figura 72. Local onde pode dar-se início à transmissão para o Youtube, após a opção estar disponível.....	77
Figura 73. Local onde é possível parar a transmissão no Youtube.	78
Figura 74. Suportes criados para a Tetha S: o primeiro criado a partir de esferovite, o segundo aproveitando um copo de plástico e o terceiro a partir de um mini-tripé.	78
Figura 75. Suporte usado para a câmara Theta S, no primeiro caso de uso, utilizando um tripé de luzes de estúdio.	79
Figura 76. Ecrã após a câmara se desligar, devido ao aquecimento. À esquerda, em transmissão Full HD e à direita transmissão em HD.	79
Figura 77. Imagem que retrata a passagem da transmissão primeiro por uma mesa de transmissão e só depois para um computador. Nela, também pode ser visto que a imagem que chega ao codificador é circular e não retangular.	80
Figura 78. Disposição do equipamento utilizado na transmissão do Centro de Dados da UTAD.	82
Figura 79. Publicitação da transmissão do Centro de Dados, assim como a publicação da hiperligação, em direto, no Youtube, após resolução do problema encontrado com a plataforma Jitsi.....	83

Figura 80. Alcance da publicação (252 pessoas) do caso de uso sobre o Centro de Dados da UTAD.	83
Figura 81. Fotografia final dos intervenientes na conversa acerca do 48º Circuito Internacional de Vila Real. Da esquerda para a direita: Andreia Pinto, Filipe Fernandes, Eduardo Ferreira, Ricardo Fortuna, Rafael Lobato, Pedro Mendes, Eduardo Santelmo e Luís Mendonça.....	86
Figura 82. Disposição do material utilizado na transmissão da conversa relativa ao 48º Circuito Internacional de Vila Real.	87
Figura 83. Suporte usado para a transmissão do direto da conversa acerca do 48º Circuito Internacional de Vila Real na Universidade FM.....	88
Figura 84. Exemplo de uma das publicidades criadas, pela Universidade FM, de forma a publicitar a transmissão da conversa acerca do 48º Circuito Internacional de Vila Real.	88
Figura 85. Algumas das partilhas da publicidade referente a transmissão da conversa acerca do 48º Circuito Internacional de Vila Real na Universidade FM.	89
Figura 86. Algumas das entradas no direto, em que podemos ver a presença do Presidente da Câmara Rui Santos e do Piloto Manuel Fernandes. Podem, ainda, ser observadas algumas das partilhas efetuadas.	89
Figura 87. Alguns comentários mais descontraídos, que permitiram criar um ambiente mais familiar.....	90
Figura 88. Extrato de perguntas realizadas aos convidados sugeridas pelos espetadores. À esquerda, perguntas realizadas a partir da caixa de comentários da transmissão, e à direita, perguntas realizadas a partir de mensagens privadas recebidas ao longo da transmissão.....	91
Figura 89. Algumas das reações ao longo da transmissão. Curiosidade para o segundo comentário, que foi uma correção, por parte de uma pessoa ligada ao traçado do percurso, acerca do comprimento total do percurso.	91
Figura 90. Imagem referente às interações obtidas após cerca de uma semana depois da transmissão, onde é possível consultar o número do alcance, visualizações, reações, comentários e partilhas.	92
Figura 91. À esquerda, a distância que tinha que ser percorrida pelo cabo de rede de forma a ligar o router ao computador. À direita, os cabos de rede e um Hub fornecidos pelos SIC, para tentar solucionar o problema.	94

Figura 92. Falha de rede na transmissão para o Facebook, sendo que, pouco tempo depois, a rede foi restabelecida e a transmissão voltou ao ativo.	94
--	----

Figura

93. Misturador de Áudio do OBS onde é possível ver que apenas o som vindo do Desktop está ativo.	95
---	----

Glossário, Acrónimos e Abreviaturas

Lista de Acrónimos

Sigla	Expansão
API	<i>Application Programming Interface</i>
AVC	<i>Advanced Video Coding</i>
CAVE	<i>Cave Automatic Virtual Environment</i>
CDN	<i>Content delivery network</i>
CRUD	<i>Create, Read, Update e Delete</i>
Full HD	<i>Full High Definition</i>
HD	<i>High Definition</i>
HDMI	<i>High-Definition Multimedia Interface</i>
HMD	<i>Head-Mounted Display</i>
HMVC	<i>Hierarchical Model-View-Controller</i>
HTML	<i>HyperText Markup Language</i>
ICE	<i>Interactive Connectivity Establishment</i>
IETF	<i>Internet Engineering Task Force</i>
IP	<i>Internet Protocol</i>
MPEG	<i>Moving Picture Experts Group</i>
MVC	<i>Model-View-Controller</i>
NAT	<i>Network Address Translation</i>
OBS	<i>Open Broadcaster Software</i>
P2P	<i>Peer-to-Peer</i>
PHP	<i>Hypertext Preprocessor</i>
RAD	<i>Rapid Application Development</i>
RTCP	<i>RTP Control Protocol</i>
RTP	<i>Real Time Transport Protocol</i>
SCTP	<i>Stream control Transmission protocol</i>
SDP	<i>Session Description Protocol</i>
SIC	<i>Serviços Informáticos e Comunicações</i>
SO	<i>Sistema Operativo</i>
SRTP	<i>Secure Real-time Transport Protocol</i>

STUN	<i>Session Traversal Utilities for NAT</i>
TURN	<i>Traversal Using Relays around NAT</i>
UDP	<i>User Datagram Protocol</i>
URL	<i>Uniform Resource Locator</i>
UTAD	<i>Universidade de Trás-os-Montes e Alto Douro</i>
W3C	<i>World Wide Web Consortium</i>
WebRTC	<i>Web Real-Time Communication</i>

Lista de Abreviaturas

Sigla	Expansão
2D	<i>2 dimensões</i>
App	<i>Aplicação</i>
e.g.	<i>Por exemplo</i>
et al.	<i>E outros (autores)</i>
etc.	<i>Etecetera, outros</i>
Web TV	<i>Web Television</i>

1. Introdução

Neste primeiro capítulo é feito o enquadramento da dissertação, apresentada a motivação para a sua realização e quais os seus objetivos. É descrita, ainda, a organização deste documento.

1.1. Enquadramento

A forma como se vê a televisão está a mudar. Já não é uma simples caixa pousada numa mesa que recebe sinais, por cabo ou antena, e nos apresenta um vídeo. Nos dias de hoje, basta ter uma ligação a Internet para poder aceder a canais televisivos ou eventos, que podem ser considerados como conteúdos multimédia, seja a partir de um computador ou de dispositivos móveis. Mas, em ambos os casos, a forma de comunicar não se altera: é unidirecional, o que significa que não existe possibilidade de interação, em tempo real, entre a entidade que emite os conteúdos multimédia e quem os recebe, tornando estes últimos, sobretudo, em espectadores. Com o aparecimento do vídeo 360°, e, mais concretamente, com a disponibilização, em direto, de conteúdos multimédia 360°, o espetador pode assumir outro papel, na medida em que pode, pelo menos, através do dispositivo de visualização que está a utilizar, interagir com a emissão e selecionar a perspetiva através da qual pretende visualizar os conteúdos multimédia.

A interação e a colaboração *online* entre indivíduos têm vindo a assumir relevância crescentes na sociedade (Apostolopoulos et al., 2012; Swan, Shen, & Hiltz, 2006). Com efeito, têm surgido inúmeras tecnologias e outras tantas plataformas *open source* e/ou proprietárias, que permitem

aos utilizadores interagirem uns com os outros de diversas formas (e.g. voz, áudio, vídeo, troca de ficheiros) (Djulovic & Duric, 2015). Uma dessas tecnologias, que assume particular relevância no âmbito deste trabalho, é o Web Real-Time Communication (WebRTC). O seu objetivo é simplificar a comunicação vídeo, em tempo real, entre dois ou mais *browsers*. Para o efeito, não é necessária a instalação de *plugins* ou quaisquer outras aplicações, mas tão somente a cedência do acesso, por parte de cada utilizador, à sua câmara de vídeo e ao microfone.

É nesse contexto que surge a ideia de desenvolver uma plataforma que permita disponibilizar conteúdos multimédia (vídeo 360°) e que, simultaneamente, possibilite a interação, em tempo real, entre utilizadores e a emissão em direto. Pretende-se assim, alterar o papel do utilizador, tornando-o um elemento ativo e colaborativo na experiência multimédia em direto e não somente um mero espectador, como atualmente é. Por exemplo, permitir, com mecanismos de moderação e de gestão implementados, que os utilizadores que recebam os conteúdos multimédia em direto possam questionar eventuais palestrantes, intervir em debates, comunicar com artistas depois de um espetáculo, comunicar com outros utilizadores durante a emissão, entre outros.

A plataforma que se pretende conceber, propor e desenvolver, no âmbito deste projeto terá, assim, de integrar estas tecnologias, bem como implementar ferramentas de gestão de conteúdos, de utilizadores e de interação. Propõe-se uma alteração no paradigma atual da receção de conteúdos multimédia, permitindo ao espetador de hoje tornar-se num elemento ativo e interventivo nas emissões em direto do amanhã.

1.2. Motivação

Os conteúdos multimédia hoje em dia são muito importantes e utilizados, em diversos sectores da sociedade. As empresas, independentemente da sua área de ação, usam-nos de forma a conseguir cativar clientes, demonstrar produtos, fazer publicidade, reivindicar ideias, etc. Estes conteúdos podem conter diversos elementos - som, imagem, vídeo, animação, gráficos e textos – e serem utilizados de forma simples ou compostos. Por exemplo, num telejornal de uma cadeia televisiva é utilizado o som e o vídeo para transmitir a informação para o espetador de forma eficiente. As empresas já não pretendem, somente, campanhas publicitárias físicas, mas

apostam, cada vez mais, na disponibilização dos seus conteúdos na Internet - principalmente nas chamadas redes sociais - pois consideram que é a forma mais eficaz e eficiente de atingir os seus públicos-alvo, concretamente clientes e potenciais clientes.

Mas, e independentemente do tipo de conteúdo multimédia produzido, nem sempre o cliente/espetador fica informado da melhor forma, ou então fica com curiosidade e sente a necessidade de saber algo mais. Para isso, é importante que sejam desenvolvidas ferramentas de forma a que o cliente/espetador possa comunicar, diretamente, com uma dada empresa ou entidade. Concretamente, no caso de eventos que estejam a acontecer em direto - entrevistas, telejornais, palestras, etc – é importante que os espetadores possam ter um papel mais ativo, através de meios que lhes permitam interagir, em tempo real, uns com os outros e com a entidade emissora, para que possam colocar dúvidas e obterem as respostas que pretendem.

Com o estudo de plataformas de comunicação em tempo real já existentes e com o desenvolvimento de uma plataforma que permita a interação, em tempo real, com transmissões em direto, pretende-se adquirir conhecimento e experiência nestas ferramentas, potenciar a mudança progressiva do que é hoje o papel de espectador de eventos em direto para uma situação de maior interação com o restante público e a própria emissão, assim como lançar a base para possíveis investigações futuras nesta área.

1.3. Objetivos

O desenvolvimento deste trabalho tem como objetivo principal estudar os conceitos e as tecnologias relacionadas com a colaboração e a interação *online* entre indivíduos e entre indivíduos e entidades, assim como a disponibilização de conteúdos multimédia interativos e a comunicação vídeo, em tempo real, pela web. Pretende-se, ainda, demonstrar que é possível agregar um conjunto destas tecnologias numa plataforma web única e disponibilizar as suas valências, em simultâneo, aos seus utilizadores. Assim, o trabalho foi organizado da seguinte forma:

- Rever o estado da arte relativo aos temas e às tecnologias referidas, de modo a estruturar uma plataforma que os integre;

- Definir uma arquitetura para uma plataforma web responsável pela gestão de *streams* de vídeo 360° com ferramentas complementares de interatividade e colaboração;
- Desenvolver, integrar e implementar uma plataforma de gestão das transmissões de conteúdos multimédia, em direto, via web;
- Desenvolver mecanismos que permitam uma interação, em tempo real, entre utilizadores e entre utilizadores e entidade emissora, durante uma transmissão de conteúdos multimédia;
- Descrever, pormenorizadamente, o processo a utilizar para realizar estas emissões, assim como uma lista de eventuais problemas que possam surgir e a forma de os resolver.
- Realizar casos de estudo para aferir a robustez da solução desenvolvida, os níveis de aceitação e de satisfação dos utilizadores e concluir acerca da viabilidade da plataforma proposta.

1.4. Organização do documento

Esta dissertação encontra-se organizada em cinco capítulos. Para além deste capítulo 1, no capítulo 2 é apresentado um levantamento do estado da arte relativo a alguns conceitos e algumas tecnologias, nomeadamente: colaboração e interação *online*, WebRTC, vídeo 360° em direto e *frameworks* PHP para o desenvolvimento na web. São analisados e descritos os conceitos inerentes a cada um deles, as vantagens e as desvantagens e as arquiteturas que suportam cada tecnologia. Pretendeu-se, assim, conhecer quais as opções mais indicadas para a especificação e o desenvolvimento da plataforma pretendida neste trabalho.

No capítulo 3 é feita a planificação do sistema e deixadas indicações para o seu desenvolvimento. Este capítulo inclui a análise de requisitos, casos de uso, diagramas de atividades, diagramas de estado, planificação da base de dados, alguns *mockups* realizados e a arquitetura proposta. Inclui, também, a descrição detalhada de cada etapa necessária à implementação do protótipo, bem como a solução para os problemas e questões que foram surgindo do processo

No capítulo 4 encontram-se descritos os casos de estudo desenvolvidos, nomeadamente o seu contexto, o equipamento utilizado, os problemas que foram surgindo durante o processo e como foram resolvidos, os dados obtidos e algumas conclusões.

No capítulo 5 são apresentadas as conclusões do trabalho desenvolvido e a evolução que poderá ser feita para trabalho futuro.

2. Estado da Arte

Neste capítulo são abordados conceitos e tecnologias mais diretamente relacionadas com a disponibilização de conteúdos multimédia interativos - nomeadamente o vídeo 360° - através de uma plataforma web que permita, em tempo real, a colaboração dos seus utilizadores na emissão.

2.1. Colaboração

Neste subcapítulo começa-se por apresentar os conceitos de colaboração e de colaboração *online*. Pretende-se, ainda, descrever as contribuições que a denominada televisão (TV) social aportou para a colaboração *online* e, por fim, dar o exemplo de algumas das plataformas de colaboração *online* de código-aberto existentes.

A comunicação está presente em toda a sociedade, desde os meios de comunicação considerados mais tradicionais (e.g. rádio, televisão e jornais), até aos meios mais atuais, suportados na web, como, por exemplo, videoconferências, videochamadas, *chats* instantâneos, email, entre outros. A comunicação desempenha um papel absolutamente fundamental no funcionamento da sociedade como a conhecemos, destacando-se, no contexto deste trabalho, o contributo essencial no processo de aprendizagem e na construção de conhecimento (Lock & Johnson, 2015). É, também, crítica no processo de colaboração, que pode ser definido como

uma filosofia onde existe comunicação entre pelo menos dois indivíduos e estes trabalham para um objetivo comum (Laal & Ghodsi, 2012; Lock & Johnson, 2015; Swan, Shen, & Hiltz, 2006). As plataformas de colaboração *online* vieram permitir comunicar, trocar ideias e realizar tarefas sem existir uma limitação geográfica e/ou temporal (Samuel, 2015).

2.1.1. Conceito

De acordo com Lock & Johnson (2015) existe uma certa similaridade entre o conceito de trabalho de grupo, cooperação e colaboração, mas, na verdade, cada um é único. Com efeito e dando como exemplo o processo de ensino-aprendizagem, enquanto o trabalho de grupo reúne alunos para que estes comuniquem sobre um dado assunto, a função de interatividade é indefinida, pois embora se trate de um trabalho em grupo, esta nem sempre ocorre. No que respeita à cooperação, é considerada um estilo de trabalho em que os alunos dividem tarefas e onde cada um realiza a sua parte. No final, a tarefa é finalizada com a junção de todas as partes realizadas de forma individual. Nas definições apresentadas, cada indivíduo trabalha no sentido de resolver parte do problema ou realizar parte da tarefa, o que resulta em que cada participante tenha conhecimento profundo, apenas, da sua contribuição (Lock & Johnson, 2015).

A colaboração é uma atividade complexa que envolve um esforço individual e de grupo (Swan et al., 2006) e que ocorre sempre que dois ou mais indivíduos trabalham para atingir um objetivo comum, através da partilha de conhecimentos, aprendizagem e construção de um consenso (Laal & Ghodsi, 2012; Lock & Johnson, 2015; Swan et al., 2006).

Ainda segundo Lock & Johnson, 2015, existem três elementos considerados importantes no processo de colaboração, a saber:

- Participação: a participação deve ser igual entre os indivíduos;
- Interação: os indivíduos devem ser ativos, criando uma discussão dinâmica;
- Síntese: o produto desenvolvido é representativo da síntese de ideias e o resultado da interação entre as pessoas.

Para que a colaboração entre indivíduos tenha sucesso, Swan et al, 2006 afirmam que existem alguns comportamentos que devem ser considerados:

- Dar e receber ajuda/assistência;

- Trocar recursos e informações;
- Explicar ou elaborar informações;
- Partilhar conhecimentos com outros;
- Dar e receber *feedback*;
- Desafiar a outras contribuições.

No que respeita às vantagens que o processo de colaboração pode ter, de acordo com Laal & Ghodsi, 2012 e Frick, 2015, por exemplo:

- Construção de uma compreensão da diversidade entre pessoas;
- Desenvolvimento de uma comunidade de aprendizagem;
- Redução da ansiedade dos indivíduos;
- Promoção do pensamento crítico;
- Envolvimento dos indivíduos no processo de aprendizagem.

2.1.2. Colaboração *online*

A colaboração *online* surge com a disseminação da Internet: os indivíduos interagem uns com os outros, de forma digital, no sentido de identificar e propor ideias para a resolução de problemas, para a construção de planos e/ou para desenvolver explicações para fenómenos (Lock & Johnson, 2015). Este tipo de colaboração é visto como mais equitativo e democrático que a colaboração tradicional, pois permite voz ativa a todos os participantes (Swan et al., 2006).

No entanto, a colaboração *online* não está focada no uso de tecnologia para completar uma tarefa: esta é usada para permitir a partilha de informações e a construção/manutenção de ideias ao longo do desenvolvimento de um dado projeto (Lock & Johnson, 2015).

Para além das vantagens destacadas no subcapítulo anterior, a colaboração *online* também permite, de acordo com Majumdar, 2015 e Samuel, 2015, por exemplo:

- Evitar custos com viagens e estadias para poder trabalhar com outros indivíduos;
- Poupar tempo nas deslocações e aumentar a produtividade utilizando o tempo não gasto;
- Aceder rapidamente a informação;
- Ter acesso mais rápido aos peritos de dada área e/ou a colegas de trabalho;

2.1.3. Televisão Social

Com o ritmo do dia a dia, nem sempre é possível reunir os amigos e a família para ver televisão juntos. Assim, surge o conceito de televisão social, que tenta simular a mesma experiência, permitindo aos utilizadores interagirem enquanto estão a visualizar os conteúdos multimédia, através da criação de locais de comunicação, em tempo real, utilizando texto, voz ou vídeo (Salas & Kalva, 2013). A televisão social pode ser definida, assim, como qualquer tecnologia que permita interação social, remota ou não, enquanto se assiste a conteúdo televisivo (Salas & Kalva, 2013; Shin, 2013).

Desta forma, é possível que os utilizadores sintam a presença virtual uns dos outros, pois são criadas comunidades *online* onde existe interação entre eles, dando, assim, a ilusão ou sentimento de inclusão. A televisão social é aceite pelos utilizadores como uma ferramenta de comunicação, de colaboração e de entretenimento. A comunicação constante entre utilizadores e a possibilidade de visualização dos mesmos conteúdos em simultâneo, apesar de estarem geograficamente em regiões diferentes, são alguns dos fatores que motivam a utilização da televisão social. No entanto, é necessário ter cuidado, pois aplicações sociais em excesso podem empobrecer a experiência e a visualização dos conteúdos com qualidade (Shin, 2013).

Segundo Salas & Kalva, 2013, as aplicações de televisão social podem ser divididas em três camadas (Figura 1).

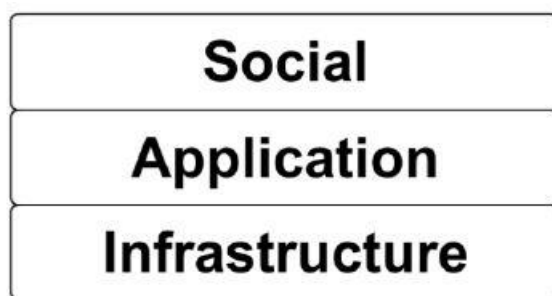


Figura 1. As diferentes camadas base de uma aplicação onde esteja inserida uma televisão social descritas por Salas e Kalva, 2013.

A camada Social é constituída por dois componentes: a interação social e o comportamento do utilizador. Para que este género de aplicação não seja intrusivo e simultaneamente incentive a interação com outros utilizadores, é essencial conhecer o comportamento de um dado utilizador

e a forma como este interage com os restantes (Shin, 2013). Essa interação pode ser considerada síncrona, quando ocorre durante o evento, ou assíncrona, quando ocorre antes ou depois de um evento.

No que respeita à camada de Aplicação, esta utiliza as ferramentas e as interfaces de programação de aplicações (do inglês Application Programming Interface - API) disponibilizados pela camada de Infraestrutura para desenvolver sistemas que vão apoiar a camada Social. É importante fornecer ao utilizador opções para que ele se sinta confortável na aplicação, principalmente no que toca à sua privacidade, pois podem surgir situações em que ele sinta necessidade de guardar maior reserva.

Por fim, a camada de Infraestrutura suporta a operação das restantes, entregando o conteúdo e possibilitando a comunicação entre os utilizadores.

A tecnologia WebRTC, que será apresentada com maior detalhe num subcapítulo posterior, permite a comunicação vídeo, em tempo real, entre *browsers*, via Internet, sem a necessidade de ver instalados *plugins* adicionais (Salas & Kalva, 2013). O WebRTC vem, assim, facilitar a utilização das televisões sociais.

2.2. Vídeo Interativo

Neste subcapítulo apresenta-se o conceito de vídeo interativo, os procedimentos necessários à implementação da sua transmissão em tempo real e algumas ferramentas disponíveis para a sua visualização.

Alguns dos vídeos desenvolvidos atualmente são captados a partir de diferentes perspetivas, recorrendo à utilização, por exemplo, de mais do que uma câmara (e.g. filmes, reportagens). Contudo, quem visualiza esses conteúdos mantém o papel de mero espectador, sem controlo efetivo, por exemplo, sobre a perspetiva através da qual pode visualizar o conteúdo multimédia. Com o vídeo interativo 360°, o espectador tem a possibilidade de selecionar a perspetiva que mais lhe interessa dos conteúdos emitidos, o que permite, também, a sensação de presença na cena emitida.

2.2.1. Conceito

No vídeo mais convencional as imagens são captadas, editadas e só depois emitidas, fazendo com que o espectador tenha uma experiência uniforme, limitada e pobre, pois não tem liberdade de mudar a sua perspectiva (Kimata et al., 2011; Neng & Chambel, 2010). Já o vídeo 360°, também chamado omnidirecional, panorâmico, esférico ou *surround*, permite ao espectador interação, nomeadamente ao permitir a alteração da perspectiva através da qual visiona o conteúdo multimédia (Bleumers, Van den Broeck, Lievens, & Pierson, 2012).

Com efeito, o vídeo interativo 360° oferece um produto mais rico e a sensação de estar num local diferente no mundo real (Gaemperle, Seyid, Popovic, & Leblebici, 2014; Kimata et al., 2011; Neng & Chambel, 2010; Quax, Liesenborgs, Issaris, Lamotte, & Claes, 2013), proporcionando uma experiência imersiva ao indivíduo (Ochi et al., 2014; Quax et al., 2013), por oposição à Realidade Virtual, onde o indivíduo é transportado, efetivamente, para uma realidade diferente daquela que vive (Gaemperle et al., 2014).

A principal característica deste tipo de vídeo é a de permitir que o indivíduo tenha controlo sobre a perspectiva segundo a qual quer experienciar o conteúdo multimédia - seja através do movimento de rotação da cabeça ou utilizando algum dispositivo para esse efeito - podendo, assim, escolher a que mais lhe interessa (Bleumers et al., 2012).

Segundo Quax et al., 2013, Bleumers et al., 2012 e Gaemperle et al., 2014, algumas das aplicações do vídeo 360° são:

- Segurança;
- Eventos desportivos;
- Concertos;
- Debates;
- Videoconferências;
- Turismo virtual;
- Documentários;
- *Reality shows*;
- Noticiários;

No entanto, esta tecnologia traz alguns desafios, dos quais se destaca:

- 1) A utilização de outros sentidos, como o olfato ou a audição, no dia-a-dia dos indivíduos (Kimata et al., 2011; Ochi, Niwa, Kameda, Kunita, & Kojima, 2015);
- 2) O tamanho de um vídeo 360° pode tornar-se um obstáculo ao seu envio e/ou transmissão (Neng & Chambel, 2010; Ochi et al., 2015).

2.2.2. Visualização

Atualmente, existem três formas principais de visualizar este tipo de conteúdo multimédia (Figura 2):



Figura 2. As diferentes formas de ver vídeo 360°. Em cima, a CAVE, em baixo, do lado esquerdo, o HMD e do lado direito, a visualização em *browsers*. Imagem obtida de Bleumers et al., 2012.

- Cave Automatic Virtual Environments (CAVE), onde o indivíduo está rodeado de monitores de grandes dimensões, que apresentam os conteúdos multimédia. Nesta situação, o utilizador tem, apenas, de olhar para a área que mais lhe interessa, através de um movimento de rotação da sua cabeça (Apostolopoulos et al., 2012; Bleumers et al., 2012; Kimata et al., 2011);
- Head-Mounted Display (HMD), em que o conteúdo multimédia é apresentado ao utilizador através de óculos. Com base no movimento da cabeça e num sensor de orientação do HMD é possível modificar a perspetiva através da qual o conteúdo é visualizado (Bleumers et al., 2012; Ochi et al., 2014);

- *Browsers*, 2D, que permitem, de forma nativa (i.e. sem instalação de *plugins*) e com base na linguagem HyperText Markup Language (HTML) versão 5, a integração de conteúdos multimédia. Mais concretamente, o elemento “*canvas*” possibilita a alteração da perspetiva do conteúdo multimédia disponibilizado no elemento “vídeo”, permitindo, assim, ao utilizador, gerir a forma como pretende visualizar o vídeo (Quax et al., 2013).

2.2.3. Arquitetura da *stream* de vídeo 360°

A transmissão de vídeos *online* tem aumentado nos últimos anos, tornando-se muito popular (Djulovic & Duric, 2015; Rhinow, Veloso, Puyelo, Barret, & Nuallain, 2014). Num estudo recente, foi previsto que, com o aumento da largura de banda para o consumidor, haja um grande crescimento na transmissão de vídeo *online* e, por isso, num futuro próximo, grande parte do conteúdo na Internet seja vídeo. Isto deve-se, não apenas pelo aumento da procura, mas também porque, o vídeo, consome mais largura de banda que outro tipo de conteúdo multimédia (Neng & Chambel, 2010; Rhinow et al., 2014).

Com o vídeo 360° surge a possibilidade de apresentar um produto ainda mais rico e com mais informação para ser apresentada (Neng & Chambel, 2010). O *stream* deste tipo de vídeo consiste, sensivelmente, em 3 partes: síntese do vídeo, codificação e *streaming* (Kimata et al., 2011). Ochi, em um dos seus artigos (Ochi et al., 2015) descreve a arquitetura de um sistema que suporte a transmissão de vídeo 360°. Nessa arquitetura (Figura 3) o vídeo é, em primeiro lugar, capturado, em formato retangular, com uma câmara 360. Em seguida, é feita uma cópia do vídeo, em baixa resolução. O vídeo é, então, dividido em vários sobrepostos, de forma a abranger toda a área que se pretende apresentar. Após essa divisão, os vídeos são codificados em altas e baixas resoluções utilizando o *codec* H.264 (MPEG-4 AVC). Através de um HMD, o utilizador seleciona a perspetiva que pretende visualizar e é, então, procurado o respetivo vídeo com maior resolução. O utilizador recebe quer o vídeo de menor quer o de maior resolução, para que não existam lacunas na visualização quando o HMD solicitar o vídeo de maior resolução. Esta técnica permite reduzir a largura de banda necessária à transmissão dos vídeos, pois reduz a transmissão de maior qualidade à área de visualização naquele momento. Nestas experiências, de forma a obter um ambiente mais próximo do real, foi, ainda, usado um microfone do tipo *shotgun*, de forma a captar o som ambiente (foi colocado debaixo da câmara para não bloquear a visão).

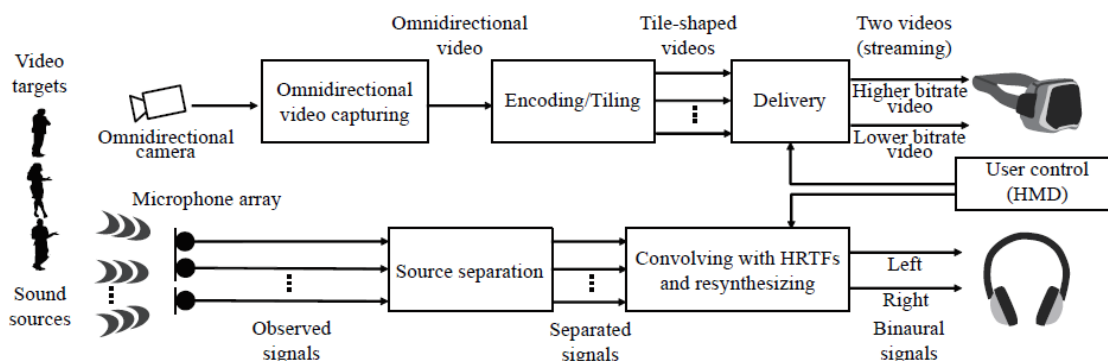


Figura 3. Arquitetura apresentada por Ochi et al. para um sistema que permite o *stream* de vídeo 360°. Imagem obtida de Ochi et al, 2015.

Já Quax et al., 2013, apesar de manterem uma arquitetura idêntica à apresentada por Ochi et al., 2015, determinam que o *stream* é realizado para a web (Figura 4). Com efeito, é utilizado uma câmara que é capaz de fazer *stitching* de múltiplos vídeos num único *frame* de saída, de formato retangular, que é colocado num *encoder* de difusão para a web. Para reduzir a latência entre o *encoder* e a câmara, e de forma a fornecer um vídeo com qualidade, é utilizada uma ligação High Definition Multimedia Interface (HDMI) para fazer essa transmissão.

O sistema é capaz de fornecer versões de alta e baixa qualidade. O de alta qualidade é enviado para uma estrutura onde é codificado, processado e guardado. Os vídeos com menor qualidade são codificados e depois processados de forma a gerar pedaços de vídeo. Uma vez criado, o pedaço de vídeo é guardado e feito o *upload* para o CDN (*Content delivery network*) assim que vai ficando disponível.

O cliente conecta-se ao CDN e solicita um Uniform Resource Locator (URL) através de um elemento vídeo em HTML. É executado um *script* que envia os cinco pedaços mais recentes para o cliente e os restantes vão sendo transmitidos assim que estejam disponíveis.

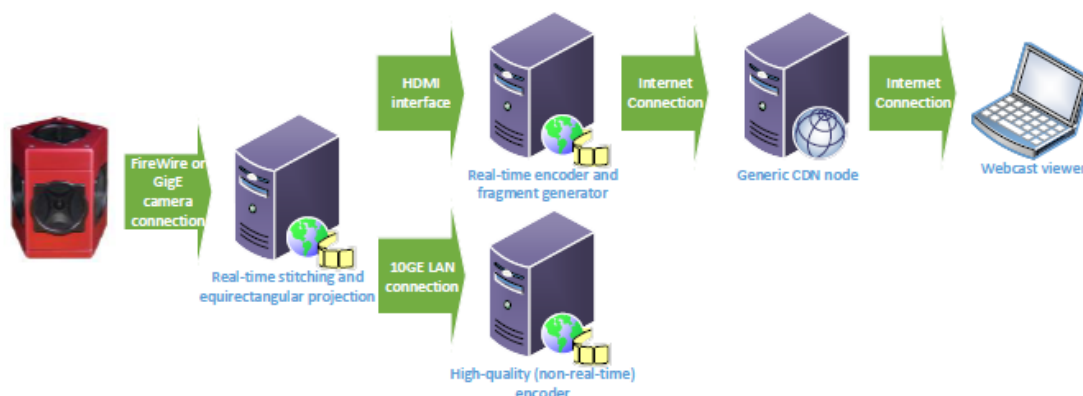


Figura 4. Arquitetura de um sistema que permite o stream de vídeo em 360°, proposta por Quax et al, 2013.

2.2.4. Plataformas existentes

Atualmente, já existem várias plataformas na web que permitem visualizar o conteúdo de vídeo em 360°, tais como Youtube, Vimeo e o Facebook.

O Youtube é uma plataforma de divulgação de vídeos que tem como missão “dar a toda a gente uma voz e mostrá-la ao mundo” (Youtube, 2017a), possuindo mais de 1 bilhão de utilizadores, ou seja, cerca de 1/3 de todas as pessoas da Internet (Youtube, 2017d). Recentemente, o Youtube permitiu que os seus utilizadores pudessem submeter e transmitir vídeos 360° (Hollister, 2015; Perez, 2016), tendo mesmo uma página onde é demonstrado o processo de submissão dos vídeos (Youtube, 2017b) e de preparação do *stream* para a transmissão em 360° (Youtube, 2017c). Algumas empresas e instituições já testaram a transmissão de conteúdos multimédia 360°. Por exemplo, a Union of European Football Associations (UEFA) já transmitiu, em direto, os sorteios da Liga dos Campeões e da Liga Europa para a época 2016/2017 (UEFA, 2017a, 2017b).

O Vimeo é uma plataforma de divulgação de vídeos criada em 2004 por um grupo de cineastas, com o intuito de partilhar os seus trabalhos e momentos da sua vida (Vimeo, 2017b). No presente, são a segunda maior plataforma global de partilha de vídeos, a seguir ao Youtube (Proença, 2017). Desde Março do ano de 2017 que o Vimeo permite que os seus utilizadores submetam os seus vídeos 360° (Lovejoy, 2017; Proença, 2017; Velasco, 2017), dando mesmo indicações de quais as especificações que os vídeos devem ter (Vimeo, 2017a). No entanto, neste momento, ainda não permite a realização de transmissões em direto: apenas o *upload*.

O Facebook, criada em 2004 por Mark Zuckerberg, é, atualmente, a rede social com mais utilizadores no mundo (Hutt, 2017; Levy, 2016) e já permite a transmissão, em direto, de conteúdos multimédia em 360°. Para facilitar a transmissão desses conteúdos multimédia, o Facebook lançou, em abril de 2017, um guia, onde indica as câmaras mais convenientes, as especificações para a transmissão, melhores práticas e como realizar a transmissão, em direto, para o Facebook (Facebook, 2017a). Contudo, esta rede social ainda não permite a visualização das transmissões diretas de conteúdos multimédia utilizando HDM ou outras páginas web que não o próprio Facebook (Facebook, 2017b).

2.3. WebRTC

Neste subcapítulo pretende-se começar por explicar o conceito de WebRTC, apresentar a sua arquitetura, os protocolos que utiliza e alguns exemplos de aplicação existentes.

Até surgir esta tecnologia era necessário, para a comunicação em tempo real via a Internet e recorrendo a *browsers*, a utilização de *plugins* ou outros softwares. Com o WebRTC passa a ser possível comunicar, em tempo real e de forma nativa, entre dois *browsers*. Sendo que os *browsers* mais utilizados implementam esta funcionalidade

2.3.1. Conceito

A comunicação tem sofrido enormes mudanças nos últimos 100 anos. As redes sociais e as televisões pela web vieram destacar a necessidade de interação entre indivíduos através da web. Por outro lado, cada vez mais existe a necessidade de comunicar à distância, que se prendem com a redução do impacto ambiental, dos custos de deslocação, entre outros. Isto torna difícil conseguir estar na presença das pessoas com que se pretende comunicar, embora a necessidade da comunicação presencial subsista (Apostolopoulos et al., 2012). Atualmente, para haver comunicações entre dois *browsers* e entre o *browser* e outros sistemas ainda é necessária a utilização de *plugins* ou softwares para que essa comunicação se processe, o que tem impedido a evolução da interação em tempo real, pois a necessidade de instalação de aplicações ou a falta de sincronização entre dispositivos limita a experiência do utilizador (Djulovic & Duric, 2015; Jennings, Hardie, & Westerlund, 2013).

A tecnologia WebRTC veio simplificar essa comunicação, pois permite a ligação, em tempo real, entre dois *browsers* sem a necessidade de um servidor, *plugins* ou clientes adicionais (Djulovic & Duric, 2015; Jenngins et al., 2013; Loreto & Romano, 2012; Rhinow et al., 2014; Salas & Kalva, 2013). Esta tecnologia está a ser desenvolvida pelos grupos Real-Time Communication in WEB-browsers da Internet Engineering Task Force (IETF RTCWeb) e Web Real-Time Communications Working Group do World Wide Web Consortium (W3C WebRTC). Consiste num conjunto de protocolos, *standards* e numa API em Javascript, que permite estabelecer uma ligação peer-to-peer (P2P) entre diferentes *browsers* (Djulovic & Duric, 2015; Salas & Kalva, 2013), assim como a troca de conteúdos multimédia e outros dados, em tempo real, entre *browsers* (Jenngins et al., 2013; Loreto & Romano, 2012). Resumindo, vários utilizadores podem aceder a um sítio web (servidor) que possua a tecnologia WebRTC. É dada uma identificação para cada utilizador no browser. É a partir dessa identificação que é estabelecida a ligação (Rhinow et al., 2014).

No que respeita à segurança na utilização desta tecnologia, persistem alguns desafios, como sejam (Jenngins et al., 2013):

- O JavaScript pode ser transferido sem o consentimento do utilizador;
- Os utilizadores devem poder aceitar o uso dos recursos audiovisuais do seu computador (câmara e microfone);
- Não deve ser possível transferir grandes quantidades de dados sem o consentimento do utilizador;
- Deve-se manter a privacidade do utilizador e não divulgar os dados do mesmo, especialmente a terceiros;
- Deve haver confidencialidade e autenticação em qualquer transferência de áudio e/ou vídeo.

2.3.2. Arquitetura

A arquitetura base do WebRTC (Figura 5) consiste em, pelo menos, três elementos: um provedor e dois *peers*. Cada um dos *peers* realiza o download da *framework* WebRTC para estabelecer comunicação entre si (Jenngins et al., 2013).

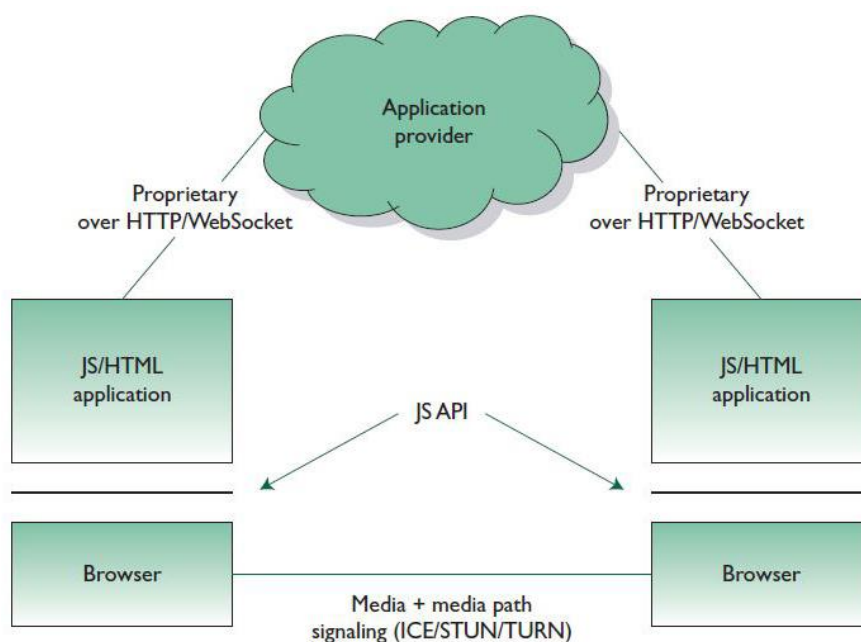


Figura 5. Arquitetura base da tecnologia WebRTC, tal como apresentada no artigo de Loreto e Romano, 2012.

A WebRTC é constituída por três componentes: `RTCPeerConnection`, `RTCDataChannel` e `MediaStream` (Djulovic & Duric, 2015). O `RTCPeerConnection` permite gerir (estabelecer, manter ou terminar) a ligação entre dois *peers*. Após ter uma ligação estabelecida, torna-se possível enviar objetos `MediaStream` entre os *peers* (Djulovic & Duric, 2015; Loreto & Romano, 2012; Rhinow et al., 2014; Salas & Kalva, 2013). Já o `RTCDataChannel` disponibiliza um serviço de transporte genérico de dados, permitindo a troca de dados bidirecional entre *peers* (Djulovic & Duric, 2015; Loreto & Romano, 2012; Rhinow et al., 2014). Por fim o `MediaStream` é a representação de um fluxo de dados de áudio ou vídeo, que pode ser um fluxo remoto ou local. Um `LocalMediaStream` é a representação de um fluxo de voz e/ou dados de um dispositivo de captura local, como uma *webcam* ou um microfone. Para aceder a esses dispositivos, é utilizada a função “`getUserMedia`”, onde é especificado o tipo de *media* que se pretende aceder (Djulovic & Duric, 2015; Jenngins et al., 2013; Loreto & Romano, 2012; Rhinow et al., 2014; Salas & Kalva, 2013).

De acordo com Jennings et al., 2013, para estabelecer uma `RTCPeerConnection` é necessário um processo constituído por vários passos:

- 1) É efetuado um pedido para estabelecer uma ligação a um *peer*;
- 2) O provedor do serviço encaminha o pedido para o *peer*;
- 3) O *peer* aprova ou não a ligação;
- 4) Caso a ligação seja aprovada, são trocados os parâmetros necessários, incluindo o endereço do *peer* pretendido;
- 5) A sincronização da ligação pode ser perdida devido às tentativas para estabelecer a ligação.

A informação que os *browsers* trocam entre si é constituída pela descrição da sessão multimédia, pelo tipo de *media* (voz e/ou vídeo), pelo formato e pelos restantes parâmetros de configuração de forma a estabelecer a ligação (Jennings et al., 2013; Loreto & Romano, 2012).

2.3.3. Protocolos utilizados

No sentido de tornar possível a comunicação e, também, para proteger os dados a transmitir, o `RTCPeerConnection` opera com base em camadas protocolares apresentado na Figura 6. Os principais protocolos, explicados no contexto desta discussão, são (Djulovic & Duric, 2015; Jennings et al., 2013; Loreto & Romano, 2012; Salas & Kalva, 2013):

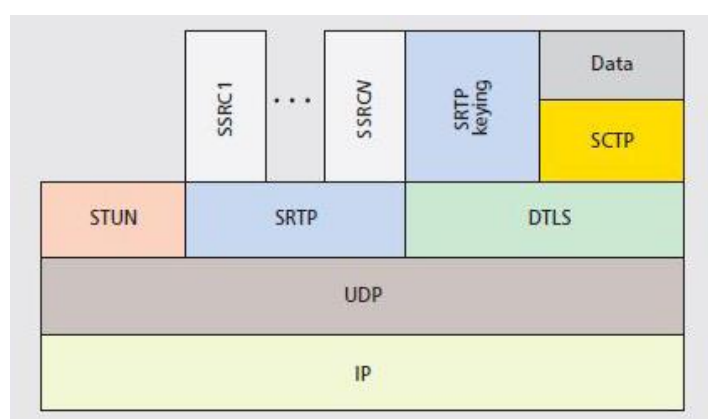


Figura 6. Camadas protocolares do `RTCPeerConnection`, obtido de Jennings et al., 2013.

- Session Description Protocol (SDP) - utilizado para trocar informações de configurações entre os *peers*, para possibilitar a ligação;

- WebSocket - disponibiliza canais de comunicação bidirecionais e permite a troca de mensagens de sinalização, no sentido de estabelecer uma conexão P2P e permitir a troca de mensagens entre um browser e o servidor;
- Real-Time Transport Protocol (RTP) - protocolo usado para o transporte de voz e/ou vídeo em tempo real;
- Secure Real-time Transport Protocol (SRTP) - Garante a confidencialidade e a autenticidade da mensagem transferida;
- Stream control Transmission protocol (SCTP) - Utilizado para o transporte de dados, suporta transporte de mensagem e múltiplos fluxos de dados;
- RTP Control Protocol (RTCP) - Para monitorizar as transmissões dos *media*;
- User Datagram Protocol (UDP) - Possibilita estabelecer um fluxo de transporte P2P, quando os *peers* estão atrás de uma *firewall* ou numa rede que utiliza Network Address Translation (NAT);
- Session Traversal Utilities for NAT (STUN), Traversal Using Relays around NAT (TURN) e Interactive Connectivity Establishment (ICE) - são utilizados para determinar o endereço do Internet Protocol (IP) e a respetiva porta, para estabelecer a comunicação entre *peers*. Permitem, também, lidar com redes que utilizem NAT e/ou estejam atrás de *firewalls*, para estabelecer ligação entre *peers*.

2.3.4. Plataformas existentes

Atualmente já existem várias plataformas de código aberto que utilizam a tecnologia *WebRTC*, como por exemplo: a OpenWebRTC, SimpleWebRTC e a Jitsi Meet. Existem, ainda, algumas empresas focadas neste sector, como, por exemplo a appear.in.

A plataforma SimpleWebRTC é uma plataforma criada pela empresa &yet, que surgiu por volta de 2008, e que fornece o código para “construir coisas engraçadas com o WebRTC em 5 minutos” (&yet, 2017). Esta empresa fornece o suporte necessário e os elementos de software para desenvolver uma aplicação de comunicação em vídeo, áudio ou para transferir ficheiros. No entanto, não possibilitam o desenvolvimento de uma aplicação com todas as funcionalidades incluídas, nem permitem a visualização de hiperligações externas. Ainda assim, trata-se de uma boa solução para quem quiser criar a sua própria solução de WebRTC.

A OpenWebRTC começou por ser desenvolvida pela Ericsson e foi divulgada em outubro do ano de 2014. Consiste numa *framework* que possibilita a construção de aplicações de modo a comunicar entre *browsers*. Contudo, permite, apenas, a comunicação de áudio, vídeo ou *chat*, não possibilitando a transferência de ficheiros nem a abertura de hiperligações externas. Além disso, também não possui uma aplicação formada, mas sim uma API para ser possível desenvolver soluções próprias.

A appear.in é uma *startup* que faz parte do ambiente de incubadoras da Telenor Digital. Foi lançada em agosto de 2013 e a sua ambição é um serviço de comunicações global, que os seus utilizadores apreciem usar (appear.in, 2017). A appear.in permite criar salas de videoconferência públicas ou privadas e convidar pessoas para as mesmas. Para além das típicas funcionalidades de videoconferência, partilha de vídeo e áudio, permite, ainda, partilhar o ecrã do computador após a instalação de uma extensão. Apesar de ser possível criar múltiplas salas, não ser necessário registo às mesmas e de ser de fácil compreensão e utilização, a plataforma desta empresa tem algumas limitações, como, por exemplo, apenas permitir oito utilizadores simultâneos em modo gratuito ou doze em modo pago. Não possibilita, também, o acesso a hiperligações externas nem a existência de instâncias próprias para aceder às videoconferências: terá sempre de ser utilizada a plataforma appear.in.

A Jitsi Meet é um conjunto de projetos de código aberto que permite desenvolver soluções de videoconferência seguras, tendo surgido no ano de 2003. A Jitsi Meet é uma plataforma que permite fazer videoconferência, partilhar o ecrã, editar documentos de forma partilhada, trocar mensagens, partilhar uma hiperligação da Youtube, começar/acabar uma transmissão para a plataforma Youtube, entre outros. Permite, ainda, criar uma instância da plataforma em servidor, o que se traduz numa não dependência de servidores externos e na possibilidade de adaptar os serviços a contextos mais específicos.

2.4. Frameworks de desenvolvimento web

Neste subcapítulo começa-se por definir *framework* para o desenvolvimento web, apresentar as suas vantagens e desvantagens e comparar as *frameworks* mais utilizadas atualmente, particularmente as baseadas em Hypertext Preprocessor (PHP).

A evolução das aplicações desenvolvidas para a web, nomeadamente a necessidade da existência de uma representação digital, a sua complexidade, prazos de desenvolvimento e implementação e imperativos de atualização quer de valências quer gráficas, tem vindo a exigir alterações significativas nos procedimentos de desenvolvimento para a web. Assim, as *frameworks* surgem como forma de potenciar um conjunto de funcionalidades e de valências comuns à maioria dos sítios web, mas também procuram criar padrões e modelos de implementação que, simultaneamente, permitam uma maior produtividade, garantindo as exigências de qualidade e de segurança.

2.4.1. Conceito

O conceito de *framework* é definido pelo dicionário de Oxford (Oxford, 2017) como um conjunto de crenças, ideias ou regras que são usadas como base para fazer julgamentos e decisões. Esta definição coincide com a que consta no dicionário de Cambridge (Cambridge, 2017) e Macmillan (Macmillan, 2017).

No contexto do desenvolvimento de software, uma *framework* proporciona a base sobre a qual é desenvolvida uma aplicação específica (Techterms, 2013). Esta base pode consistir em programas de apoio, bibliotecas de código, linguagens de *scripting* ou outros *softwares* (Franco, Piedade, & Rêgo, 2014). Com efeito, no caso específico do desenvolvimento de aplicações para a web, uma *framework* disponibiliza, tipicamente, interfaces para gestão de bases de dados, gestão de sessões e de *cookies*, criação e gestão da informação em *templates* gráficos, validação de entradas de dados e operações de criação, leitura, modificação e remoção de conteúdos (do inglês *create, read, update and delete* - CRUD) (Statz, 2010).

No que concerne às vantagens obtidas pela utilização de *frameworks* no desenvolvimento para a web podem-se destacar (1stWebDesigner, 2015; Bezerra & Juliano, 2016; Noupe, 2009; Wang, 2011):

- Tornar o processo de desenvolvimento mais expedito. Nesse sentido, uma tarefa que poderia demorar algumas horas para desenvolver é realizada em pouco tempo, caso a *framework* a suporte, o que promove um desenvolvimento mais rápido de aplicações;
- A existência de uma comunidade de apoio e de revisões regulares do código da *framework*, o que permite lidar com eventuais erros no código e torná-la mais segura;
- A disponibilização de documentação detalhada e de uma comunidade ativa, o que permite o rápido e eficaz esclarecimento de eventuais dúvidas, questões e a obtenção de melhores práticas;
- Serem em código-aberto, o que se traduz, também, na sua gratuidade.

Relativamente às desvantagens das *frameworks* de desenvolvimento para a web, podem-se destacar (1stWebDesigner, 2015; Bezerra & Juliano, 2016):

- Não são indicadas para todo o tipo de aplicações. Se a aplicação for simples, como por exemplo uma página estática, não faz sentido a sua utilização, pois não se tira proveito da utilização da *framework*;
- Nem sempre a utilização de uma *framework* na construção de uma aplicação significa a obtenção de um melhor desempenho. Se a aplicação for demasiado simples, o uso de uma *framework* terá um impacto negativo no seu desempenho;
- Uma *framework*, sendo código aberto, está disponível para quem tem boas e más intenções. Isto significa que podem ser usadas para encontrar falhas ou atacar as aplicações que usem essa *framework*;
- Desenvolver numa dada *framework*, não significa que também se estejam a adquirir conhecimento na linguagem de programação que a suporta, o que pode ser limitador caso se pretenda desenvolver elementos de software para completar ou modificar a *framework*, no sentido de a ajustar a um dado contexto ou para lhe adicionar suporte a alguma especificidade.

2.4.2. Frameworks de PHP

O PHP é uma linguagem de *scripting* de código-aberto e com um elevado desempenho, muito utilizada no desenvolvimento para a web (Figura 7). Permite uma aprendizagem rápida para quem se está a iniciar, disponibiliza muitos recursos para os programadores mais avançados e possui uma documentação detalhada, com diversos exemplos. A comunidade é vasta, o que potencia quer o processo de aprendizagem, quer a procura de soluções para eventuais dúvidas. (Bezerra & Juliano, 2016; Franco et al., 2014; PHP, 2017). A partir da sua versão 4, passou a suportar programação orientada a objetos, o que permite, também, o desenvolvimento de aplicações mais robustas e, dessa forma, conseguiu atrair ainda mais programadores (Franco et al., 2014).

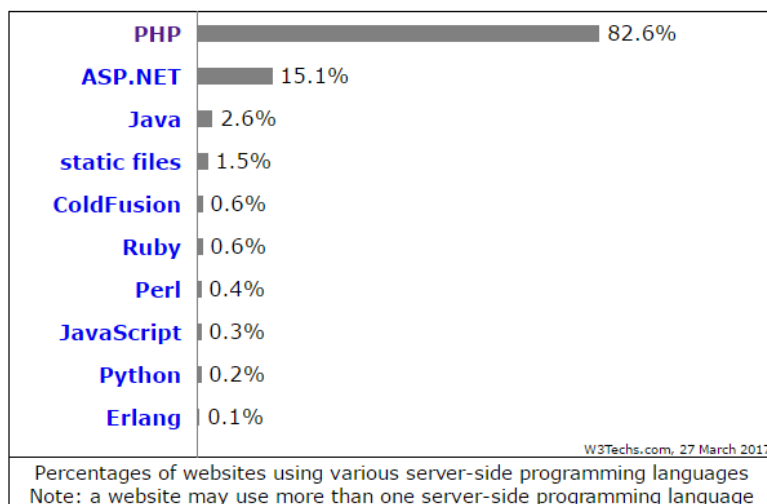


Figura 7. As linguagens de programação para a web, que operam do lado do servidor, mais utilizadas no mês de março de 2017. Pode-se constatar que o PHP está presente em 82,6% dos sítios web consultados (W3Techs, 2017).

O estilo arquitetónico Model-View-Controller (MVC) é utilizado por diversas *frameworks* de desenvolvimento para a web como forma de organizar os elementos que constituem uma aplicação. Este modelo, apresentado na Figura 8, surge da necessidade de compartimentar o código que respeita à componente de apresentação de informação da componente de negócio da aplicação. Assim, é possível desenvolver, alterar e realizar a manutenção de cada um destes componentes sem interferir nos restantes (Noupe, 2009; Pop & Altar, 2014; Wang, 2011). Tendo esta regra por base, têm vindo a surgir diversas variações deste modelo, mas, sobretudo, das interações entre cada um dos seus três componentes.

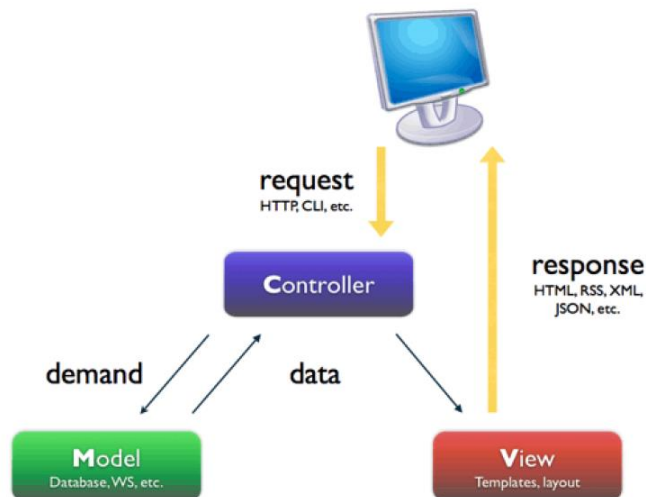


Figura 8. Exemplo do estilo arquitetônico MVC e da respetiva interação entre os seus três elementos, tal como apresentada no trabalho de Pop & Altar, 2014.

Resumidamente, a função de cada um dos elementos que constitui o estilo arquitetónico MVC é (Franco et al., 2014; Pop & Altar, 2014; Wang, 2011):

- Modelo (Model): elemento que gere os modelos de dados da aplicação, responde a pedidos de informação sobre o seu estado e a instruções para a alteração de estado;
- Visualização (View): interface com o utilizador, que é responsável por controlar a forma como a informação é apresentada ao utilizador e como este interage com ela. Podem existir diversas visualizações para um modelo de dados, conforme o fim a que se destinam;
- Controlador (Controller): é responsável pela gestão de eventos na aplicação. Estes podem ser acionados pelas interações de um utilizador ou por um processo do próprio sistema. Assim, o controlador aceita as entradas dos utilizadores e as instruções do modelo, agindo de acordo com elas.

Atualmente existem diferentes *frameworks* PHP de desenvolvimento para a web disponíveis. As que foram consideradas de melhor qualidade, no ano de 2016, segundo Banerjee, 2016; Monus, 2017; World Web Technology, 2016, foram a Laravel, Codeigniter, Zend, CakePHP e Symphony.

2.4.3. Comparação entre *Frameworks* PHP

Para seleccionar uma *framework* de desenvolvimento para a web devem ser considerados diversos fatores (Franco et al., 2014; Noupe, 2009), como, por exemplo, a:

- Facilidade de uso e a estrutura de pastas;
- Popularidade junto de outros programadores;
- Dimensão da comunidade e o suporte existente;
- Documentação que a suporta, nomeadamente se esta é atualizada, completa, compreensível e acessível;
- Flexibilidade da *framework*. Mais concretamente, se é ou não fácil integrar código próprio desenvolvido pelo programador.

Assim, considerando as *frameworks* para o desenvolvimento de aplicações web mais pesquisadas nos 12 meses anteriores ao dia 17 de abril de 2017 (Figura 9) e a lista de *frameworks* já apresentada, na secção 2.4.2, como tendo sido considerada de melhor qualidade no mesmo ano, é feita, de seguida, uma breve apresentação sobre as *frameworks* Laravel, CodeIgniter e Symfony.

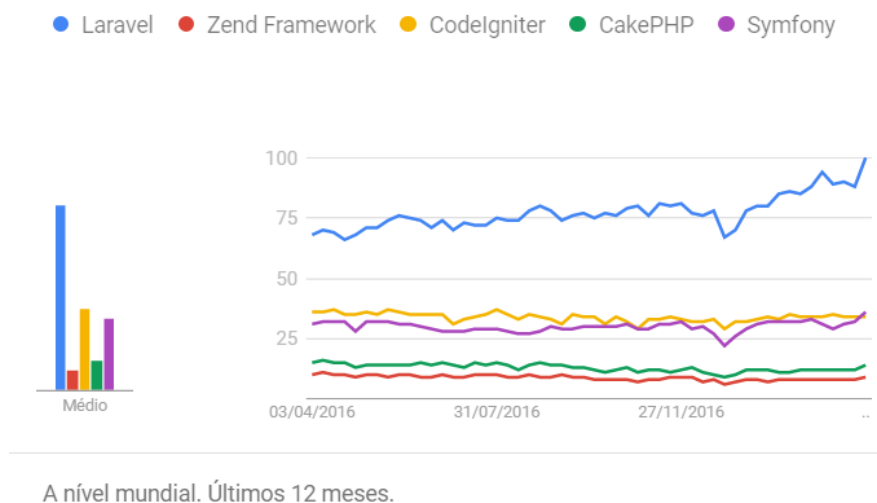


Figura 9. *Frameworks* PHP de desenvolvimento para a web mais pesquisadas nos últimos 12 meses. Esta consulta foi realizada no dia 17 de abril de 2017, recorrendo à ferramenta Google Trends.

A Laravel é uma *framework* relativamente nova - surgiu em 2011 - que surgiu com o propósito de desenvolver uma ferramenta onde fosse divertido programar. Em poucos anos tornou-se uma das melhores frameworks PHP de desenvolvimento para a web (Bezerra & Juliano, 2016; Monus, 2017; World Web Technology, 2016). Possui uma comunidade muito ativa, documentação detalhada, é estável, tem recursos que facilitam o Rapid Application Development (RAD) e é de fácil aprendizagem (Banerjee, 2016; Bezerra & Juliano, 2016; Monus, 2017).

A Symfony é descrita pela empresa como um conjunto de componentes PHP, uma *framework* de aplicações Web, uma filosofia e uma comunidade onde todos trabalham juntos, em harmonia. Os componentes da Symfony são bibliotecas reutilizáveis que facilitam algumas tarefas, tais como: criar formulários, configurar objetos, rotas, autenticação. Estes componentes são utilizados em muitos projetos como o Drupal, o phpBB e até mesmo a Laravel (Monus, 2017; Symfony, 2017; World Web Technology, 2016).

A Codeigniter é descrita pela empresa que a gere como uma poderosa *framework* PHP, construída para programadores que necessitam de ferramentas simples e elegantes para criar aplicações web complexas (CodeIgniter, 2017). Teve o seu lançamento em 2006, tem um processo de instalação rápido e simples, funciona na maioria das plataformas de alojamento, permite integrar aplicações de terceiros para funções mais complexas e tem uma documentação completa e uma curva de aprendizagem suave (Banerjee, 2016; Monus, 2017; World Web Technology, 2016).

A Tabela 1 apresenta uma comparação entre estas três *frameworks* PHP no que respeita à data da última versão, que bases de dados suportam, quais os requisitos para a utilização e o qual o modelo de organização utilizado.

	Laravel	Symfony	CodeIgniter
Última Versão	5.4.0 (janeiro 2017)	3.2.4 (fevereiro de 2017)	3.1.4 (março de 2017)
Padrão	MVC	MVC	MVC/HMVC ¹
Requisitos	• Laravel Homestead ou PHP >= 5.6.4 • Extensão OpenSSL PHP • Extensão PDO PHP • Extensão Mbstring PHP • Extensão Tokenizer PHP • Extensão XML PHP	• PHP >= 5.5.9 • Extensão JSON • Extensão ctype. • php.ini tem de ter a definição date.timezone	PHP >= 5.6
Base de dados	• MySQL • Postgres • SQLite • SQL Server	Livraria Doctrine (MySQL, PostgreSQL, Microsoft SQL, MongoDB)	• MySQL • Oracle • PostgreSQL • MS SQL • SQLite • CUBRID • ODBC

Tabela 1. Comparação entre as frameworks PHP Laravel, Symfony e Codeigniter

¹ Hierarchical Model-View-Controller (HMVC) é um modelo onde é obrigatório usar o controlador, mas o uso do modelo e da visualização é opcional (Banerjee, 2016; Monus, 2017).

3. Plataforma de vídeo 360° com integração de tecnologias de interatividade e de colaboração em tempo real

Neste capítulo será descrito o processo de desenvolvimento de uma plataforma de vídeo 360°, com integração de tecnologias de interatividade e de colaboração em tempo real, nomeadamente a sua definição, a seleção das tecnologias utilizadas e a sua implementação. É, ainda, apresentado, um guia passo-a-passo que permite reproduzir a implementação da plataforma proposta, onde são referidos, também, alguns dos problemas encontrados e as suas respetivas soluções.

A plataforma proposta tem um cariz sobretudo informativo, onde é permitido ver e comentar notícias e/ou vídeos. Permite, também, aceder a transmissões de eventos em direto e possibilita a interação com outros utilizadores, no seu decurso. Possui, ainda, diferentes níveis de utilizador, com permissões diversas.

3.1 Conceção e Modelo

Nesta secção pretende-se apresentar o processo de conceção e planeamento da plataforma proposta, nomeadamente: a análise de requisitos, os casos de uso, o diagrama de atividades, os diagramas de estado, a estrutura da base de dados e a proposta de *layout*.

3.3.1 Análise de requisitos

Como fase inicial no processo de conceção e definição da plataforma proposta foi realizada uma análise de requisitos (funcionais e não-funcionais). Com ela, foi possível encontrar cinco atores (Administrador, Jornalista, Moderador, Utilizador não Autenticado e Utilizador Autenticado) com base nos quais foram construídos outros tantos pacotes de serviços (Tabela 2).

	Pacote	Descrição
P1	Serviços Gerais	Todos os utilizadores têm o P1 e depois os seus respetivos pacotes de serviços. Estes podem aceder a informações da plataforma, mas não interagir com a mesma como, por exemplo: consultar/partilhar os conteúdos, visualizar a transmissão de eventos em direto, registar-se, etc.
P2	Serviços Gerais dos Utilizadores Registados	Requisitos dos utilizadores autenticados na plataforma, que, para além dos referidos em P1, podem, ainda: ver/editar o seu perfil, comentar os conteúdos multimédia, participar nas transmissões de eventos em direto.
P3	Serviços dos Jornalistas	Requisitos referentes aos jornalistas e à responsabilidade de gestão do conteúdo (notícias e vídeos) da plataforma.
P4	Serviços dos Moderadores	Requisitos referentes aos moderadores e à gestão da interação do utilizador com a plataforma/direto: gestão dos utilizadores, gestão do direto, gestão dos comentários aos conteúdos multimédia.
P5	Serviços do Administrador	Requisitos referentes ao administrador e à gestão global da plataforma: gestão da informação dos utilizadores, consulta de estatísticas da plataforma, gestão dos conteúdos multimédia a apresentar, gestão da transmissão de eventos em direto.

Tabela 2. Esta tabela apresenta os cinco pacotes de serviços pensados para a plataforma proposta, bem como uma pequena descrição de cada um deles.

Depois de definir, de uma forma mais geral, quais os serviços necessários à plataforma e tê-los agrupados por tipo de utilizador, procurou-se, por pacote de serviços, detalhar as valências que se pretende ver implementadas. Assim, foram criadas tabelas para cada um dos pacotes de serviços definidos – P1 (Tabela 3), P2 (Tabela 4), P3 (Tabela 5), P4 (Tabela 6) e P5 (Tabela 7) – que contém uma identificação do serviço (código) e uma breve descrição.

Código	Descrição
UC1	Visualizar as notícias mais lidas.
UC2	Visualizar as últimas notícias inseridas.
UC3	Visualizar os vídeos mais vistos.
UC4	Visualizar os últimos vídeos inseridos.
UC5	Ver informações gerais da plataforma.
UC6	Consultar os contactos referentes à plataforma.
UC7	Partilhar os conteúdos da plataforma (vídeos e notícias).
UC8	Registo de utilizador.
UC9	Registo através da rede social Facebook.
UC10	Pesquisa de conteúdos (notícias e vídeo).
UC11	Consulta de conteúdos (notícias e vídeo).
UC12	Visualizar a transmissão de eventos em direto.
UC13	Visualizar os comentários dos conteúdos (notícias e vídeos).
UC14	Efetuar a autenticação.
UC15	Listagem de notícias.
UC16	Listagem de vídeos.
UC17	Confirmação do registo de utilizador através do email.
UC18	Login através da rede social Facebook.
UC19	Recuperação da palavra-passe.

Tabela 3. Requisitos funcionais dos Serviços Gerais (P1): funcionalidades a que todos os utilizadores (incluindo os não autenticados) podem aceder.

Código	Descrição
UC1	Efetuar o <i>logout</i> .
UC2	Editar o perfil de utilizador.
UC3	Comentar conteúdos multimédia (notícias e vídeos).
UC4	Aceder à emissão em direto da transmissão interativa.
UC5	Contactar a administração da plataforma.

Tabela 4. Requisitos funcionais dos utilizadores registados (P2): funcionalidades a que os utilizadores autenticados podem aceder, além dos do pacote comum P1.

Código	Descrição
UC1	Inserir uma notícia.
UC2	Editar uma notícia.
UC3	Apagar uma notícia.
UC4	Inserir um vídeo e informações sobre o mesmo.
UC5	Editar um vídeo e informações sobre o mesmo.
UC6	Apagar um vídeo.
UC7	Listar as notícias.
UC8	Listar os vídeos.
UC9	Visualização de uma notícia.
UC10	Visualização de um vídeo.
UC11	Consulta das estatísticas da plataforma.
UC12	Alterar a visibilidade do conteúdo (notícia e vídeo).
UC13	Pesquisar notícia.
UC14	Pesquisar vídeo.

Tabela 5. Requisitos funcionais dos Serviços dos Jornalistas (P3): funcionalidades que os jornalistas podem aceder, além dos do pacote comum (P1).

Código	Descrição
UC1	Apagar comentário do conteúdo multimédia.
UC2	Ver listagem de utilizadores.
UC3	Ver perfil de utilizador.
UC4	Bloquear utilizador.
UC5	Desbloquear um utilizador.
UC6	Listagem dos comentários.
UC7	Visualizar comentários.
UC8	Consulta de estatísticas da plataforma.

Tabela 6. Requisitos funcionais dos Serviços dos Moderadores (P4): funcionalidades que os moderadores podem aceder, além do pacote comum (P1).

Código	Descrição
UC1	Consulta de estatísticas da plataforma.
UC2	Alterar o nível de utilizador.
UC3	Criar utilizador do tipo “Jornalista”.
UC4	Criar utilizador do tipo “Moderador”.
UC5	Envio de dados de utilizador por email.
UC6	Listagem de utilizadores.
UC7	Alterar informações sobre a plataforma.
UC8	Listagem dos eventos em direto.
UC9	Adicionar um evento em direto.
UC10	Editar um evento em direto.
UC11	Apagar um evento em direto.
UC12	Pesquisar um evento em direto.

Tabela 7. Requisitos funcionais dos Serviços do Administrador (P5): funcionalidades que os administradores podem aceder, além do pacote comum (P1).

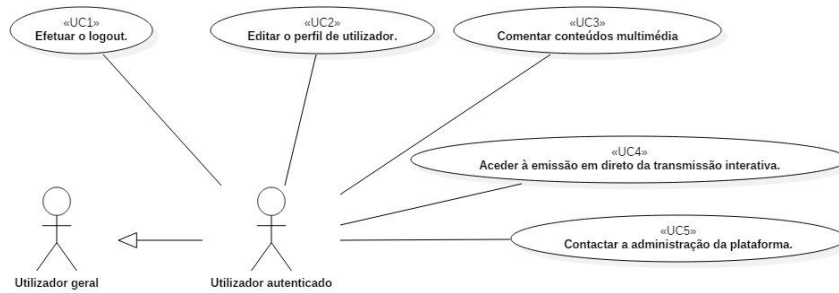


Figura 11. Caso de uso para os utilizadores registados, onde é apresentada a relação entre um utilizador autenticado e as respetivas funcionalidades da plataforma.

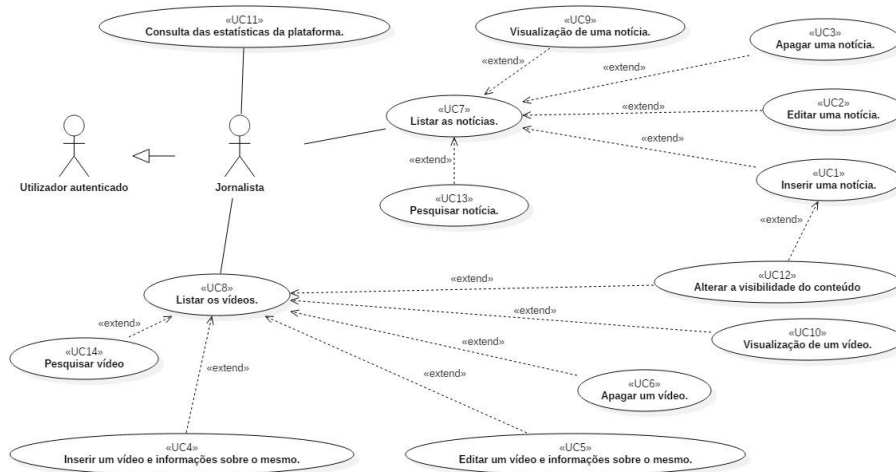


Figura 12. Caso de uso que apresenta a relação entre os utilizadores do tipo Jornalista e as funcionalidades da plataforma com que podem interagir.

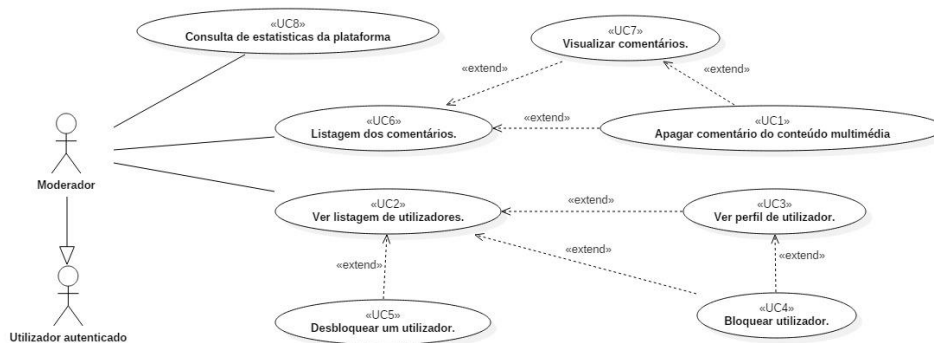


Figura 13. Caso de uso que apresenta a relação entre os utilizadores do tipo Moderador e as funcionalidades da plataforma com que podem interagir.

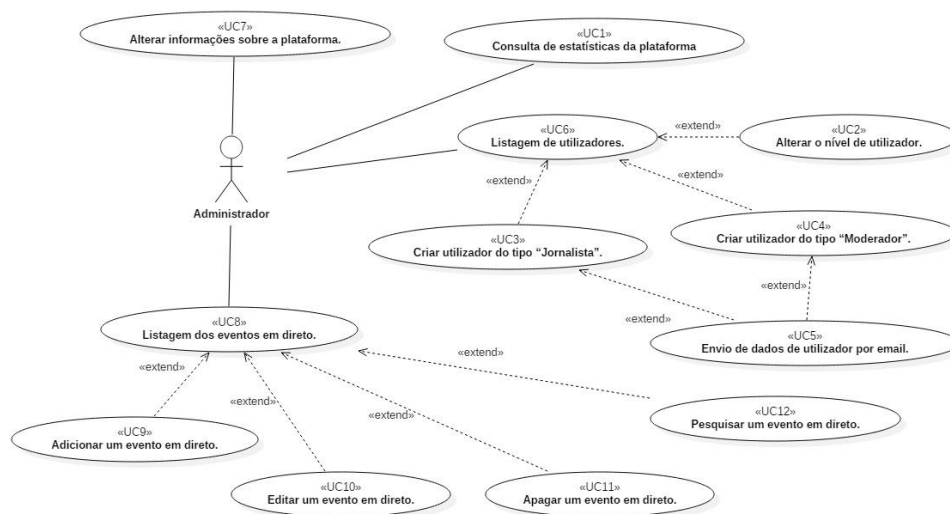


Figura 14. Caso de uso para o utilizador do tipo Administrador.

3.3.3 Diagramas de atividades

Após a conclusão dos casos de uso foi necessário procurar perceber melhor a forma como o utilizador e a plataforma iriam interagir. Nesse sentido, foram desenvolvidos dois diagramas de atividades: (1) acesso à plataforma (Figura 15); e (2) registo de utilizador (Figura 16).

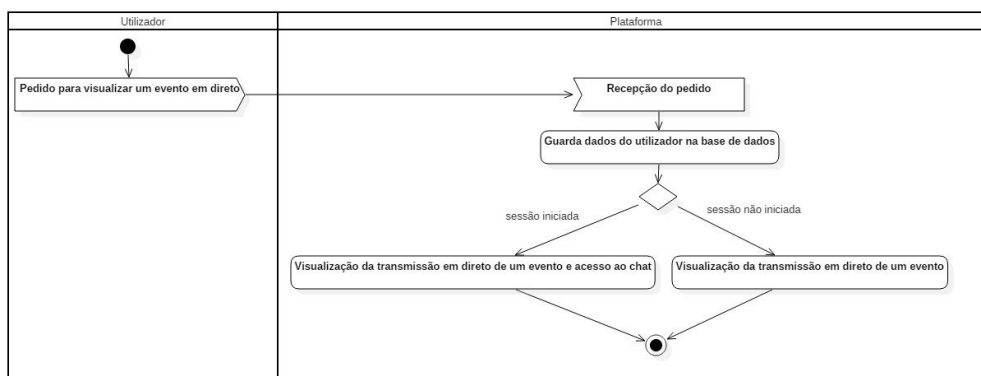


Figura 15. Diagrama de atividades da do acesso à plataforma, onde é perceptível o procedimento necessário para visualizar a transmissão, em direto, de um evento, bem como a participação nele, caso esteja autenticado.

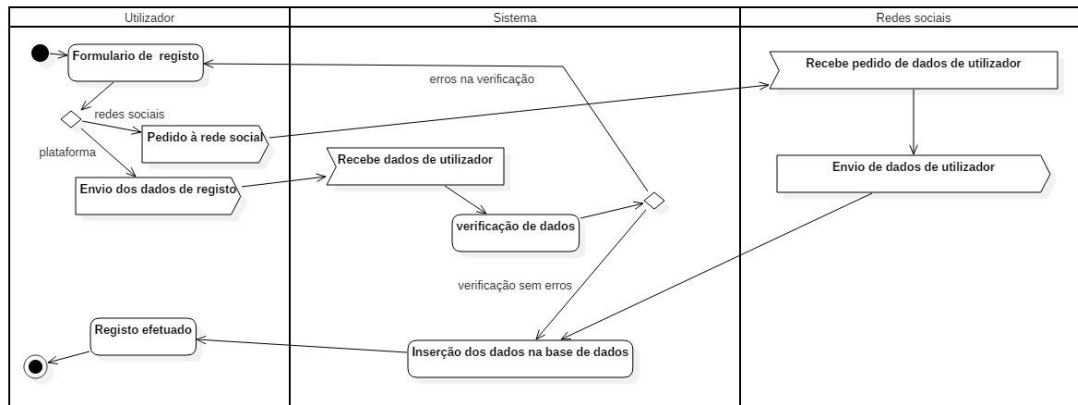


Figura 16. Diagrama de atividades que apresenta as interações necessárias no processo de registo de um utilizador na plataforma, quer em formulário próprio, quer utilizando os dados provenientes de uma rede social.

3.3.4 Diagramas de estado

Nesta subsecção são apresentados três diagramas referentes aos estados que conteúdos multimédia, utilizadores e comentários de utilizadores podem ter no sistema. Assim, temos:

- Conteúdo multimédia: tal como apresentado na Figura 17, pode assumir o estado público, privado ou em destaque (apenas no caso de se tratar de uma notícia);

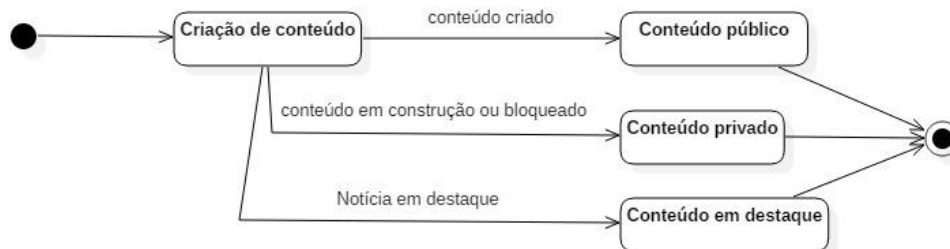


Figura 17. Diagrama de estados de um conteúdo multimédia. Pode assumir o estado público, privado ou em destaque.

- Utilizadores: um utilizador pode assumir o papel de administrador, jornalista, moderador ou, apenas, utilizador geral. Pode, ainda, estar na situação de ativo na plataforma ou bloqueado. A Figura 18 apresenta este diagrama de estados;

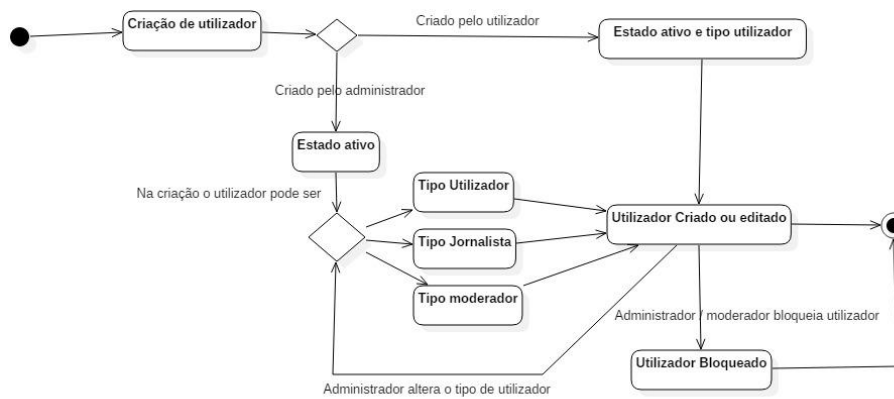


Figura 18. Diagrama relativo ao tipo e ao estado do utilizador. Um utilizador pode estar ativo ou bloqueado e assumir o papel de Administrador, Jornalista, Moderador ou Utilizador geral.

- Comentários de utilizadores: os comentários submetidos pelos utilizadores à plataforma e relativo a algum dos conteúdos multimédia disponíveis ou em transmissão, podem assumir o estado de ativo ou bloqueado, tal como é apresentado na Figura 19.

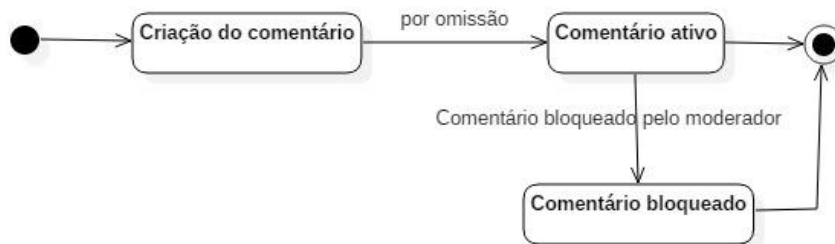


Figura 19. Diagrama de estado relativo aos comentários dos utilizadores a algum conteúdo multimédia da plataforma. Podem assumir o estado ativo ou bloqueado.

3.3.5 Base de dados

A base de dados que suporta o funcionamento é constituída por sete tabelas (Figura 20). Além da gestão da informação da plataforma, pretendem-se, ainda, a extração de dados para constituir alguma informação estatística relativa às transmissões em direto, grau de satisfação e de adesão dos utilizadores, entre outros.

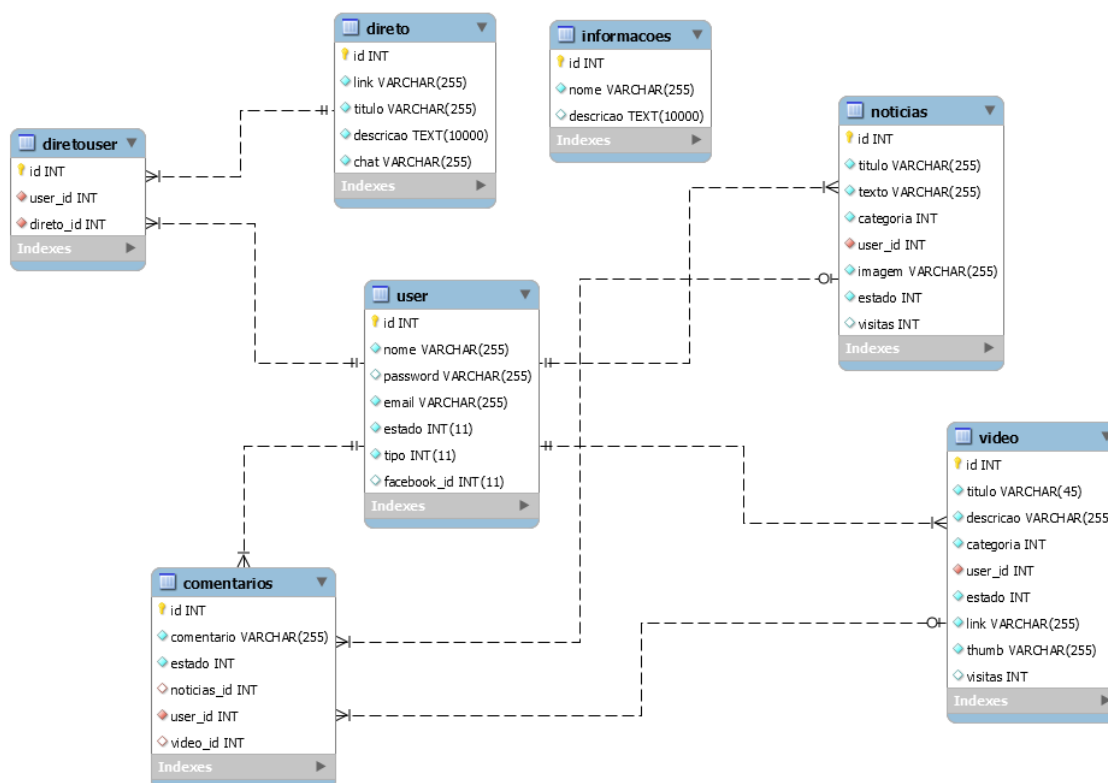


Figura 20. Base de dados desenvolvida para a plataforma de transmissão de vídeo 360° com integração de tecnologias de comunicação para suporte de interação em tempo real.

A Tabela 8 apresenta uma descrição sumária dos objetivos e dos campos existentes em cada uma das sete tabelas que constitui a base de dados da plataforma.

Tabela	Objetivo	Campos Relevantes
user	Informação base acerca do utilizador	id, password, email, estado, tipo, facebook_id
informacoes	Conteúdo das páginas que são fixas na plataforma	id, nome
noticias	Informação relativas às notícias da plataforma	id, categoria, user_id, estado, visitas
videos	Informação relativas aos vídeos da plataforma	id, categoria, user_id, estado, link, thumb, visitas
comentarios	Informação acerca dos comentários de conteúdos de notícias/vídeos	id, estado, noticias_id, user_id, video_id
directo	Informação acerca dos diretos assim como o link para o mesmo e para a sala de chat	id, link, chat
direto_user	Informação acerca das pessoas que entraram na página do direto	id, user_id, direto_id

Tabela 8. Descrição sumária dos objetivos e dos campos existentes nas tabelas da base de dados.

3.3.6 Proposta de *Layout*

Nesta subsecção apresenta-se o desenho desenvolvido para algumas das páginas que vão constituir a plataforma. Procurou-se, com estas propostas, determinar a organização do sítio web, a estrutura da informação a apresentar aos utilizadores e a forma como estes podem interagir com a plataforma da maneira mais eficiente e simples. No sentido de estruturar a informação da melhor forma, procurou-se estudar plataformas já existentes nas áreas abrangidas. Esse estudo foi a base da conceção e do desenvolvimento dos *layouts* propostos. Algumas das plataformas visitadas foram: Notícias ao Minuto, O Jogo, Publico, Jornal de Notícias, Economia Online, e-konomista, entre outros.

De seguida apresentam-se alguns dos *layouts* desenvolvidos:

- Página principal, onde os utilizadores podem ter acesso aos conteúdos multimédia mais recentes, assim como às notícias, incluindo as em destaque e às mais lidas (Figura 21);

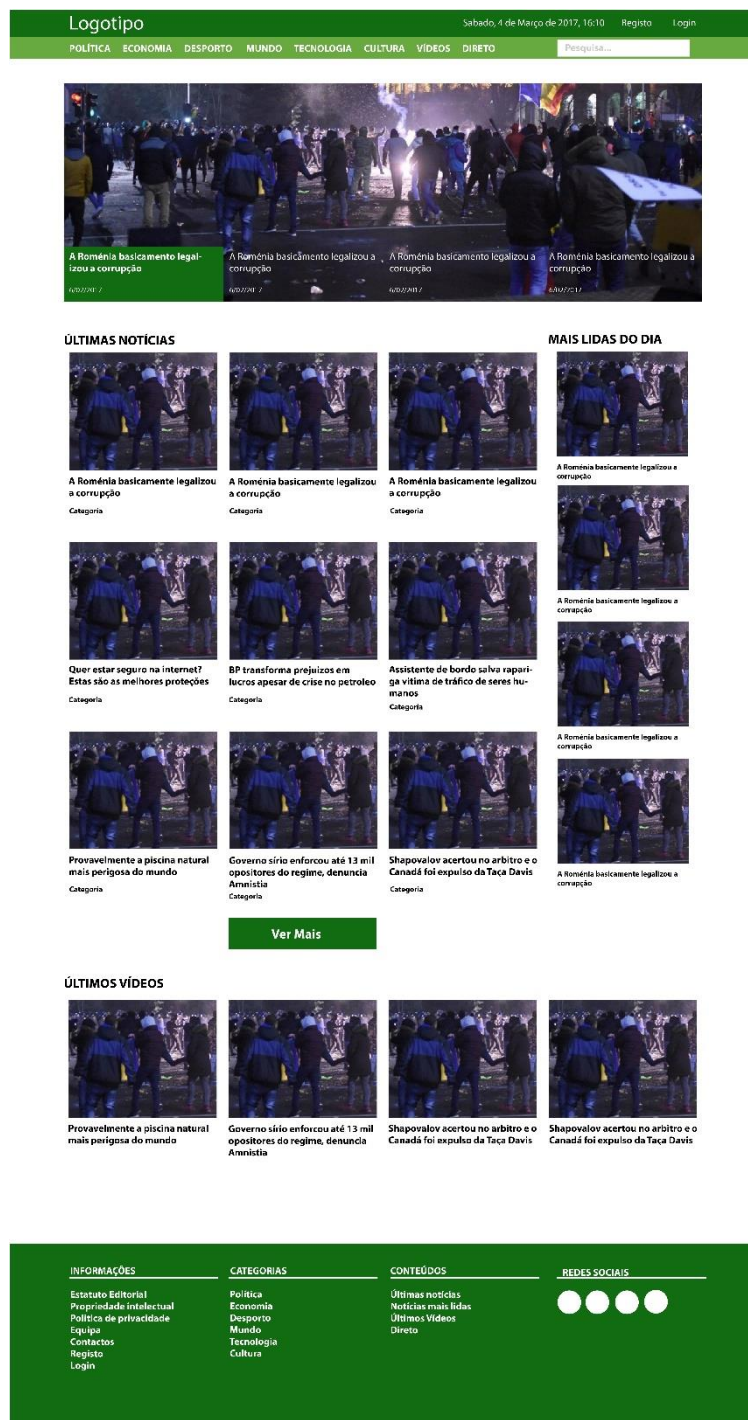


Figura 21. Layout da página principal da plataforma, onde são apresentadas quatro notícias em destaque, as últimas notícias inseridas, as notícias mais lidas e os últimos vídeos inseridos.

- Página de Categoria (Figura 22), onde podem ser acedidos às notícias mais recentes de uma dada categoria, assim como às mais lidas e aos últimos conteúdos multimédia disponibilizados;

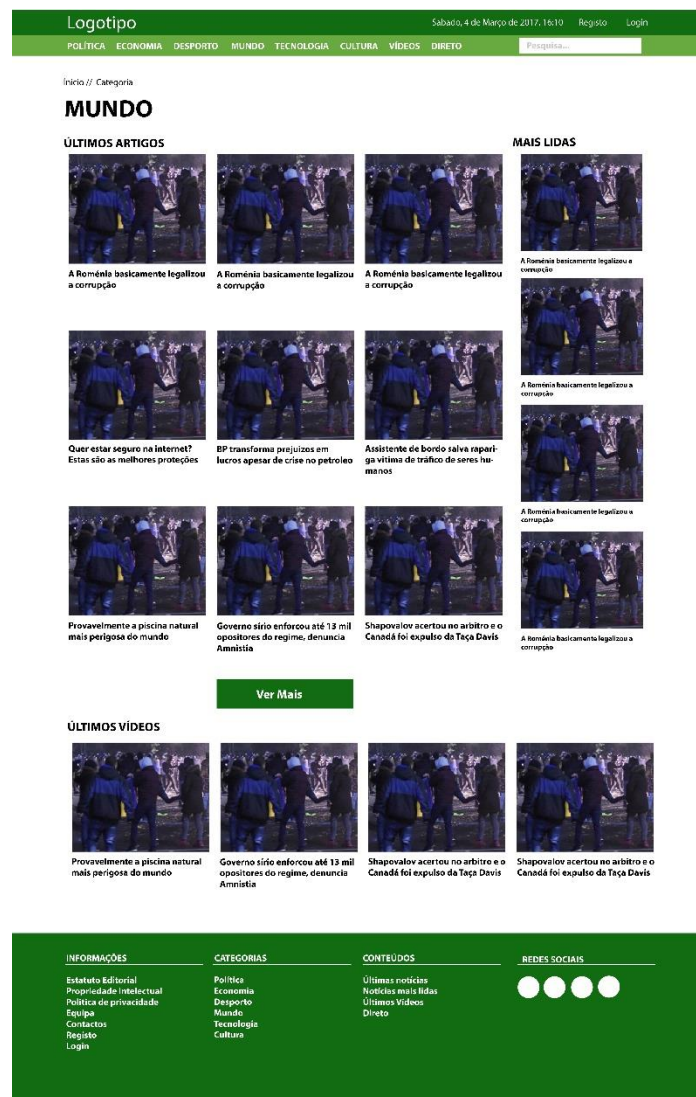


Figura 22. Layout da página de uma categoria, onde é possível aceder às últimas notícias dessa categoria, às mais lidas e aos últimos conteúdos multimédia inseridos.

- Página da notícia. Apresenta o detalhe de uma dada notícia, a possibilidade de partilhar os seus conteúdos, os comentários e a hipótese de submeter um novo comentário, assim como as notícias mais lidas e mais recentes da plataforma (Figura 23);

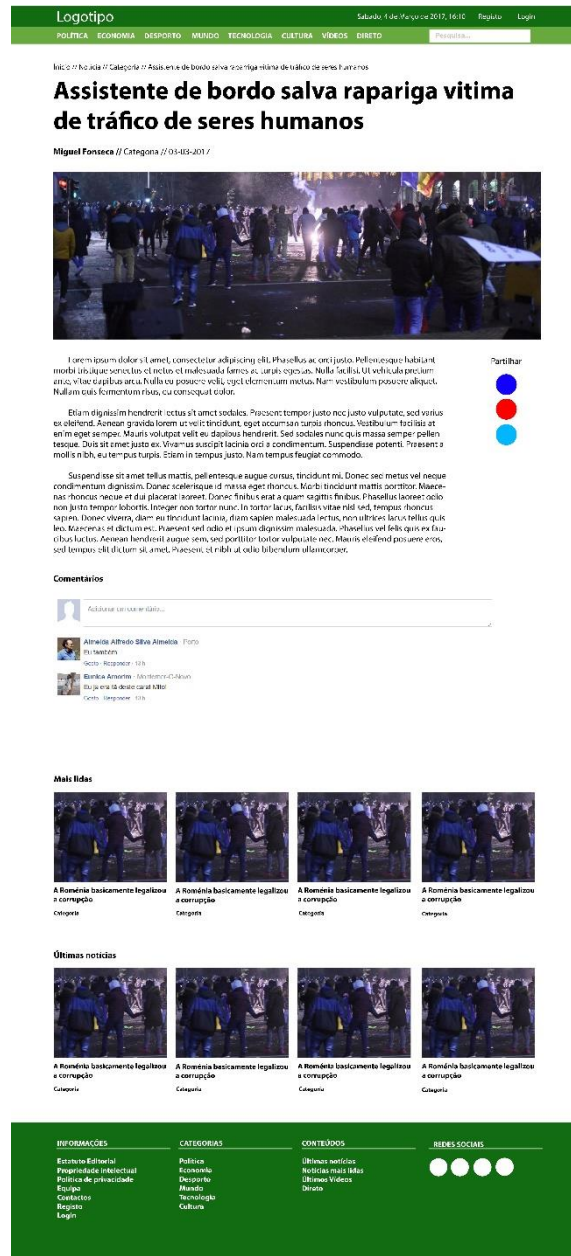


Figura 23. Layout da página de uma notícia, onde é possível ver a informação da notícia, os seus comentários, comentar (caso o utilizador esteja autenticado), as notícias mais lidas, as últimas notícias e, ainda, a possibilidade de partilhar a notícia.

- A Figura 24 apresenta a páginas dos conteúdos multimédia, onde é disponibilizado um vídeo e respetiva informação, a possibilidade de o partilhar nas redes sociais, assim como de o comentar e de ler os comentários já disponíveis. É, ainda, possível, ter acesso aos conteúdos multimédia mais recentes e aos mais visualizados;

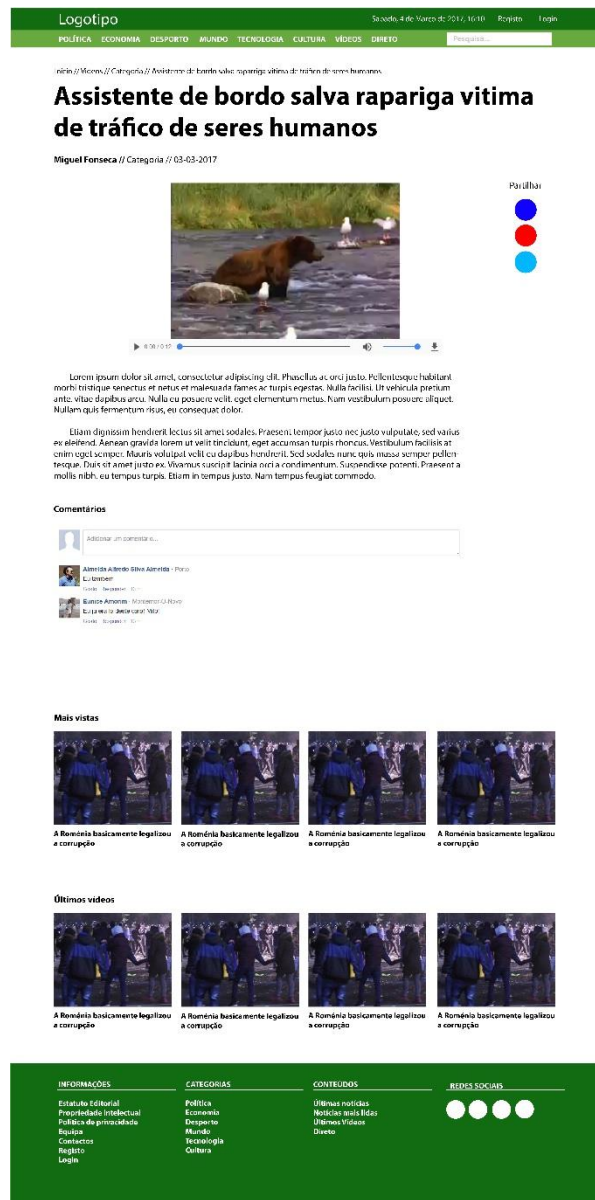


Figura 24. Layout da página de um conteúdo multimédia, onde é possível ver informação acerca dele, os comentários já efetuados, submeter um novo comentário (caso o utilizador esteja autenticado), ter acesso aos conteúdos multimédia mais recentes e os mais vistos, bem como a possibilidade de partilhar o conteúdo multimédia.

- Página da transmissão, em direto, de eventos (Figura 25), onde é possível aceder à transmissão, em tempo real, assim como a informação sobre a mesma. O utilizador pode, também, partilhar o evento e aceder às últimas notícias e conteúdos multimédia;

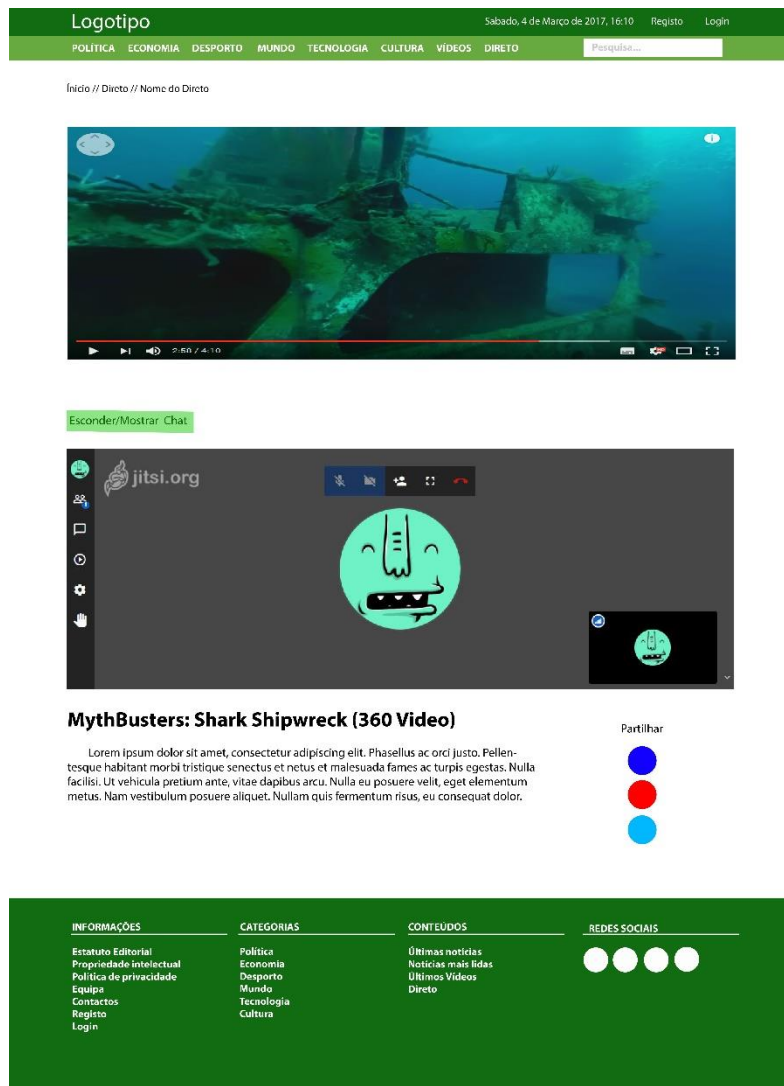


Figura 25. Layout da página da transmissão em direto, onde pode ser consultada informação relativa à transmissão do evento, é permitida a partilha do evento e, caso o utilizador esteja autenticado, é permitido aceder ao chat, em tempo real, com os restantes utilizadores e com a gestão da emissão.

- Página do Jornalista. Nela é apresentada uma listagem das notícias do jornalista, ordenadas cronologicamente, assim como as mais lidas (Figura 26).



Figura 26. Layout da página de um jornalista, onde é possível aceder as últimas notícias, às mais lidas e é disponibilizada a hipótese de o utilizador entrar em contacto com o jornalista.

3.2 Arquitetura proposta

A proposta de uma plataforma que integre o *streaming* de conteúdos multimédia e tecnologias de interatividade e de colaboração em tempo real está assente numa arquitetura que consiste em três componentes principais: a base de dados, as aplicações e a interface (Figura 27).

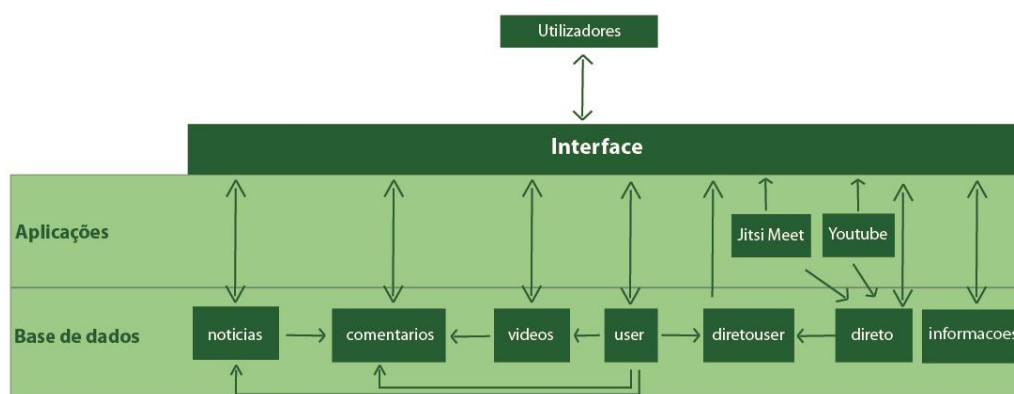


Figura 27. Arquitetura proposta para uma plataforma de emissão de vídeo 360°, com integração de tecnologias de interatividade e de colaboração em tempo real.

A base de dados foi implementada em MySQL (versão 5.0.12) e está estruturada em sete tabelas principais: "comentarios", "direto", "diretouser", "informacoes", "noticias", "users" e "videos". Cada uma destas tabelas é apresentada com maior detalhe na secção 3.3.13.3.5 deste documento.

A componente das Aplicações é constituída pelas ferramentas que foram integradas de forma a possibilitar o desenvolvimento da plataforma proposta, a saber:

- O Youtube, que permite o *stream* de conteúdos multimédia em 360°. Esta escolha deveu-se a ser a única plataforma, das que foram estudadas, que permite integrar o *stream* nos sítios web. Para além disso, há a possibilidade de integrar os vídeos na plataforma de WebRTC selecionada, a Jitsi Meet.
- A Jitsi Meet, que permite a interação em tempo real, com base na tecnologia WebRTC. A razão da escolha desta ferramenta deveu-se, principalmente, à possibilidade de poder ser criada uma instância em servidores próprios (algo que as restantes ferramentas

estudadas não permitiam). Para além disso, a Jitsi Meet suporta funcionalidades tais como o *chat*, a videoconferência e a integração de hiperligações da plataforma Youtube, o que vai ao encontro do que é pretendido com este trabalho.

Por último, a componente de Interface contém os elementos necessários para gerir e aceder às valências da plataforma proposta (e.g. gestão da base de dados, acesso às tecnologias de interação e de colaboração). Este componente foi desenvolvido com base nas *frameworks* Laravel (PHP, *back end*) e Bootstrap (HTML, *front end*), assim como na tecnologia JavaScript, nomeadamente no que respeita à sua adaptabilidade a diferentes tamanhos dos dispositivos de acesso. Na seleção de ambas as *frameworks*, mas sobretudo da de *back end*, considerou-se a sua popularidade, comunidade e documentação (Figura 9, secção 2.4.3), bem como a experiência já existente no desenvolvimento de projetos anteriores.

3.3 Implementação

Esta secção descreve o processo de implementação do protótipo desenvolvido, assim como alguns problemas encontrados e a forma de os resolver. Pretende-se, assim, constituir um manual de implementação passo a passo, que permita replicar facilmente o trabalho desenvolvido.

3.3.1 Configuração dos servidores

Em primeiro lugar foi necessário iniciar o processo de criar as máquinas virtuais para alojar e suportar a operação do protótipo: uma para a plataforma propriamente dita (cujo domínio é <http://laravel.utad.pt>) e outra para a Jitsi Meet (com o domínio <http://jitsi.utad.pt>). Nesse sentido, os Serviços de Informática e Comunicações (SIC) da UTAD disponibilizaram credenciais que permitiram o acesso aos servidores da instituição, onde opera o software de virtualização VMWare (Palo Alto, Califórnia, Estados Unidos da América). De seguida, apresenta-se o procedimento utilizado na criação e na configuração das máquinas virtuais referidas:

- 1) Selecionar o servidor e a opção “Create a new virtual machine”;
- 2) Optar por uma configuração típica (“Typical”);
- 3) Dar um nome à máquina virtual;
- 4) Definir o (ou os) disco(s) rígido(s) virtual a utilizar no servidor;

- 5) Selecionar o sistema operativo (SO) a ser instalado e configurado. Para este trabalho optou-se pelo Ubuntu Linux (64 bit);
- 6) Definir a rede "Network" a ser utilizada. Neste âmbito optou-se por uma rede existente na estrutura informática da UTAD - Virtual Local Area Network (VLAN) 42);
- 7) Quantificar o espaço a utilizar no disco escolhido em 4);
- 8) Depois de criar o servidor, é necessário editar as suas configurações ("Edit Settings"), nomeadamente definir o número de unidades centrais de processamento (CPU) e a memória Random-Access Memory (RAM). Nesta instância foram 4 e 4 GB, respetivamente;
- 9) O próximo passo é a proceder à instalação do SO, selecionando, ainda nas configurações, a interface ótica virtual (CD/DVD) e o ficheiro que contém a sua imagem, no formato ISO (Ubuntu Server v16.4);
- 10) Inicia-se a máquina virtual, selecionado "Power on the virtual machine";
- 11) Proceder-se, agora, à instalação e à configuração do SO. Nesse sentido, seleciona-se a linguagem e a opção "Install Ubuntu Server";
- 12) São selecionados o país e a linguagem de instalação pretendidos;
- 13) O passo seguinte na instalação do SO é escolher o nome para a máquina e as credenciais de superutilizador;
- 14) Segue-se a operação de particionamento do disco ("Guided - use entire disk and set up LVM");
- 15) Opta-se pelo método automático de instalação de atualizações e, em seguida, selecionam-se os softwares "LAMP Server" e "OpenSSH Server", para que sejam instalados e os serviços necessários para o alojamento web e para o funcionamento da plataforma - servidor web (APACHE), motor de bases de dados (MySQL) e a linguagem para suporte ao desenvolvimento de *back-end* PHP - disponibilizados. Num servidor que tenha outros objetivos de serviço, estas opções podem, obviamente, ser diferentes;
- 16) Instalar o gestor de arranque da máquina ("GRUB boot loader") e finalizar a instalação reiniciando a máquina virtual;

Este procedimento permite a instalação dos servidores virtuais para o protótipo implementado, mas ainda é necessária proceder a instalações de software acessório e a mais algumas configurações antes de poder dar por concluída esta tarefa. Assim, a instalação de um editor de texto é crítica para poder editar os diversos ficheiros de configuração disponíveis nos serviços

do SO. A escolha, entre os disponíveis, recaiu sobre o "Joe". Para proceder à instalação de software no Ubuntu, basta, geralmente, executar:

- 1) `sudo apt-get update`
- 2) `sudo apt-get install joe`

A configuração da *firewall* ("iptables") é o passo seguinte. Ela permite a implementação de regras de segurança, de acesso e, também, para a rede. Foi seguido o procedimento disponível em <https://wiki.debian.org/iptables>.

O penúltimo passo consiste em associar o domínio e o seu endereço de rede à máquina, de forma a que ela fique acessível quando estes forem utilizados. Assim, é editado o ficheiro das interfaces de rede utilizando o comando "`sudo joe /etc/network/interfaces`", como apresentado, para o servidor web, Figura 28.

```
# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto ens32
iface ens32 inet static
    address 193.136.42.33
    netmask 255.255.255.0
    network 193.136.42.0
    broadcast 193.136.42.255
    gateway 193.136.42.254
    # dns-* options are implemented by the resolvconf package, if installed
    dns-nameservers 193.136.40.3
    dns-search utad.pt
```

Figura 28. Ficheiro das *interfaces* do servidor web do protótipo, com a configuração de rede implementada.

Finalmente e para concluir a configuração do servidor web do protótipo, é necessário declarar os domínios através dos quais a máquina virtual pode ser acedida. Assim, procede-se à modificação dos ficheiros de *hosts*, através do comando "`sudo joe /etc/hosts`". A Figura 29 apresenta a configuração efetuada para este servidor.

```
127.0.0.1    localhost
193.136.42.33 larmiguel
193.136.42.33 laravel.utad.pt laravel
```

Figura 29. Ficheiro de *hosts* após a alteração, onde é possível ler o endereço e os domínios que estão associados ao servidor web.

3.3.2 Plataforma Jitsi Meet

Após a finalização da configuração dos servidores, o próximo passo foi a instalação da plataforma que irá suportar a interação em tempo real: a Jitsi Meet. O procedimento de instalação e de configuração seguido foi o disponibilizado na documentação oficial (<https://jitsi.org/downloads/ubuntu-debian-installations-instructions>) e que se resume a:

- 1) Instalação da chave do repositório da Jitsi no sistema, através do comando:
`wget -qO - https://download.jitsi.org/jitsi-key.gpg.key | sudo apt-key add -`
- 2) Criação do ficheiro “sources.list.d” com o repositório:
`sudo sh -c "echo 'deb https://download.jitsi.org stable/' > /etc/apt/sources.list.d/jitsi-stable.list"`
- 3) Atualização da lista de pacotes do Ubuntu:
`sudo apt-get -y update`
- 4) Instalação da plataforma Jitsi Meet:
`sudo apt-get -y install jitsi-meet`

Após a conclusão do processo de instalação, já é possível aceder à plataforma Jitsi Meet via um *browser web*. Assim e como é apresentado na Figura 30, basta colocar o domínio definido para a plataforma na barra de endereços.

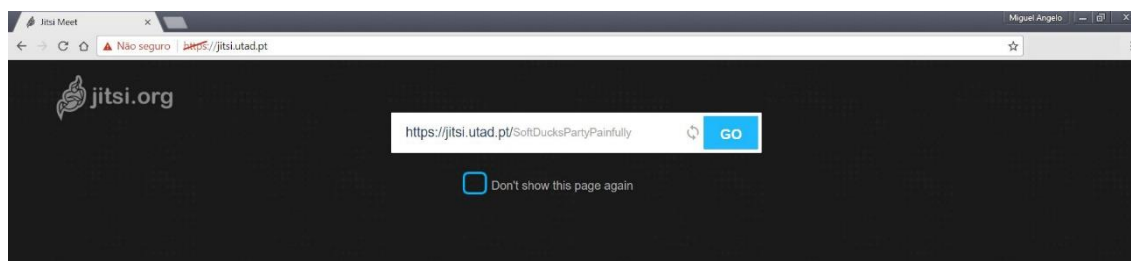


Figura 30. Página inicial da Jitsi Meet, após a conclusão da sua instalação.

É, ainda, de salientar, que durante a instalação desta plataforma surgiram alguns problemas sobretudo relacionados com a in experiência relativamente ao SO Ubuntu e quanto à configuração de servidores. De entre eles, destacam-se dois:

- A necessidade de configurar o servidor virtual com um endereço de rede estático, de forma a manter o acesso disponível independentemente da atribuição automática de endereços, via servidor de Dynamic Host Configuration Protocol (DHCP) da UTAD;

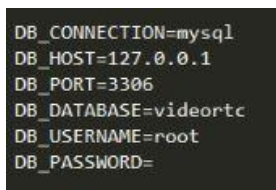
- A obrigatoriedade de proceder à instalação enquanto superutilizador do servidor, sob pena da execução dos comandos presentes no procedimento oficial de instalação da plataforma resultar em erros sucessivos.

3.3.3 Framework Laravel

O desenvolvimento do protótipo da plataforma proposta teve início após a criação e configuração dos servidores. Contudo, a opção foi desenvolver localmente, de início (SO MS Windows e utilização do XAMPP). Apenas depois de concluído foi colocado *online*.

Nesta subsecção pretende-se apresentar o processo de instalação e de configuração da *framework* PHP que suporta o protótipo, bem como explicar alguns dos elementos que a constituem e algumas das livrarias utilizadas.

Os processos de criação de um projeto, configuração, atualização e gestão de dependências da *framework* Laravel, assim como o desenvolvimento de aplicações para a web, são facilitados através da instalação do Composer (<http://getcomposer.org>), um gestor de dependências para a linguagem PHP. Após este passo e no sentido de criar um projeto base Laravel, terá de ser executado o comando "composer create-project laravel/laravel nome_pretendido_projeto --prefer-dist" no terminal do MS Windows. Em seguida, terá de ser configurada a interface com a base de dados já existente no motor de base de dados do sistema - neste caso, denominada "videortc" - através da edição do ficheiro ".env" disponível na raiz do projeto criado (Figura 31).



```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=videortc
DB_USERNAME=root
DB_PASSWORD=
```

Figura 31. Configuração do ficheiro “.env” disponível na pasta raiz do projeto Laravel criado, de forma a ser possível aceder a uma base de dados.

O próximo passo consiste em implementar a estrutura de tabelas prevista para a base de dados. Tal pode ser feito de duas formas: através da interface de gestão gráfica do motor de base de dados (phpMyAdmin) ou recorrendo a elementos da Laravel. Enquanto que a primeira forma exige a criação manual de cada tabela e dos respetivos campos, assim como a sua configuração

(e.g. tipo de dados, tamanho), através da interface web, a segunda forma baseia-se na utilização do Composer e de um elemento do Laravel: a classe Migration. Com base no terminal, este processo consiste em começar por criar uma nova migration executando o comando "php artisan make:migration nome_da_migration". Este comando gera um ficheiro na pasta "migrations", localizada na pasta "database", com o nome "nome_da_migration". Editando este ficheiro, configura-se a tabela que se pretende criar na base de dados. A Figura 32 apresenta o exemplo da tabela "comentarios".

Depois de finalizar a configuração da tabela, efetiva-se a criação da tabela na base de dados através do comando "php artisan migrate". Este processo terá de ser repetido para todas as tabelas previstas na estrutura da base de dados.

```
class CreateComentarios extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('comentarios', function (Blueprint $table) {
            $table->increments('id');
            $table->string('comentario',255);
            $table->integer('estado');
            $table->integer('id_user')->unsigned();
            $table->integer('id_noticia')->unsigned()->nullable();
            $table->integer('id_video')->unsigned()->nullable();
            $table->timestamps();

            $table->foreign('id_user')->references('id')->on('users');
            $table->foreign('id_noticia')->references('id')->on('noticias');
            $table->foreign('id_video')->references('id')->on('videos');
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::dropIfExists('comentarios');
    }
}
```

Figura 32. Exemplo de um ficheiro de configuração de uma tabela - a relativa aos comentários - realizada através do elemento Migration do Laravel.

A *framework* Laravel, como já referido, está estruturada com base no padrão MVC. Assim e para perceber melhor a sua forma de funcionamento, importa apresentar alguns dos elementos que a constituem: rotas, controladores, verificações, modelos e vistas. As rotas permitem definir

os caminhos disponíveis ao utilizador, considerando as permissões que possui. Assim, quando uma hiperligação é acedida, o ficheiro "web.php" (Figura 33), localizado na pasta "routes" do projeto, é consultado e obtido o par controlador/método previsto para lidar com o acesso efetuado. Contudo, algumas das rotas podem não estar explícitas neste ficheiro, pelo que uma forma de conhecer a totalidade das rotas é executar, na consola, o comando "php artisan route:list".

```
//frontend
Route::post('/resetpass', 'PaginasController@reset');
Route::get('/', 'PaginasController@inicio');
Route::post('/pesquisa', 'PaginasController@search');
Route::get('/novidades', 'PaginasController@novidades');
Route::get('/maislidas', 'PaginasController@maislidas');
Route::get('/cat/{cat}', 'PaginasController@categoria');
Route::get('/cat/{cat}/{id}', 'PaginasController@noticia');
Route::post('/comentar', 'PaginasController@comentar');
Route::post('/comentarvideo', 'PaginasController@comentarvideo');
Route::get('/jornalista/{id}', 'PaginasController@jornalista');
Route::get('/vid/', 'PaginasController@listvideo');
Route::get('/vid/{id}', 'PaginasController@video');
Route::get('/direto/', 'PaginasController@direto');
Route::get('/direto/entrada', 'PaginasController@entrada');
Route::get('/info/', 'PaginasController@inicio');
Route::get('/info/{id}', 'PaginasController@informacao');
Route::get('/editperfil/', 'PaginasController@editperfil');
```

Figura 33. Algumas das rotas criadas para o protótipo, onde é possível ter alguns exemplos de pares controlador/método previstos para diferentes acessos.

Já no que respeita aos controladores, estes são responsáveis por, mediante as interações do utilizador, indicar qual a visualização a apresentar e solicitar ao modelo, se for necessária, alguma ação a realizar com dados eventualmente enviados. Cabe, também, aos controladores, entre outros, gerir o acesso dos utilizadores a partes da aplicação web. A partir do Composer, podem ser criados dois tipos de controladores (Figura 34):

- Sem métodos definidos, através do comando "php artisan make_controller nome_controlador";
- Com métodos definidos, sendo, apenas, necessário, implementar cada um dos métodos. O comando utilizado é "php artisan make:controller nome_controlador -resource". Neste caso, é possível declarar, de forma automática, as rotas para este controlador, ao invés de o fazer uma a uma. Assim, no ficheiro "web.php" já referido, utiliza-se o comando "Route::resources('caminho', 'nome_controlador');", onde o "caminho" se refere à localização do controlador no projeto e o "nome_controlador" indica a designação do controlador criado.

Depois de criado o controlador, podem ser definidos e/ou modificados métodos, de acordo com as necessidades da aplicação web. A Figura 35 apresenta um exemplo de um método definido num dos controladores adstrito às notícias a serem apresentadas no protótipo.



```
<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;

class NomeController extends Controller
{
    public function index()
    {
        //
    }

    public function create()
    {
        //
    }

    public function store(Request $request)
    {
        //
    }

    public function show($id)
    {
        //
    }

    public function edit($id)
    {
        //
    }

    public function update(Request $request, $id)
    {
        //
    }

    public function destroy($id)
    {
        //
    }
}

<?php
namespace App\Http\Controllers;
use Illuminate\Http\Request;

class NomeController extends Controller
{
    //
}
```

Figura 34. Dois tipos de controladores que podem ser criados a partir da consola: à esquerda um controlador já com métodos definidos e à direita um controlador sem métodos definidos.

```
public function index()
{
    //permissoo
    if(auth()->guest() || (Auth::user()->tipo == 4)){
        return redirect('/');
    }

    // verificar se existe pedido de ordenamento da lista
    $coluna = request()->input('coluna');
    $ordem = request()->input('ordem');
    $permitido = ['titulo', 'categoria', 'name'];

    // controlo para ordenamento da lista
    if (!isset($coluna) || !in_array($coluna, $permitido)) {
        $coluna = 'titulo';
    }
    if (isset($ordem) && $ordem == 'desc') {
        $ordem = 'desc';
    } else {
        $ordem = 'asc';
    }

    $selementos = DB::table('noticias')
        ->join('users', 'noticias.id_user', '=', 'users.id')
        ->select('users.name', 'noticias.id', 'noticias.titulo', 'noticias.texto', 'noticias.categoria', 'noticias.estado', 'noticias.imagem', 'noticias.created_at')
        ->orderBy($coluna, $ordem)
        ->orderBy('created_at', 'desc')
        ->paginate(10);

    return view('admin.noticias.index', [
        'elementos' => $selementos,
        'ordem' => ($ordem == "asc" ? "desc" : "asc")
    ]);
}
```

Figura 35. Exemplo de um método desenvolvido para o protótipo. Permite, caso o utilizador esteja autenticado e seja moderador, jornalista ou administrador, acesso às notícias disponíveis no protótipo, assim como a ordená-las por título, categoria ou pelo nome de quem as criou.

De forma a conseguir interagir com uma tabela da base de dados, os controladores, enviam a informação para o modelo para este trate dessa interação.

No modelo, é possível definir qual a tabela afetada, os campos passíveis de ser alterados, os escondidos, entre outros (Figura 36). Para criar um novo modelo é necessário executar, no terminal, o comando "php artisan make:model nome_modelo". O novo modelo é, então, colocado na pasta "app" do projeto.

```
class User extends Authenticatable
{
    use Notifiable;

    protected $table = 'users';
    /**
     * The attributes that are mass assignable.
     *
     * @var array
     */
    protected $fillable = [
        'name', 'email', 'password', 'estado', 'tipo', 'facebook_id'
    ];

    /**
     * The attributes that should be hidden for arrays.
     *
     * @var array
     */
    protected $hidden = [
        'password', 'remember_token',
    ];
}
```

Figura 36. Exemplo de um modelo - o de utilizadores - desenvolvido para o protótipo.

Contudo e antes de existir uma inserção de dados em qualquer tabela da base de dados da aplicação web, é necessário que estes não só respeitem a estrutura implementada como os requisitos definidos para cada campo, como, por exemplo, o tipo de dados e o tamanho. Assim e antes de ser efetuado qualquer pedido ao modelo, é possível que os dados sejam validados utilizando um ficheiro de validações (Request). Um ficheiro para validações é criado através do comando "php artisan make:request nome_request", sendo colocado na pasta "Requests", localizada dentro da pasta "app/Http" do projeto. Nele são definidas quer as regras de validação, quer as mensagens de erro a apresentar ao utilizador. A Figura 37 apresenta um exemplo de um ficheiro de validação criado para a confirmar a sanidade dos dados aquando da inserção de novas notícias.


```

public function rules()
{
    $rules = [
        'titulo' => 'required|max:255',
        'texto' => 'required',
        'categoria' => 'required',
        'imagem' => 'required|mimes:jpeg,bmp,png,PNG|max:2048'
    ];

    return $rules;
}

public function messages()
{
    return [
        'titulo.required' => 'É necessário inserir um título ',
        'titulo.max' => 'É necessário ter no máximo 255 caracteres',
        'texto.required' => 'É necessário inserir a notícia',
        'imagem.required' => 'É necessário inserir uma imagem',
        'imagem.mimes' => 'A imagem só pode ser do formato jpg, png, bmp',
        'imagem.max' => 'A imagem só pode ser no máximo 2 MB'
    ];
}

```

Figura 37. Exemplo de um ficheiro de validação desenvolvido para confirmar a sanidade dos dados, aquando da introdução de uma nova notícia. São, ainda, definidas as mensagens de erro a apresentar ao utilizador, caso estes venham a ocorrer.

É, ainda, necessário associar o ficheiro de validação ao método do controlador onde se pretenda que a validação ocorra. Nesse sentido e como é apresentado na Figura 38, é necessário, na definição do método, inscrever o nome do ficheiro de validação.

```

/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @return \Illuminate\Http\Response
 */
public function store(Requests\CriarNoticiasRequest $request)
{

```

Figura 38. Associação de uma validação - o relativo às notícias - ao método correspondente.

Depois de lidar com as interações do utilizador, o controlador, por norma, apresenta, através de uma visualização, o resultado da interação. Usualmente, é o próprio controlador a disponibilizar a informação que deve ser apresentada ao utilizador. Apesar de a *framework* Laravel ter a sua forma de o fazer, pode, também, ser utilizado o método tradicional no PHP. A Figura 39 apresenta um exemplo com as duas formas de apresentar o conteúdo de uma variável.

```

<?php echo $noticia->titulo; ?>

{{ $noticia -> titulo }}

```

Figura 39. Impressão de uma variável utilizando o “echo” do PHP, em cima, e utilizando a forma que o Laravel utiliza, em baixo. Ambas apresentam o mesmo resultado final.

Por fim e no que concerne ao conteúdo de cada página web, a framework Laravel possui uma ferramenta de gestão de *templates*: o Blade. Através dele, é possível criar páginas-tipo, com secções alimentadas pelos ficheiros que possuem os conteúdos a serem apresentados (Figura 40). Dentro de cada ficheiro de conteúdos, basta ter uma referência à página tipo e à secção a que o conteúdo se refere, como mostra a Figura 41. As páginas da Blade têm de ter a extensão “.blade.php” e estão normalmente armazenadas na pasta “resources/views”. De forma a criar estes ficheiros, basta na pasta das “views” criar um ficheiro novo e ao guardar o ficheiro guardar com a extensão “.blade.php”.

```
<section id="main-content">
  <section class="wrapper">

    @yield('content')

  </section>
</section>
```

Figura 40. Exemplo de como se declara uma secção numa página-tipo, através da utilização da Blade.

```
{!! Form::open(['url'=>'/diretos','enctype'=>'multipart/form-data', 'files'=>true]) !!}

<div class="form-group">
  {!! Form::label('titulo', 'Titulo:', array('class' => 'col-md-12 obrigatorio control-label')) !!}
  <div class="col-md-12">
    {!! Form::text('titulo', '', array('class' => 'form-control ')) !!}
  </div>
</div>

<div class="form-group">
  <label class="control-label obrigatorio col-md-12">Descrição:</label>
  <div class="col-md-12">
    <textarea class="form-control ckeditor" name="descricao" rows="6"></textarea>
  </div>
</div>

<div class="form-group">
  {!! Form::label('link', 'Link do direto:', array('class' => 'col-md-12 obrigatorio control-label')) !!}
  <div class="col-md-12">
    {!! Form::text('link', '', array('class' => 'form-control ')) !!}
    <span class="help-block"><strong>Nota:</strong>https://www.youtube.com/watch?v=<strong>5IACwIZBvO4</strong></span>
  </div>
</div>

<div class="form-group">
  {!! Form::label('estado', 'Estado:', array('class' => 'col-md-12 obrigatorio control-label')) !!}
  <div class="col-md-12">
    <select name="estado" class="form-control m-bot15">
      <option value="1">Iniciado</option>
      <option value="2">Finalizado</option>
    </select>
  </div>
</div>

<div class="form-group">
  {!! Form::label('chat', 'Sala de Chat:', array('class' => 'col-md-12 obrigatorio control-label')) !!}
  <div class="col-md-12">
    {!! Form::text('chat', '', array('class' => 'form-control ')) !!}
  </div>
</div>

<!-- Botao de submit-->
<div class="form-group">
  <p class="help-block">* - Campos obrigatórios</p>
  <div class="col-md-12">
    {!! Form::submit('Criar direto', ['class'=>'btn btn-primary ']) !!}
  </div>
</div>

{!! Form::close() !!}
```

Figura 41. Exemplo de um ficheiro de conteúdo, onde é possível ver, na primeira linha, a referência à página-tipo e na restante página, a referência à secção da página-tipo e ao conteúdo a nela inserir.

No que respeita ao desenvolvimento do protótipo, foram utilizados alguns componentes da *framework* Laravel, instalados via Composer. Destacam-se:

- Laravel Collective: permite a implementação rápida e intuitiva de formulários (HTML), assim como a sua ligação aos respetivos controladores. A Figura 42 apresenta o exemplo do formulário desenvolvido para a criação de uma transmissão, em direto, de um evento;

```
 {!! Form::open(['url' => '/diretos', 'enctype' => 'multipart/form-data', 'files' => true]) !!}

<div class="form-group">
  {!! Form::label('titulo', 'Titulo:', array('class' => 'col-md-12 obrigatorio control-label')) !!}
  <div class="col-md-12">
    {!! Form::text('titulo', '', array('class' => 'form-control ')) !!}
  </div>
</div>

<div class="form-group">
  <div class="control-label obrigatorio col-md-12">Descrição:</div>
  <div class="col-md-12">
    <textarea class="form-control ckeditor" name="descricao" rows="6"></textarea>
  </div>
</div>

<div class="form-group">
  {!! Form::label('link', 'Link do direto:', array('class' => 'col-md-12 obrigatorio control-label')) !!}
  <div class="col-md-12">
    {!! Form::text('link', '', array('class' => 'form-control ')) !!}
    <span class="help-block"><strong>Nota:</strong><strong>https://www.youtube.com/watch?v=<strong>5IACwIZBV04</strong></span>
  </div>
</div>

<div class="form-group">
  {!! Form::label('estado', 'Estado:', array('class' => 'col-md-12 obrigatorio control-label')) !!}
  <div class="col-sm-12">
    <select name="estado" class="form-control m-bot15">
      <option value="1">Iniciado</option>
      <option value="2">Finalizado</option>
    </select>
  </div>
</div>

<div class="form-group">
  {!! Form::label('chat', 'Sala de Chat:', array('class' => 'col-md-12 obrigatorio control-label')) !!}
  <div class="col-md-12">
    {!! Form::text('chat', '', array('class' => 'form-control ')) !!}
  </div>
</div>

<!-- Botao de submit -->
<div class="form-group">
  <div class="help-block">* - Campos obrigatórios</div>
  <div class="col-md-12">
    {!! Form::submit('Criar direto', ['class' => 'btn btn-primary ']) !!}
  </div>
</div>

{!! Form::close() !!}
```

Figura 42. Exemplo do formulário desenvolvido para a criação de uma transmissão, em direto, de um evento, com base no componente Laravel Collective. Poderá, também, ser observada, a integração do formulário no HTML.

- Laravel Socialite: permite fazer autenticação através das redes sociais: Facebook, Twitter, Google, LinkedIn, GitHub e Bitbucket;
- Authentication Quickstart: permite, através do Composer, criar a base (migrations, controller, views, model) de todo o sistema de autenticação. Para isso, basta aceder ao terminal do projeto e executar o comando “php artisan make:auth”. De forma a combinar o Socialite e o Authentication Quickstart, é necessário fazer algumas alterações no modelo (Figura 43) e na migração (Figura 44), criados por omissão ao executar a linha de código apresentada.

```
protected $table = 'users';
/**
 * The attributes that are mass assignable.
 *
 * @var array
 */
protected $fillable = [
    'name', 'email', 'password', 'estado', 'tipo', 'facebook_id'
];

/**
 * The attributes that should be hidden for arrays.
 *
 * @var array
 */
protected $hidden = [
    'password', 'remember_token',
];
```

Figura 43. Modelo referente ao utilizador, após as alterações introduzidas no sentido de combinar os módulos Socialite e Authentication Quickstart. Foi adicionado o campo "facebook_id", sendo que os campos "estado" e "tipo" estão relacionados com os valores acerca do estado e do tipo de utilizador.

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->increments('id');
        $table->string('name');
        $table->string('email')->unique();
        $table->string('password')->nullable();
        $table->integer('estado');
        $table->integer('tipo');
        $table->integer('facebook_id')->nullable();
        $table->rememberToken();
        $table->timestamps();
    });
}

/**
 * Reverse the migrations.
 *
 * @return void
 */
public function down()
{
    Schema::dropIfExists('users');
}
```

Figura 44. Migração referente ao utilizador, onde foi necessário adicionar o campo "facebook_id" para guardar o valor de identificação da rede social Facebook do utilizador.

Após o desenvolvimento das duas plataformas, foi necessário uni-las. Numa primeira fase, a solução passava por usá-las em separado e criar uma ligação para aceder a uma sala de conversação, na Jitsi, relativa à transmissão em direto a ser visualizada. Isto porque a Jitsi permite a integração de vídeo da plataforma Youtube, o que permitiria criar nela a visualização do evento em direto e, também, a interação entre utilizadores e entre utilizadores e responsáveis pela emissão. Contudo, devido a um problema que será detalhado na secção 4.1.2 deste documento, foi necessário encontrar uma alternativa.

Assim, a solução passou por integrar a sala de comunicação, em tempo real, na própria página da transmissão do evento, através da etiqueta "<object>" do HTML. Esta etiqueta permite, entre outros, integrar um sítio web dentro de outro, pelo que foi, apenas, necessário, utilizar o atributo "data" e colocar lá a hiperligação para a sala da Jitsi adstrita à transmissão em direto do evento, como é apresentado na Figura 45.

```
@if(auth()->guest())
@else
    @if(auth()->user()->estado == 1)
        <a onclick="myfunction()" class="img-rounded entdireto" > Esconder/Mostrar Chat</a>
        <object id="myDIV" type="text/html" width="100%" height="400vh" data="https://jitsi.utad.pt/{{$direto->chat}}">
        </object>
    @else
        <div class="alert alert-danger">
            Encontra-se bloqueado, por favor se quiser visualizar a televisão colaborativa resolva a situação com a administração
        </div>
    @endif
@endif
```

Figura 45. Elemento <object> do HTML, em que é possível ver o atributo data com a ligação para a sala da Jitsi criada para o direto. O comando "\$direto->chat" imprime o nome da sala criada para o direto. Caso o utilizador seja, apenas, um visitante, não tem a opção de ter apresentada a sala de conversação.

Foi, ainda, possibilitada ao utilizador, a valência de esconder a sala de conversação, caso apenas queira visualizar a transmissão, em tempo real, do evento (Figura 46).

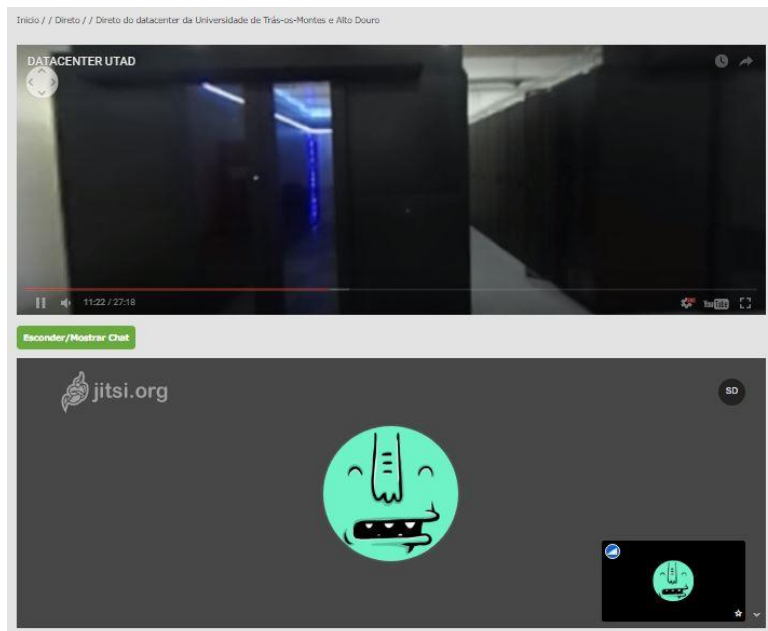


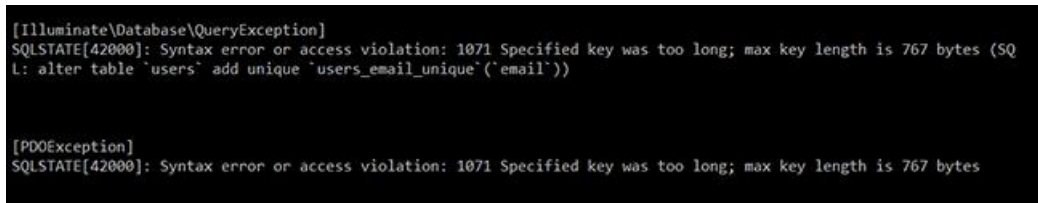
Figura 46. Integração das plataformas de transmissão e de interação, onde é possível, também, ver a opção que existe para esconder a sala de conversação, caso o utilizador assim o entenda.

Para finalizar o processo de implementação do protótipo é necessário colocá-lo online e disponível para o público em geral. Assim, foi necessário transferir os ficheiros do ambiente de

desenvolvimento local para o remoto, através de um cliente de File Transfer Protocolo (FTP) e colocá-los na pasta web da máquina virtual ("/var/www/html/"). Neste passo, deve ter-se o cuidado de transferir, igualmente, o ficheiro ".htaccess", que é muito importante para o funcionamento da aplicação. De seguida, é necessário alterar as permissões - atribuindo permissão de escrita - das pastas da aplicação "storage" e "vendor", assim como as pastas onde serão colocados os elementos multimédia a serem disponibilizados na plataforma: "img" e "video". Como último passo, o ficheiro ".env" da aplicação deve ser configurado, nomeadamente no que concerne aos dados de acesso à base de dados da máquina cuja estrutura, entretanto, deverá ter sido devidamente importada do ambiente de desenvolvimento local, utilizando, por exemplo, o interface gráfico de gestão do motor de bases de dados, o phpMyAdmin.

Neste processo, importa, ainda, destacar dois problemas que surgiram. O primeiro foi encontrado durante a criação das tabelas para a base de dados: ao executar o comando "php artisan migrate", surgia o erro "Syntax error or access violation" (Figura 47). Este erro não era apresentado em versões anteriores da *framework* Laravel. Com efeito, a partir da versão 5.4 da *framework*, a base de dados tem, por omissão, a codificação utf8mb4. Isto poderá causar erros em aplicações que recorram ao MySQL com versão inferior à 5.7.7. A solução passa por editar o ficheiro "AppServiceProvider.php", localizado na pasta "Http/Providers" e alterar o método "boot", para ficar tal como é apresentado na Figura 48.

O segundo problema surgiu quando se tentou criar uma miniatura após a inserção de um novo vídeo no protótipo. A solução passou por instalar um software - o "ffmpeg" - no servidor para o conseguir fazer. A Figura 49 apresenta a configuração necessária à criação de uma miniatura a partir de um vídeo colocado no protótipo utilizando o "ffmpeg".



```
[Illuminate\Database\QueryException]
SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was too long; max key length is 767 bytes (SQL: alter table `users` add unique `users_email_unique` (`email`))

[PDOException]
SQLSTATE[42000]: Syntax error or access violation: 1071 Specified key was too long; max key length is 767 bytes
```

Figura 47. Erro de “syntax error or access violation”, que ocorre ao tentar fazer a migração para a criação das tabelas da base de dados.

```

class AppServiceProvider extends ServiceProvider
{
    /**
     * Bootstrap any application services.
     *
     * @return void
     */
    public function boot()
    {
        Schema::defaultStringLength(191);
    }
}

```

Figura 48. Alteração necessária ao método "boot" do ficheiro "AppServiceProvider.php" para resolver o problema identificado e apresentado na Figura 47.

```

//CRIAÇÃO DO THUMB DO VIDEO
// local onde esta situado o ffmpeg, no servidor web por norma esta em /usr/bin/ffmpeg
$ffmpeg = 'C:\ffmpeg\bin\ffmpeg.exe';
//destino do video
$video = $destinationPathImg.$filenameImg;
//nome da imagem e destino
$nameImg = str_random(3).'thumb'.str_random(3).'.jpg';
$image = public_path().'/video/thumbs/'.$nameImg;
//intervalo de tempo para tirar a screenshot
$interval = 1;
//tamanho da imagem
$size = '640x480';
//comando para gerar a miniatura
$cmd = "$ffmpeg -i $video -deinterlace -an -ss $interval -f mjpeg -t 1 -r 1 -y -s $size $image 2>&1";
$return = `$cmd`;

```

Figura 49. Configuração necessária para criar uma miniatura a partir de um vídeo colocado no protótipo, utilizando o software "ffmpeg".

3.4 Emissão direto 360° de evento

Nesta secção pretende-se descrever o processo de transmissão, em direto, de um evento, em 360°, para as plataformas Youtube e Facebook, bem como os desafios encontrados na seleção do hardware necessário.

3.4.1 Escolha da câmara 360°

Para efetuar a transmissão 360°, em direto, de eventos, foram testadas duas câmaras 360°: a Fly HD (360fly, USA) e a Theta S (Ricoh Company, Japão). A primeira câmara estava disponível na instituição, o que permitia não adquirir novo hardware para realizar este trabalho. Contudo, percebeu-se que apenas é possível realizar transmissões em direto utilizando a câmara Fly 4K (judiosan, 2016). No sentido de confirmar esta informação, foram feitos alguns testes e pesquisas sobre o tema. A Figura 50 apresenta algumas imagens adquiridas utilizando a câmara Fly HD.



Figura 50. Imagens adquiridas com a câmara Fly HD, enquanto decorriam os testes para testar a viabilidade de realizar a transmissão 360°, em direto, de um evento.

Durante a pesquisa realizada, foram encontrados alguns dos motivos que impedem a realização de uma transmissão em direto utilizando esta câmara. Destaca-se:

- Ser necessário ter o *firmware* 2.1.4 para ser possível usar a câmara como *webcam* e, assim, poder-lhe aceder para realizar a transmissão em direto (Figura 51).

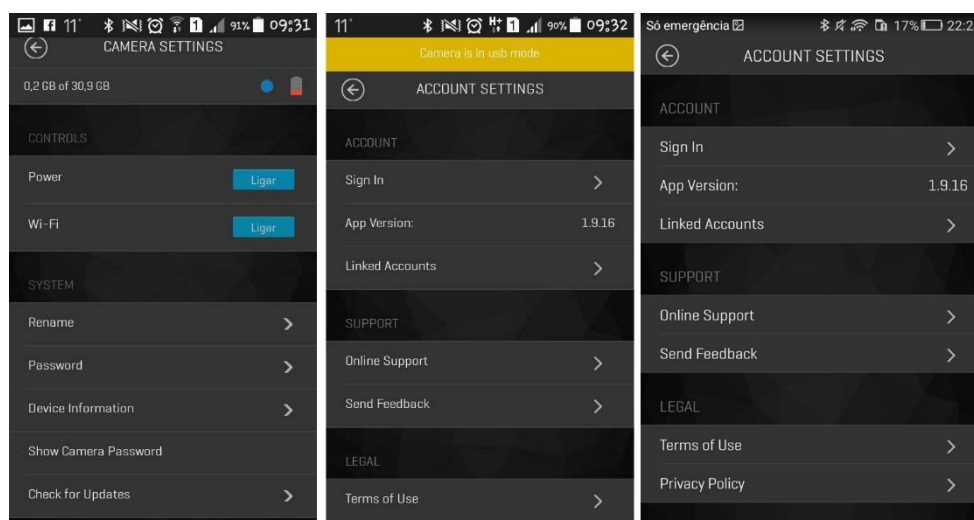


Figura 51. Imagens onde pode ser verificado que não existe a opção de *webcam* (1.^a imagem), e em que tanto em Android (2.^a imagem) como em iPhone (3.^a imagem) a versão do *firmware* é a 1.8.16 e não a necessária 2.1.4. Não foi possível, também, atualizar o *firmware* da câmara FlyHD.

- Não ser possível fazer *stream* de vídeo em 360°, mesmo com a Fly 4K (craftyazz, 2017), mas sim com duas faixas de 180° (Figura 52).



Figura 52. Transmissão realizada com a câmara Fly 4k onde é possível verificar que a imagem é dividida em duas imagens de 180°. Imagem retirada de um comentário de craftyazz, 2017, no fórum da própria empresa, 360fly.

- O formato de saída não ser suportado pelas plataformas Youtube e Facebook para uma transmissão em direto (potro777, 2016), já que é necessário ter uma imagem de formato retangular. Contudo, a imagem de saída desta câmara é circular, tal como é referido na Figura 53.

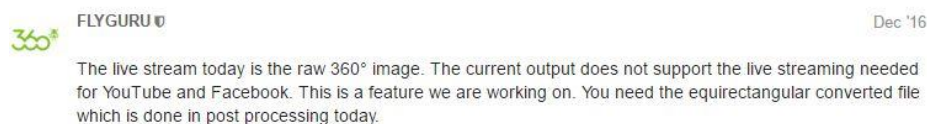


Figura 53. Comentário onde é explicado o tipo de saída da câmara Fly HD e o motivo pelo qual não é suportado pelo Facebook e Youtube (potro777, 2016). Tradução: “(...)O atual output não suporta a transmissão em tempo real para o YouTube e o Facebook. (...). É necessário que o ficheiro seja retangular, o que, atualmente, é, apenas, obtido, no pós-processamento.”

Não sendo possível utilizar a câmara existente - Fly HD - para o objetivo pretendido, foi necessário estudar alternativas que respeitassem, simultaneamente, os requisitos do projeto e as restrições financeiras existentes para efetuar uma aquisição. Assim, surgem como opções: Theta S, ALLie (ALLie, USA), Orah (VideoStitch, USA) e Samsung Gear 360 (Samsung, Japão). A opção foi para a câmara Theta S. As razões foram o preço mais acessível, estava disponível no continente Europeu (a alternativa, nestas condições, era a Samsung Gear 360) e já existiam testes realizados, comprovando ser possível realizar a transmissão 360°, em direto, de eventos,

como, por exemplo, a transmissão dos sorteios referentes à época 2017/2018 da Liga dos Campeões (UEFA, 2017a) e da Liga Europa (UEFA, 2017b).

Após a aquisição da câmara foram realizados alguns testes fotográficos para perceber como operá-la e, também, qual a qualidade da imagem obtida em diferentes cenários: exterior com boa iluminação (Figura 54), interior com pouca luminosidade (Figura 55) e interior com boa luminosidade (Figura 56).



Figura 54. Fotografia de exterior, com boa luz, captada com a câmara Theta S, no exterior da Residência de Estudantes Além-Rio da UTAD, em Vila Real.

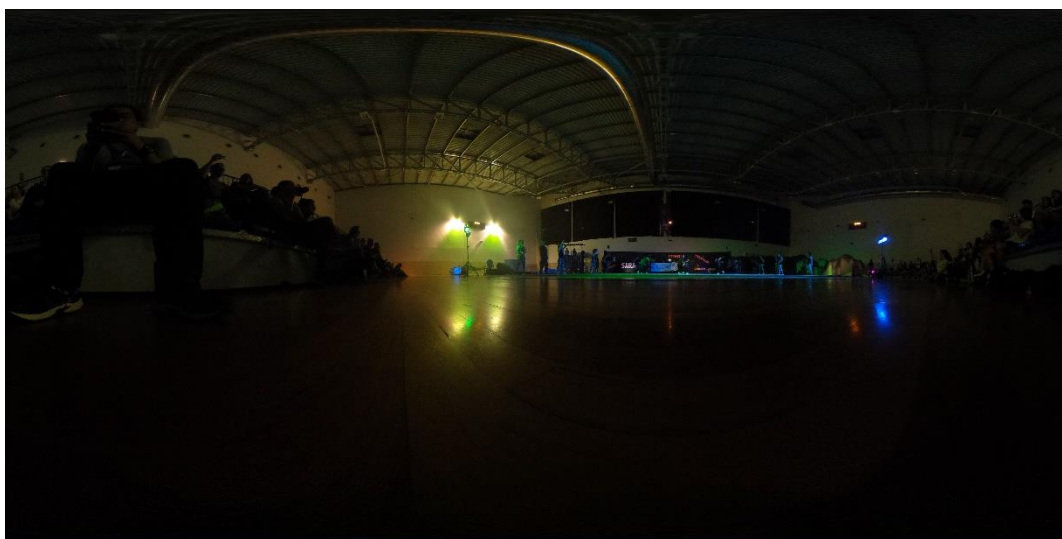


Figura 55. Fotografia de interior, com fraca luminosidade, captada com a câmara Theta S, no interior do Pavilhão dos Desportos de Vila Real, durante o XX Sarau de Atividades Gímnicas e Academias.



Figura 56. Fotografia de interior, com boa luminosidade, captada com a câmara Theta S, no Laboratório de Artes Visuais e Multimédia da UTAD, durante uma aula da unidade curricular de Web TV, do Mestrado em Comunicação e Multimédia.

3.4.2 *Encoder*

Para possibilitar a realização de uma transmissão 360° em direto, é necessário recorrer a um software de codificação (*encoder*), para proceder à captura de conteúdo multimédia - através de câmaras, microfones e/ou captura de ecrã - e enviá-lo para onde será emitido. Há diversas opções como, por exemplo, AirServer, Epiphan Webcaster, Wirecast e XSplit. Contudo, por ser de código-aberto e cumprir os requisitos estabelecidos para este projeto, a escolha recaiu no Open Broadcaster Software (OBS). Após instalar este *encoder*, é necessário obter e instalar a aplicação de *streaming* a partir do sítio web do fabricante da câmara Theta S. O procedimento subsequente para configurar a câmara e o OBS é:

- 1) Iniciar o OBS;

- 2) Iniciar a Theta S, pressionando, em simultâneo, os 1.º e 3.º botões. Assim é ativada a luz de Live (Figura 57) e pode, de seguida, ser ligada ao computador;



Figura 57. À esquerda, uma imagem que apresenta os dois botões que é necessário acionar para colocar a câmara no modo direto e à direita, apresenta-se a imagem da câmara em modo direto.

- 3) Adicionar um “dispositivo de captura de vídeo” (Figura 58).

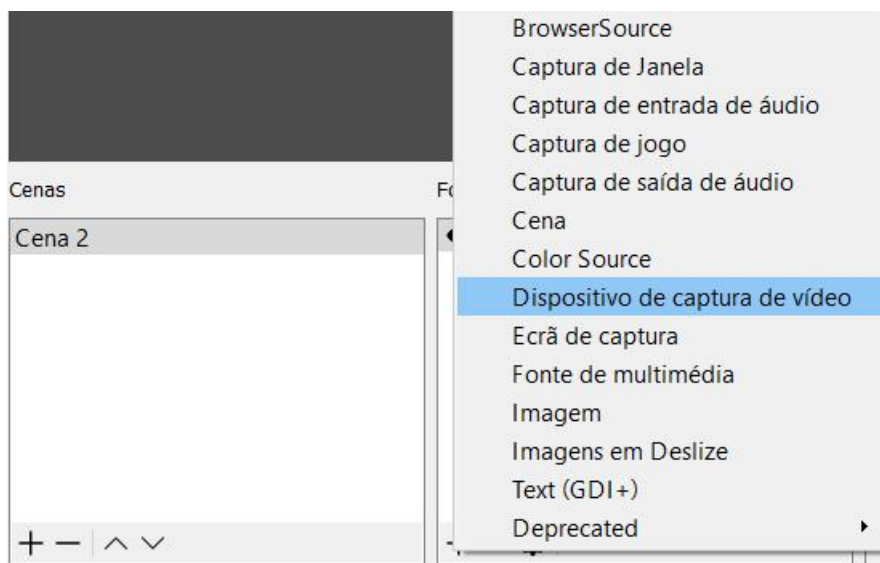


Figura 58. Localização, no OBS, da opção para adicionar um dispositivo de captura de vídeo.

- 4) Em “Dispositivo”, escolher “THETA UVC FULL HD BLENDER” ou “THETA UVC HD BLENDER”, consoante a transmissão for em Full HD ou, apenas, HD (Figura 59).

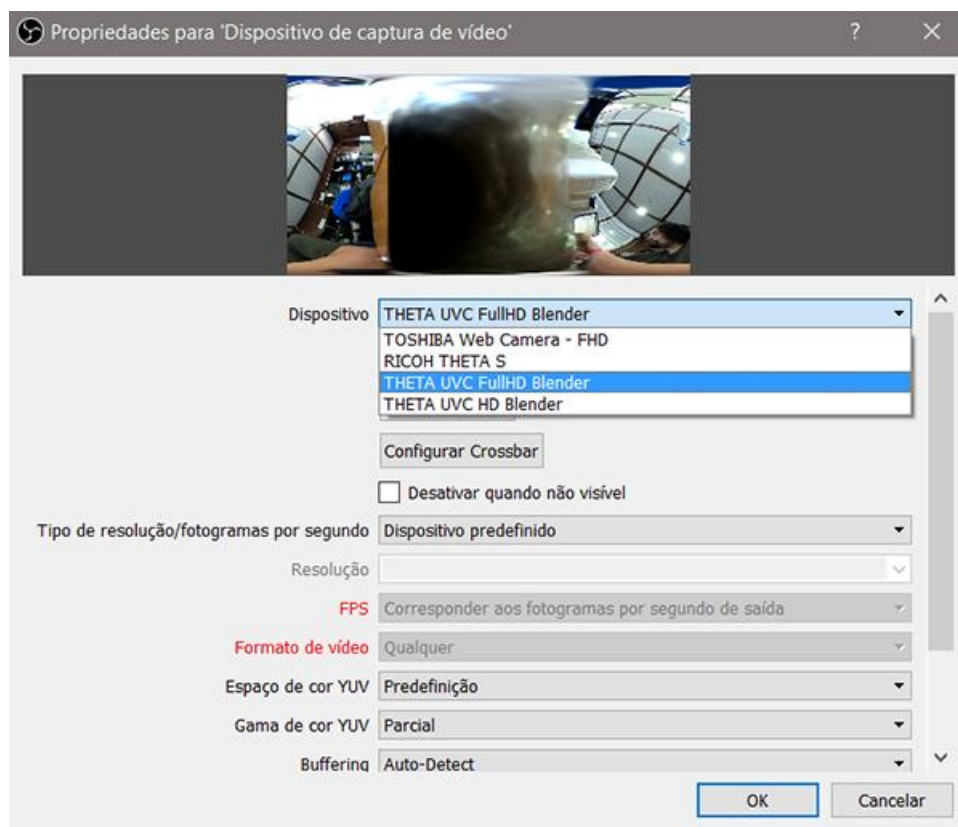


Figura 59. Localização, no OBS, onde se seleciona a câmara que se deseja, e, no caso da Theta S, o modo de transmissão: HD ou Full HD.

- 5) Uma das soluções para evitar a existência de uma faixa preta no fundo da imagem capturada consiste em esticar a imagem. A deformação que daí resulta não é perceptível aquando da visualização da transmissão. Para fazer esta operação, seleciona-se, com o botão direito do rato, o dispositivo de captura inserido e escolhe-se a opção "Transformar->Esticar o Ecrã), como é apresentado na Figura 60;
- 6) Por fim e dependendo da plataforma através da qual se pretende transmitir - Youtube ou Facebook - o procedimento é diferente e está detalhado nas secções 3.4.3. e 3.4.4.

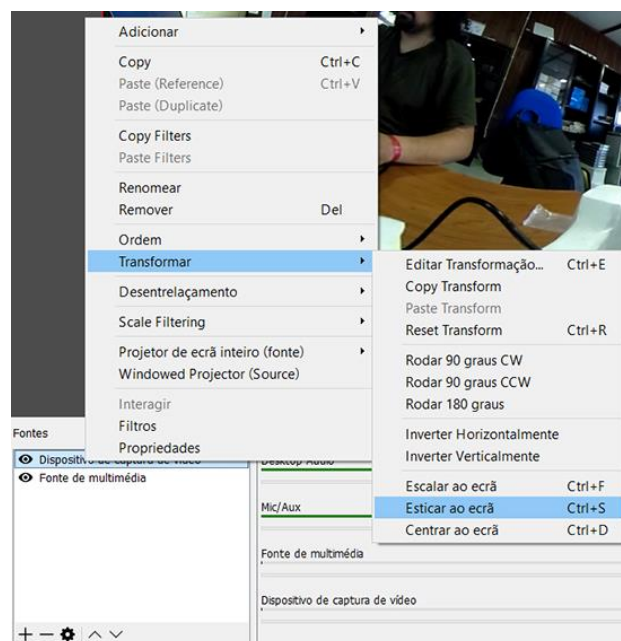


Figura 60. Localização da opção para esticar a imagem da câmara Theta S, de forma que esta ocupe o ecrã todo. Esta transformação não é notada aquando da visualização da transmissão em direto.

É importante garantir que o *firmware* da câmara Theta S está, no mínimo, na sua versão 1.82. Caso contrário, não é possível aceder à imagem da câmara e realizar a transmissão 360° em direto. Para fazer esta verificação e, se necessário, atualizar o *firmware*), é necessário obter, a partir do sítio web do fabricante, e instalar a aplicação da câmara no computador. Depois, basta aceder ao menu "File->Firmware Update" para verificar a versão e, se necessário, atualizá-la (Figura 61 e 62).

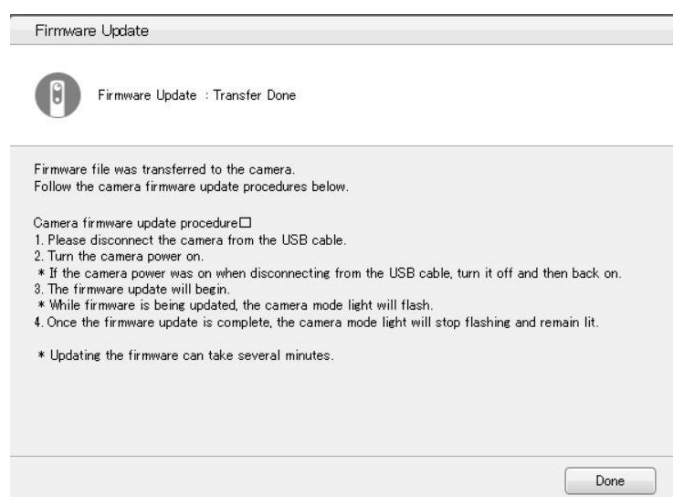


Figura 61. Ecrã que contém as instruções necessárias para atualizar o *firmware* da câmara Theta S.

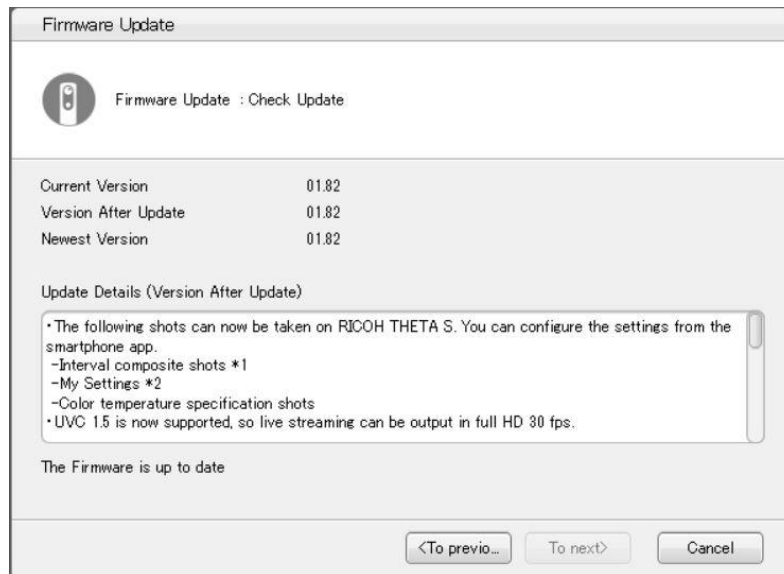


Figura 62. Ecrã que surge quando o *firmware* da Theta S é o mais recente.

3.4.3 Transmissão do direto 360° através do Facebook

Nesta subsecção detalha-se o procedimento para uma transmissão 360°, em direto, através do Facebook:

- 1) Aceder a “<https://www.facebook.com/live/create>” e seleccionar a opção “Criar transmissão em direto”;
- 2) Seleccionar o local, no Facebook, onde se pretende que a transmissão seja partilhada e escolher, também, o modo 360° (Figura 63);



Figura 63. Página, no Facebook, onde é possível escolher o local de publicação e seleccionar que a transmissão é em 360°.

- 3) Escrever a descrição e o título da transmissão. Em seguida, copia-se a chave de transmissão para colocar no OBS, de forma a estabelecer a ligação entre o Facebook e o OBS (Figura 64);



Figura 64. Página, no Facebook, onde é possível escolher a descrição e o título da transmissão em direto, assim como aceder à chave de transmissão.

- 4) Após copiar a chave é necessário voltar ao OBS, selecionar "Definições", escolher "Transmissão" e, lá, "Facebook Live" como serviço. Depois, cola-se a chave em "Chave de transmissão" e seleciona-se "Aplicar", seguido de "OK", como é apresentado na Figura 65;

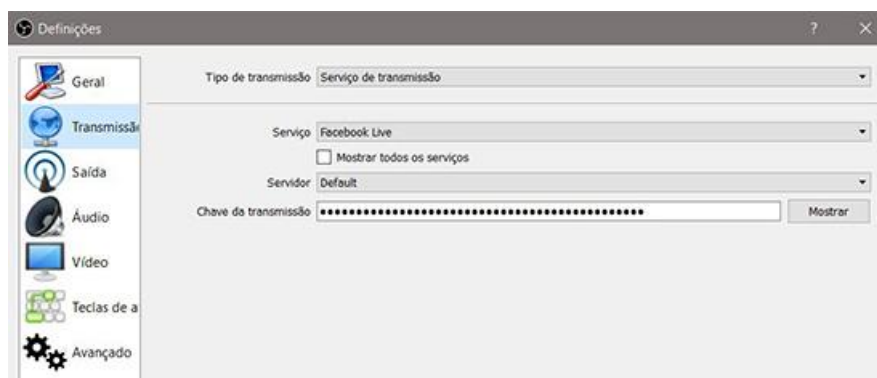


Figura 65. Ecrã, no OBS, onde se deve escolher o serviço de transmissão e colocar a respetiva chave, para estabelecer a ligação entre o Facebook e o OBS.

- 5) Por fim, escolher "Iniciar Transmissão", ir ao Facebook e esperar até que seja possível ver a transmissão (Figura 64). Quando isto acontecer, selecionar "Transmitir em Direto".

A partir desse momento a transmissão para o Facebook já está a ocorrer. É, então, possível visualizar quem aderiu ao direto, qual o número de pessoas e interagir com elas, entre outros (Figura 66).

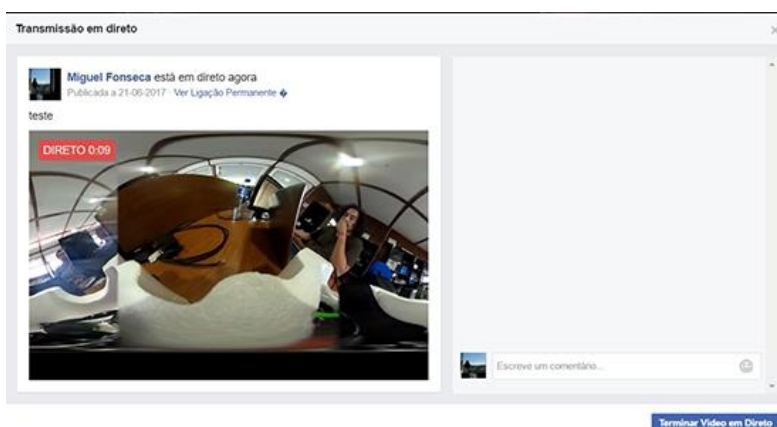


Figura 66. Ecrã da transmissão no Facebook, onde é possível ver o direto, o número de utilizadores atuais, quem aderiu, os comentários e responder aos mesmos, assim como terminar o direto.

3.4.4 Transmissão do direto 360° através do Youtube

Nesta subsecção é descrito o procedimento para a transmissão 360° de um evento, em direto, através da plataforma Youtube:

- 1) Aceder a “https://www.youtube.com/my_live_events” e seleccionar em “Agendar um novo evento”;
- 2) Introduzir as definições básicas da transmissão (Figura 67);

Figura 67. Página, no Youtube, onde podem ser inserir as informações básicas da transmissão 360°, em direto.

- 3) Seleccionar a secção definições avançadas e a *stream* em 360° (Figura 68). Depois, seleccionar "Criar Evento";

Mostrar Todos

Ordenar por Comentários principais

☒ Os utilizadores podem visualizar as classificações deste vídeo

Vídeo de 360° ?

☒ Esta é uma transmissão em direto de 360°

DVR

☒ Ativar DVR (Os visitantes podem recuar até 4 horas durante a transmissão)

Figura 68. Local, nas definições avançadas, da criação da transmissão para o Youtube, onde se deve seleccionar que a transmissão será em 360°.

- 4) Escolhe-se a taxa de bits referentes a transmissão e depois copia-se o código onde diz “nome da transmissão” de forma a estabelecer-se a ligação entre o Facebook e o OBS (Figura 69);

Escolha a taxa de bits máxima sustentada do seu codificador *

Selecione um intervalo que corresponda à saída contínua máxima fornecida pelo seu codificador.

☒ Carregamento básico

1500 Kbps - 4000 Kbps (720p)

☐ Ativar 60 fps ?

☐ Carregamento personalizado ?

Selecione o seu codificador

O YouTube Live fornece suporte para uma variedade de codificadores. Selecione uma das opções de

Outros codificadores

1. Configure o seu codificador

Taxas de bits e definições recomendadas ?

2. Copiar e colar no seu codificador

Introduza os nomes de transmissão em fluxo contínuo e URLs nas opções de configuração do software de codificação.

Nome da Transmissão

we3f-xyqh-d8bd-ezg2

URL do Servidor Principal

rtmp://a.rtmp.youtube.com/live2

URL do Servidor de Cópias de Segurança

rtmp://b.rtmp.youtube.com/live2?backup=1

Figura 69. Página, no Youtube, onde é possível escolher a taxa de bits e onde é acedida a chave da transmissão para colocar no OBS.

- 5) Após copiar a chave, é necessário voltar ao OBS, seleccionar "Definições" seguido de "Transmissão" e escolher "Youtube" como serviço. De seguida, colar a chave em "Chave de transmissão" e seleccionar "Aplicar", seguido de "OK" (Figura 70);

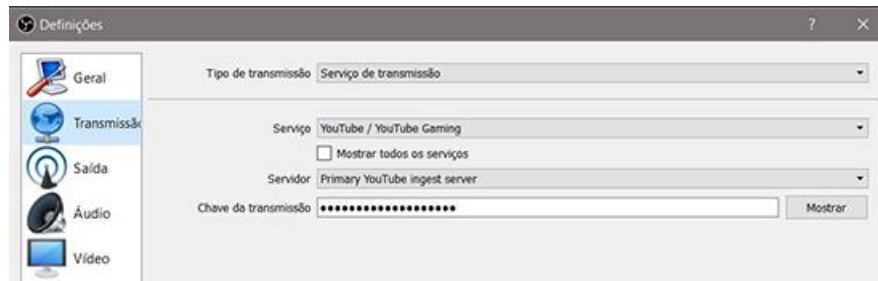


Figura 70. Ecrã, no OBS, onde se deve escolher o serviço de transmissão e colocar a chave de transmissão de forma a fazer a ligação entre o Youtube e o OBS.

- 6) Seleccionar "Iniciar Transmissão", ir ao Youtube e escolher a secção "Sala de controlo em direto";
- 7) Seleccionar "Pré-visualizar" (Figura 71) e esperar até que seja possível iniciar a transmissão;



Figura 71. Local onde é possível aceder a pré-visualização da transmissão no Youtube de forma a confirmar se está tudo de acordo com o pretendido.

- 8) Quando for possível, basta clicar em "Iniciar Transmissão em Fluxo Contínuo" (Figura 72) e a transmissão passa a estar disponível;



Figura 72. Local onde pode dar-se início à transmissão para o Youtube, após a opção estar disponível.

Quando a transmissão estiver finalizada, basta, na secção “Sala de controlo em direto” seleccionar o botão “Parar a Transmissão em Fluxo Contínuo” (Figura 73).

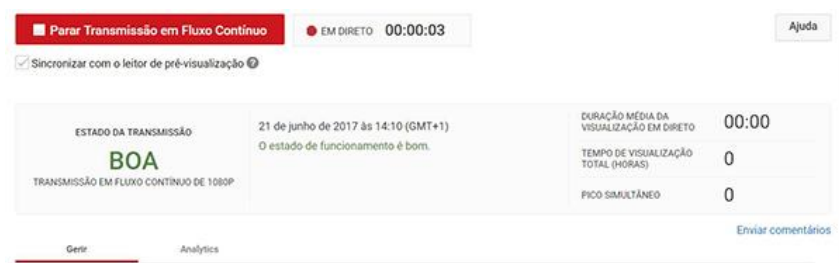


Figura 73. Local onde é possível parar a transmissão no Youtube.

3.4.5 Problemas

Nesta subsecção serão expostos os maiores contratemplos encontrados durante a fase de testes e antes da realização dos casos de uso: o suporte para a câmara e o facto de a câmara aquecer e desligar-se após algum tempo de transmissão.

Quando se adquiriu a câmara Theta S, não se viu a necessidade de comprar o tripé próprio para a mesma e quando se testaram os tripés existentes constatou-se um problema: ao colocar a Theta S no tripé tornava-se impossível colocar o cabo micro USB de forma a conseguir conectar a câmara ao computador. Para tentar resolver este contratempo e conseguir realizar os primeiros testes, foram desenvolvidos alguns apoios artesanais para o efeito (Figura 74).



Figura 74. Suportes criados para a Tetha S: o primeiro criado a partir de esferovite, o segundo aproveitando um copo de plástico e o terceiro a partir de um mini-tripé.

No entanto, após testadas várias alternativas, foi testado um tripé que suportava uma das luzes de estúdio e verificou-se que serviria para a realização dos casos de uso, pois era muito estável e era possível colocar a câmara e o cabo micro USB (Figura 75).



Figura 75. Suporte usado para a câmara Theta S, no primeiro caso de uso, utilizando um tripé de luzes de estúdio.

No dia anterior ao primeiro caso de uso a temperatura rondou os 31°C. Ao realizar um último teste de longa duração, constatou-se que a câmara, a transmitir em *Full HD*, funcionava durante cerca de 21 min e em *HD* cerca de 50 min e que depois deixava de transmitir (Figura 76). No entanto, nas pesquisas realizadas acerca do assunto, foram encontrados diretos com duração de 10 horas (Westrate, 2016) e 5 horas (BoinxSoftwareLtd, 2016) a partir da Theta S, pelo que foi necessário ter cuidado com a temperatura no local e com o tempo que o direto iria durar, de forma a escolher a melhor opção para o realizar e proporcionar uma boa experiência a quem estivesse a assistir.

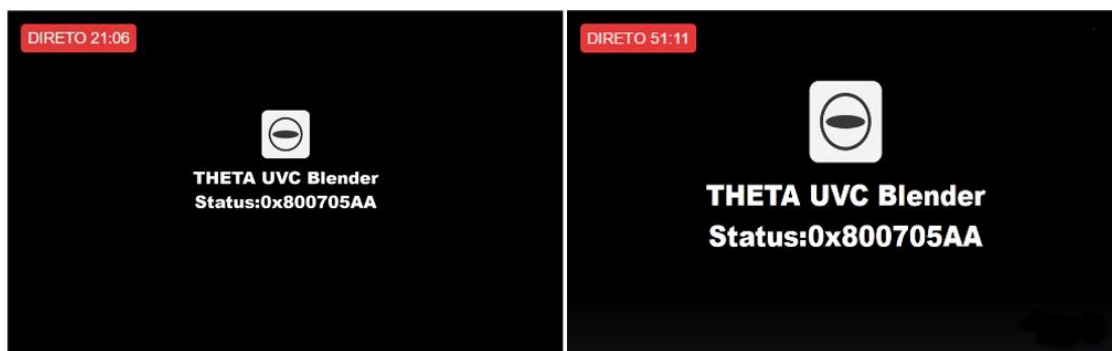


Figura 76. Ecrã após a câmara se desligar, devido ao aquecimento. À esquerda, em transmissão Full HD e à direita transmissão em HD.

Apenas como nota final, pois não pode ser considerado um contratempo, durante os testes realizados foi, ainda, testada, por curiosidade, a passagem da transmissão por uma mesa de

transmissão, ao invés de passar diretamente da câmara para o computador. No entanto, nesse teste foi possível verificar, que, para já, tal não é possível, pois a imagem que chegava ao *encoder* era em formato circular e não retangular (Figura 77).



Figura 77. Imagem que retrata a passagem da transmissão primeiro por uma mesa de transmissão e só depois para um computador. Nela, também pode ser visto que a imagem que chega ao codificador é circular e não retangular.

4. Casos de estudo

Neste capítulo é apresentada toda a informação relativa dos casos de estudo realizados no âmbito desta dissertação: o contexto, o material utilizado, os dados obtidos, os contratempos e as conclusões que puderam observar em cada um deles.

4.1 Centro de dados da UTAD

O primeiro caso de uso foi realizado a 26 de junho de 2017 e consistiu na transmissão a partir do centro de dados da UTAD de modo a perceber como a plataforma Jitsi se comportaria com vídeo 360, com acessos realizados por dispositivos móveis, estudando o impacto criado/interação com os utilizadores.

4.1.1 Equipamento

De forma a ser possível realizar a transmissão foi necessário colocar o todo equipamento dentro do centro de dados (Figura 78):

- Câmara Theta S: utilizada para a captura do vídeo em 360°;
- Suporte das luzes de estúdio: usado de forma a manter a câmara ao nível do olho humano;
- Lenovo Z51: computador utilizado na realização da transmissão;
- OBS: *encoder* usado para a transmissão do direto para o Youtube;
- Cabo mini USB: usado para fazer a ligação entre a câmara e o computador de forma a passar as imagens capturadas para o computador;

- *Playlist da Spotify* - Para esconder o som ambiente e para evitar que pudessem ouvir algumas conversas a ocorrer dentro do centro de dados.



Figura 78. Disposição do equipamento utilizado na transmissão do Centro de Dados da UTAD.

4.1.2 Dados

A transmissão teve a duração total de aproximadamente 4 horas - 3 h e 34 min na primeira transmissão e 24 min a segunda transmissão – tendo um total de 73 visualizações - 62 na primeira e 11 na segunda. Quatro dessas visualizações, duas em cada transmissão, equivaleram a apenas 1 pessoa, pois eram do computador que estava a efetuar a transmissão, para além de outro computador que estava a ser utilizado para a verificação da transmissão. Houve a necessidade da existência de um moderador para as possíveis interações. Durante a fase de testes, tal como referido na secção 3.4.5, foi detetado um problema de um possível aquecimento da câmara durante a transmissão. Foi registada a temperatura ambiente, constatando que esta se encontrava entre 19°C e 19,5°C. A transmissão, a hiperligação da plataforma Jitsi, foi publicitada apenas na rede social Facebook, sendo que primeiramente foi partilhada numa página pública (Figura 79) e depois partilhado para um perfil e grupo privado. Essa publicação teve um alcance de 252 pessoas (Figura 80), sendo que as reações à transmissão foram 6 *likes* e alguns comentários a alertar para um problema que estava a acontecer na visualização da transmissão pela Jitsi, pois não era possível mudar o ângulo de visão.



Figura 79. Publicitação da transmissão do Centro de Dados, assim como a publicação da hiperligação, em direto, no Youtube, após resolução do problema encontrado com a plataforma Jitsi.



Figura 80. Alcance da publicação (252 pessoas) do caso de uso sobre o Centro de Dados da UTAD.

4.1.3 Problemas

Durante a realização da transmissão foram encontradas algumas limitações, que levaram a reavaliação de algumas opções e ponderar soluções de forma a criar transmissões mais sólidas e estáveis. Os problemas encontrados foram:

- Instabilidade da rede wireless: em determinados instantes a rede wireless estava muito instável, levando a algumas falhas na transmissão, levando a que pontualmente a imagem de vídeo não fosse fluida;
- Ao testar a transmissão no telemóvel, ao contrário do que acontecia quando foi analisada a plataforma, foi necessário a instalação de uma aplicação de forma a ser possível aceder à transmissão;
- Ao visualizar o vídeo 360 na plataforma Jitsi, inicialmente apenas quem criou a transmissão poderia alterar o ângulo de visualização, sendo que aos restantes utilizadores era apresentado sempre o mesmo ângulo de visão sem possibilidade de alteração. Mesmo após alguma pesquisa não foi possível perceber a razão do problema, visto que até na plataforma demo apresentada no sitio web da Jitsi o problema verificava-se. No entanto, quando a plataforma foi analisada, esse problema não acontecia sendo que qualquer utilizador podia escolher o seu ângulo de visão;
- A transmissão (www.youtube.com/watch?v=Mg67znVfBYg) foi bloqueada por causa dos direitos das músicas da lista da spotify, após cerca de 3h e 34 min.

No decorrer da transmissão, tentou-se resolver alguns destes problemas. No primeiro e segundo ponto não foi possível retificar os mesmos, não foi possível utilizar a internet wireless nem evitar a instalação da aplicação. No terceiro ponto, para resolver o problema, foi divulgada a hiperligação direta para a transmissão no Youtube de forma a que os utilizadores pudessem aceder à mesma (Figura 79). Por último, o quarto ponto, a solução encontrada foi a criação de uma segunda transmissão (www.youtube.com/watch?v=88cYuir4f2o), sem áudio.

4.1.4 Conclusões do caso de estudo

Após a realização do caso de uso, olhando para os dados e problemas observados, durante a transmissão, foi possível concluir que:

- É necessário pensar melhor no conteúdo da transmissão em 360°. Embora possa, o centro de dados da UTAD, ser um tema que desperte algum interesse, não é algo que atraia muitas pessoas e os levem a interagir e/ou convidar outros indivíduos a aderirem à transmissão. Não é uma experiência que “prenda” alguém muito tempo;
- É necessário ter cuidado com a utilização de conteúdos de terceiros. Com o uso destes conteúdos, e o consequente bloqueio da transmissão pode levar que o utilizador se sinta inseguro ou desconfortável, não lhe proporciona uma boa experiência, possivelmente o contrário;
- É importante utilizar uma rede estável e com largura de banda suficiente, na transmissão do conteúdo em 360°. Uma rede instável leva a que haja falhas e pausas na transmissão causando uma experiência negativa para a assistência;
- Evitar o uso da reprodução de hiperligações de conteúdos em 360° na Jitsi, pois em casos de mudanças e atualizações na plataforma pode levar a que funcionalidades não se comportem corretamente, ou deixem de funcionar e isso pode levar a que quem esteja a assistir à transmissão tenha uma experiência não satisfatória.

4.2 Conversa acerca do 48º Circuito Internacional de Vila Real na Universidade FM

No âmbito do 48º Circuito Internacional de Vila Real, surgiu a oportunidade, no dia 5 de Julho de 2017, de efetuar um direto em 360º para o Facebook, durante uma conversa na radio Universidade FM, com varias pessoas ligadas às corridas: Pedro Mendes (Chicane Motores), Rafael Lobato (Piloto), Filipe Fernandes (Vice-Presidente da Associação Promotora do Circuito Internacional de Vila Real), Eduardo Ferreira (Chefe de Segurança do Clube Automóvel de Vila Real), Eduardo Santelmo (Membro da Associação Promotora do Circuito Internacional de Vila Real e elemento dos Castra Dura), Ricardo Fortuna (Membro da Associação Promotora do Circuito Internacional de Vila Real e elemento dos Castra Dura) e Luís Mendonça (Radio Universidade FM). Esta conversa teve ainda a Andreia Pinto como moderadora. A verificação do direto e o apoio no que tocava à interação nas redes sociais foi facultado por nós (Figura 81).



Figura 81. Fotografia final dos intervenientes na conversa acerca do 48º Circuito Internacional de Vila Real. Da esquerda para a direita: Andreia Pinto, Filipe Fernandes, Eduardo Ferreira, Ricardo Fortuna, Rafael Lobato, Pedro Mendes, Eduardo Santelmo e Luís Mendonça.

Inicialmente, os intervenientes encontravam-se um pouco retraídos e tímidos, mas com o decorrer da transmissão e a interação do publico, os convidados começaram a ficar mais “soltos”, tornou-se numa conversa descontraída e mais familiar, onde foi possível abordar diversos temas: o WTCC, a “joker lap”, a segurança do circuito, a situação do Tom Coronel, as expetativas para o 48º Circuito Internacional de Vila Real, a perspetiva de um piloto acerca de vários assuntos, o passado, presente e futuro das corridas, como o publico, e em especial os habitantes de Vila Real vivem as corridas, entre outros temas.

Este caso de uso, serviu para verificar o interesse das pessoas no direto 360º e observar o tipo de interação que as redes sociais podem ou não criar numa transmissão em direto.

4.2.1 Equipamento

De forma a ser possível a realização desde direto foi necessário a utilização do seguinte equipamento (Figura 82):



Figura 82. Disposição do material utilizado na transmissão da conversa relativa ao 48º Circuito Internacional de Vila Real.

- Theta S: utilizada para a captura do vídeo em 360°;
- Capacete de Rafael Lobato e o suporte artesanalmente criado a partir do mini tripé (Figura 83): usado de forma a manter a câmara ao nível do olho, não foi possível fazê-lo apenas com o mini tripé;
- Toshiba Satellite P50-A-120: computador utilizado na realização da transmissão;
- Cabo de rede: de forma a obter uma melhor qualidade de rede para realizar a transmissão;
- OBS – *encoder* usado para a transmissão do direto para o Facebook;
- Mini USB - usado para realizar a ligação entre a câmara e o computador para transferir as imagens capturadas para o computador.



Figura 83. Suporte usado para a transmissão do direto da conversa acerca do 48º Circuito Internacional de Vila Real na Universidade FM.

4.2.2 Dados

Devido à proximidade das corridas e a pedido de alguns intervenientes, foi necessário antecipar a transmissão um dia em relação à data prevista, de forma a que estes se pudessem concentrar na mesma, pelo que a campanha publicitária da emissão apenas foi possível durante a tarde em que a transmissão ia ocorrer. Para o efeito a radio Universidade FM criou 5 publicações (Figura 84) em que se destaca a emissão e os diversos convidados presentes na mesma.



Figura 84. Exemplo de uma das publicidades criadas, pela Universidade FM, de forma a publicitar a transmissão da conversa acerca do 48º Circuito Internacional de Vila Real.

Estas publicações tiveram um alcance de cerca de 10 400 pessoas, 92 gostos e 23 partilhas (Figura 85).



Figura 85. Algumas das partilhas da publicidade referente a transmissão da conversa acerca do 48º Circuito Internacional de Vila Real na Universidade FM.

A transmissão teve uma duração de 1 hora e 21 minutos com uma média entre as 30/40 pessoas a visualizar o direto. Teve um alcance total de 18500 pessoas, 3900 visualizações, 123 reações, 5 partilhas, 35 comentários e 3 mensagens privadas. Devido ao problema, detetado durante a fase de testes relativamente ao possível aquecimento da câmara durante a transmissão, foi registada a temperatura do ambiente, verificando-se que esta se manteve nos 25/26°C. Apenas como curiosidade, dessas 3900 visualizações podemos destacar a presença do Presidente da Câmara de Vila Real Rui Santos, e do piloto Manuel Fernandes (Figura 86).

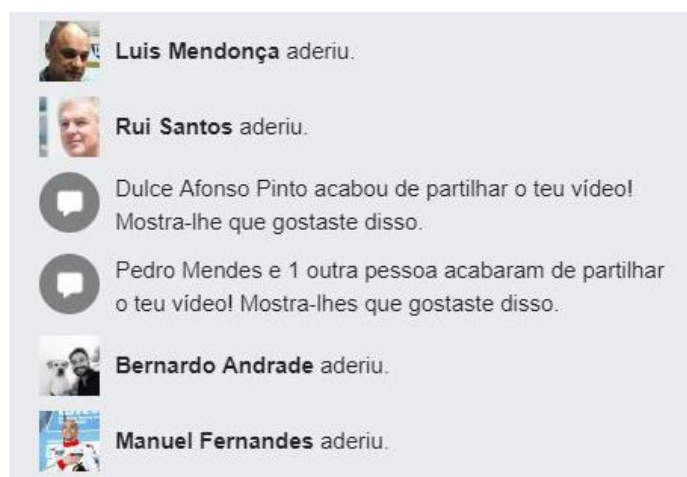


Figura 86. Algumas das entradas no direto, em que podemos ver a presença do Presidente da Câmara Rui Santos e do Piloto Manuel Fernandes. Podem, ainda, ser observadas algumas das partilhas efetuadas.

No que concerne à interação com o público, é possível destacar dois tipos: as *private joke* e as perguntas realizadas aos intervenientes.

Quando temos algum conhecido, amigo ou colega de trabalho a participar em algum tipo de concurso/entrevista a tendência natural é de espreitar e ver como as coisas estão a correr. Com a participação dessas pessoas mais próximas dos intervenientes, acabaram por surgir também algumas *private joke* (Figura 87). Esse tipo de intervenções, fez com que houvesse momentos de descontração ao longo do programa, o que levou a que os convidados estivessem mais relaxados e à vontade para responder às perguntas e a partilhar histórias.

No início da conversa, o mote da conversa, o 48º Circuito Internacional de Vila Real, foi dado pela locutora, Andreia Pinto, mas com o avançar da emissão começaram a surgir as perguntas (Figura 88) e reações/comentários às respostas pelos convidados (Figura 89), criando assim, um ambiente mais interativo entre quem os presentes no estúdio e os utilizadores que estavam a visualizar a transmissão.



Figura 87. Alguns comentários mais descontraídos, que permitiram criar um ambiente mais familiar.

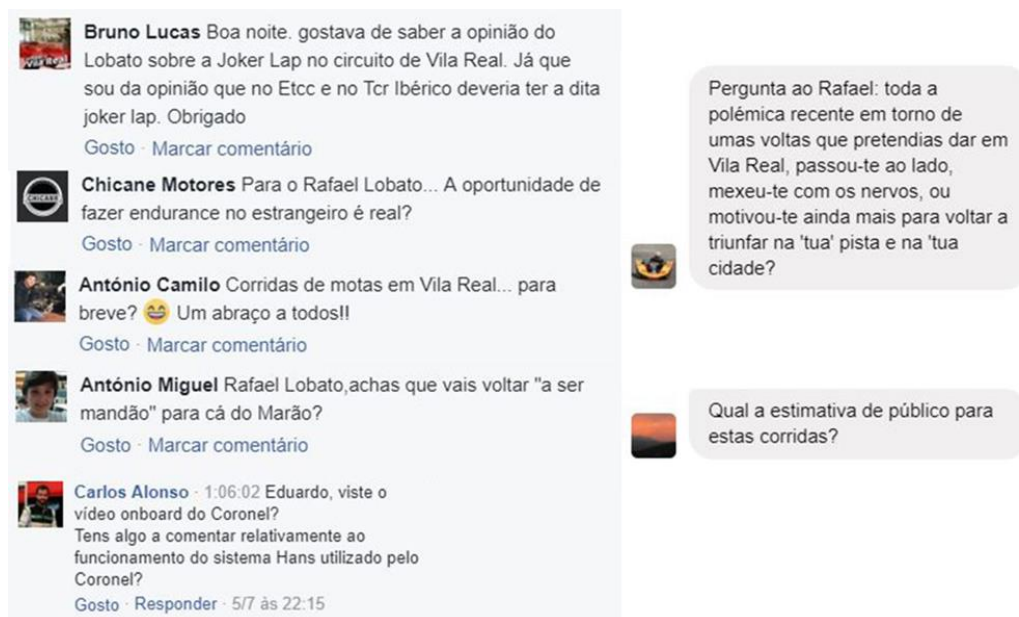


Figura 88. Extrato de perguntas realizadas aos convidados sugeridas pelos espetadores. À esquerda, perguntas realizadas a partir da caixa de comentários da transmissão, e à direita, perguntas realizadas a partir de mensagens privadas recebidas ao longo da transmissão.

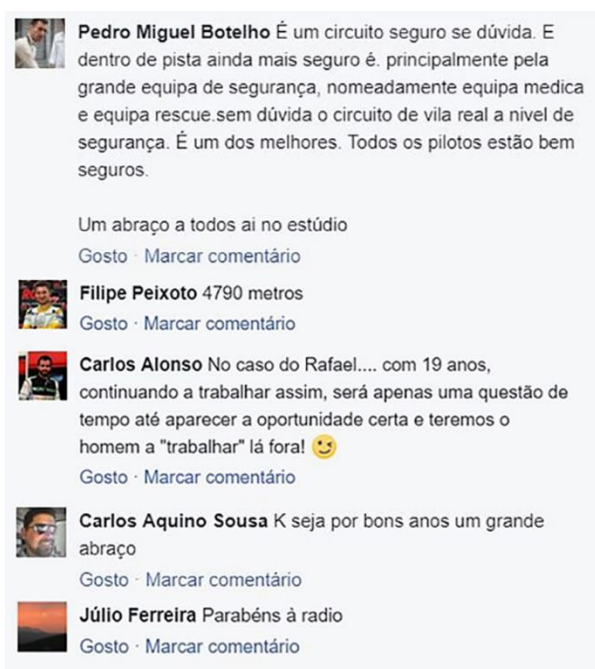


Figura 89. Algumas das reações ao longo da transmissão. Curiosidade para o segundo comentário, que foi uma correção, por parte de uma pessoa ligada ao traçado do percurso, acerca do comprimento total do percurso.

Com o fim da transmissão e já no dia seguinte, em algumas conversas privadas, foi obtido algum *feedback* por parte das pessoas que interagiram na emissão, salientando o carácter interessante e diferente da experiência. Um utilizador, que estava a tentar visualizar com o

telemóvel, achou a experiência estranha e sentiu-se um pouco desconfortável por ter que rodar sobre ele próprio para que pudesse mudar a sua perspetiva.

Uma semana após a transmissão, foram de novo consultadas as estatísticas relativas à transmissão de forma a perceber o tipo de interação que existiu pós transmissão, e sem ter havido qualquer publicitação após esta se ter realizado (Figura 90).



Figura 90. Imagem referente às interações obtidas após cerca de uma semana depois da transmissão, onde é possível consultar o número do alcance, visualizações, reações, comentários e partilhas.

Estes dados são apresentados na Tabela 9, de forma a perceber a evolução da interação por parte dos espetadores.

Dados	Final da transmissão (5/7/2017)	Uma semana após a transmissão (10/7/2017)
Alcance	18514 pessoas	25710 pessoas
Visualizações	3900	5500
Reações	123	147
Partilhas	5	13
Comentários	35	36

Tabela 9. Comparação dos valores obtidos através da plataforma Facebook após o final do caso de uso “Conversa acerca do 38º Circuito Internacional de Vila Real na Universidade FM” e decorrida uma semana.

De realçar o aumento de: cerca de 7000 pessoas alcançadas, 1600 visualizações, 24 reações e 9 partilhas. Sendo que apenas houve um aumento de um comentário à transmissão. Podemos verificar que existiu um aumento do interesse por parte dos utilizadores do Facebook, pela transmissão, dado o aumento das visualizações e principalmente pelo aumento das partilhas, que quase triplicaram após a transmissão.

4.2.3 Problemas

Em comparação ao caso de uso anterior, ocorreram menos problemas no decorrer da transmissão, mas ainda assim tivemos contratemplos:

- A conexão a Internet foi o maior problema encontrado, pois o estúdio onde se realizou o direto estava distante do *router wireless*, o que fez com que nos testes efetuados a ligação *wireless* apresentou-se fraca por vezes inexistente, o que não permitia que a transmissão se realizasse. De forma a resolver esta situação, foi feita uma tentativa de utilizar um cabo de rede em vez da ligação *wireless*. No entanto, as tomadas de rede do estúdio não permitiam a ligação de computadores não registados no sistema. Depois de alguns testes, chegou-se à conclusão que a única alternativa seria realizar uma ligação direta (configuração manual) ao *router* de forma a ser possível aceder à rede, mas devido à distância entre o estúdio e o local do *router* foi difícil conseguir um cabo que cobrisse tal distância (Figura 91). Mas, graças ao Engenheiro António Costa dos SIC da UTAD, foi possível conseguir esse cabo para ter Internet no computador que realizou a transmissão. Após mais ou menos 1 hora e 13 min de transmissão a Internet falhou por uns momentos (Figura 92) voltando pouco tempo depois, sem a necessidade de os utilizadores voltarem a conectar-se.



Figura 91. À esquerda, a distância que tinha que ser percorrida pelo cabo de rede de forma a ligar o router ao computador. À direita, os cabos de rede e um Hub fornecidos pelos SIC, para tentar solucionar o problema.



Figura 92. Falha de rede na transmissão para o Facebook, sendo que, pouco tempo depois, a rede foi restabelecida e a transmissão voltou ao ativo.

- A qualidade do áudio da transmissão. Como foi utilizado um cabo USB não foi possível transmitir o áudio para o *encoder* através da câmara Theta S. Também não foi possível passar o áudio diretamente da radio para o computador. Apenas foram encontradas duas soluções: o uso do microfone interno do computador, o que não era viável visto a distância e a distribuição física das pessoas. A utilização do áudio da *stream online* da radio foi a solução optada. Para isso, apenas foi necessário abrir no browser um separador com a *stream online* da transmissão da radio e no misturador do OBS apenas permitir que o áudio transmitido fosse o do “Desktop” (Figura 93). A utilização desta alternativa criou um ligeiro *delay* entre a imagem e o áudio.



Figura 93. Misturador de Áudio do OBS onde é possível ver que apenas o som vindo do *Desktop* está ativo.

- Pensar melhor na forma de como o moderador acede à interação dos espetadores. Em determinados momentos, como apenas existia um ecrã para controlar a interação com o público e ao mesmo tempo controlar a qualidade da emissão, criaram-se momentos de atrapalhação entre a Andreia e a parte técnica, pois ambos pretendiam aceder ao mesmo local ao mesmo tempo. Uma possível forma de resolver a situação seria a utilização de dois ecrãs possibilitando estruturar melhor a distribuição da informação acerca da transmissão e da interação. Uma solução semelhante na *stream* de jogos *online* em que, por norma, utilizam dois ecrãs e que um serve como o ecrã de jogo e no outro tem toda a informação referente à interação com o público. Neste caso poder-se-ia estruturar num ecrã tudo que fosse interação, sala de *chat* e mensagens privadas, no outro a informação relativa a *stream*, a *preview* da *stream* e o OBS para verificar os dados da mesma.

4.2.4 Conclusões do caso de estudo

Após a realização do caso de uso, olhando para os dados e problemas obtidos, durante a transmissão, foi possível concluir que:

- Apesar da existência de algumas deficiências que podem ser limadas em futuras experiências, podemos concluir que foi uma experiência bastante positiva, pois criou-se uma dinâmica diferente e teve boa aceitação e interação por parte dos espetadores e intervenientes;
- Embora não seja possível generalizar apenas com o caso de uso realizado, mas analisando o *feedback* obtido ao longo da transmissão, verifica-se que é possível utilizar o direto em 360°, neste tipo de situações, entrevista/conversas, em que o vídeo não é o principal meio de comunicação, mas sim um acessório. Outro tipo de eventos como os debates ou palestras, onde existe a possibilidade de perguntas/respostas, poderá ser também uma área segura, no entanto requer mais testes, pois, permite a existência de um moderador que pode intercalar as perguntas existentes no local com as perguntas

dos espetadores, na perspetiva do espetador poder questionar os intervenientes e ver o cenário envolvente;

- Devem ser realizados testes acerca da usabilidade e acessibilidade dos sistemas de visualização do 360, pois embora já existam soluções, é necessário começar a ter melhores soluções e/ou dar varias possibilidades de visualização ao utilizador. Por exemplo, no caso dos dispositivos moveis, dar a possibilidade de ser o próprio utilizador a escolher como quer a experiência, se a rodar sobre si ou se através da interação no ecrã, como acontece dos *browsers*.

5 Conclusões e trabalhos futuros

Podemos concluir que, numa visão geral, os objetivos desta dissertação foram atingidos pois foi possível planificar e construir uma plataforma que juntasse a transmissão de conteúdos multimédia 360° em direto e a interação, em tempo real. O protótipo desenvolvido está disponível em: <http://laravel.utad.pt/>.

Neste trabalho foram ainda realizados dois casos de uso de forma a entender como funcionaria a interação do espetador com a transmissão em 360°.

No primeiro caso, foi realizado uma transmissão do Centro de Dados da UTAD, onde foi possível verificar que a aceitação foi reduzida por parte do público. Isto poderá ter-se devido à pouca divulgação do evento, haver um interesse reduzido por parte das pessoas que acederam, mas principalmente julgamos que o grande problema foi não existirem diferentes e múltiplos pontos de interesse ao longo da transmissão, ou seja, na transmissão apenas era possível ver o ambiente dentro do centro de dados, que pouco ou nada se alterava ao longo do tempo, sendo normal que isso não prenda quem visualize. Para além disso, foi possível ter uma melhor noção dos cuidados a ter numa transmissão deste género, tais como, qualidade da Internet, a temperatura ambiental ser um fator importante para o caso da Theta S e ter cuidado com os direitos de autor do conteúdo que é incluído na transmissão.

No segundo caso, foi realizado uma entrevista a várias pessoas que estavam ligadas ao 48° Circuito Internacional de Vila Real, onde já foi possível ter uma perceção de como a interação social pode beneficiar a envolvimento entre o público e a transmissão. Foram várias as pessoas, que ao longo da transmissão, fizeram comentários e colocaram questões que ajudaram a levar a conversa para os assuntos que as pessoas tinham interesse em ver comentados ou discutidos.

Após a realização de várias experiências, julgamos ser possível a continuação da investigação nesta área.

Esse trabalho futuro, prende-se com a criação do nosso próprio sistema de transmissão em 360° e de um sistema de comunicação em tempo real, utilizando a tecnologia WebRTC. Isto porque, com base nos problemas detetados ao longo das experiências, estar a depender de aplicações de terceiros pode resultar em problemas. Se queremos um sistema que sabemos que seja imune a falhas, por causa de elementos de terceiros, provavelmente teremos que evoluir nesse sentido, isto é, na criação de um sistema próprio. Para além da criação do nosso próprio sistema, ainda existem outras áreas onde podemos testar a tecnologia do direto em 360°. No estado da arte foi possível perceber que existem várias áreas onde esta tecnologia pode ser aplicada, como por exemplo, eventos desportivos, concertos, debates, videoconferência. Como se pode perceber pelo caso de uso referente ao 48º Circuito Internacional de Vila Real, a utilização desta tecnologia em debates ou entrevistas pode ser benéfico tanto para o espetador como para a transmissão em si, pois vai mais de encontro com o tipo de tópicos que o espetador está mais sensível. Por isso, ainda é possível realizar experiências em outras áreas de forma a perceber como a transmissão em 360° funcionará nessas áreas e principalmente como enquadra a interação social nessas transmissões, e se essas serão benéficas ou não.

6. Referências Bibliográficas

- 1stWebDesigner. (2015). Web Frameworks: Pros And Cons Of Using Frameworks. Obtido 15 de Fevereiro de 2017, de <https://1stwebdesigner.com/web-frameworks/>
- Apostolopoulos, J., Chou, P., Culbertson, B., Kalker, T., Trott, M., & Wee, S. (2012). The Road to Immersive Communication. *Proceedings of the IEEE*, 100(4), 974–990. <https://doi.org/10.1109/JPROC.2011.2182069>
- appear.in. (2017). About appear.in. Obtido 20 de Abril de 2017, de <https://appear.in/information/team/>
- Banerjee, S. (2016). 12 Best PHP Frameworks To Build Amazing Website In 2016. Obtido 29 de Março de 2017, de <https://www.rswebsols.com/tutorials/programming/12-best-php-frameworks-2016>
- Bezerra, P., & Juliano, S. (2016). Desenvolvimento de aplicações cross-platform utilizando phonegap. *Revista Observatorio de la Economía Latinoamericana*.
- Bleumers, L., Van den Broeck, W., Lievens, B., & Pierson, J. (2012). Seeing the Bigger Picture: A User Perspective on 360 TV. *Proceedings of the 10th European Conference on Interactive TV and Video*, 115–124.

BoinxSoftwareLtd. (2016). *Boinx Crew watches Apple WWDC keynote (360° live stream with Theta S and mimoLive)*. Obtido de <https://www.youtube.com/watch?v=-gnNNuTtybl>

Cambridge. (2017). framework. Obtido 13 de Fevereiro de 2017, de <http://dictionary.cambridge.org/pt/dicionario/ingles/framework>

CodeIgniter. (2017). CodeIgniter. Obtido 3 de Abril de 2017, de <https://codeigniter.com/>

craftyazz. (2017). Update! Youtube and did first Live Stream test with audio now. Obtido 20 de Março de 2017, de <https://forums.360fly.com/t/update-youtube-and-did-first-live-stream-test-with-audio-now/2205>

Djulovic, I., & Duric, Z. (2015). Cross-platform web based real-time communication in web TV and video on demand systems. *2015 57th International Symposium ELMAR (ELMAR)*, 65–68. <https://doi.org/10.1109/ELMAR.2015.7334497>

Facebook. (2017a). Live 360 Getting Started Guide April 2017. Obtido 10 de Maio de 2017, de <https://fb360video.files.wordpress.com/2017/03/live-360-getting-started-guide.pdf>

Facebook. (2017b). Plataforma de apresentação do direto 360 a partir do Facebook. Obtido 10 de Maio de 2017, de <https://facebook360.fb.com/live360/>

Franco, E., Piedade, M., & Rêgo, R. (2014). Protótipo de um framework MVC para aplicações PHP de pequeno porte. *Anais do Encontro Regional de Computação e Sistemas de Informação*.

Gaemperle, L., Seyid, K., Popovic, V., & Leblebici, Y. (2014). An Immersive Telepresence System Using a Real-Time Omnidirectional Camera and a Virtual Reality Head-Mounted Display. *2014 IEEE International Symposium on Multimedia*, 175–178. <https://doi.org/10.1109/ISM.2014.62>

Hollister, S. (2015). YouTube's Ready To Blow Your Mind With 360-Degree Videos. Obtido 29 de Abril de 2017, de <http://gizmodo.com/youtubes-ready-to-blow-your-mind-with-360-degree-videos-1690989402>

- Hutt, R. (2017, Março 20). The world's most popular social networks, mapped. Obtido 15 de Abril de 2017, de <https://www.weforum.org/agenda/2017/03/most-popular-social-networks-mapped/>
- Jennings, C., Hardie, T., & Westerlund, M. (2013). Real-time communications for the web. *Communications Magazine, IEEE*, 51, 20–26.
<https://doi.org/10.1109/MCOM.2013.6495756>
- judiosan. (2016). How can I use my 360fly HD as a WebCam on my computer? Obtido 20 de Março de 2017, de <https://forums.360fly.com/t/how-can-i-use-my-360fly-hd-as-a-webcam-on-my-computer/1896>
- Kimata, H., Isogai, M., Noto, H., Inoue, M., Fukazawa, K., & Matsuura, N. (2011). Interactive panorama video distribution system. *2011 Technical Symposium at ITU Telecom World (ITU WT)*, 45–50.
- Laal, M., & Ghodsi, S. M. (2012). Benefits of collaborative learning. *Procedia - Social and Behavioral Sciences*, 31, 486–490.
<https://doi.org/http://dx.doi.org/10.1016/j.sbspro.2011.12.091>
- Levy, A. (2016, Novembro 27). The Most Popular Social Networks of 2016. Obtido 15 de Abril de 2017, de <https://www.fool.com/investing/2016/11/27/the-most-popular-social-networks-of-2016.aspx>
- Lock, J., & Johnson, C. (2015). Triangulating Assessment of Online Collaborative Learning. *Quarterly Review of Distance Education*, 16(4), 61–70.
- Loreto, S., & Romano, S. P. (2012). Real-Time Communications in the Web: Issues, Achievements, and Ongoing Standardization Efforts. *IEEE Internet Computing*, 16(5), 68–73. <https://doi.org/10.1109/MIC.2012.115>
- Lovejoy, B. (2017). Vimeo adds support for 360-degree videos, offline playback, 8K content and mor. Obtido 2 de Maio de 2017, de <https://9to5mac.com/2017/03/08/vimeo-adds-360-degree-videos-offline-playback-8k-content>

- Macmillan. (2017). framework - definition and synonyms. Obtido 13 de Fevereiro de 2017, de <http://www.macmillandictionary.com/dictionary/british/framework>
- Majumdar, A. (2015). Top 5 Benefits Of Investing In Collaborative Tools. Obtido 7 de Junho de 2017, de <https://elearningindustry.com/investing-in-collaborative-tools>
- Monus, A. (2017). 10 PHP Frameworks For Developers - Best of. Obtido 29 de Março de 2017, de <http://www.hongkiat.com/blog/best-php-frameworks/>
- Neng, L., & Chambel, T. (2010). Get Around 360° Hypervideo. *Proceedings of the 14th International Academic MindTrek Conference: Envisioning Future Media Environments*, 119–122. <https://doi.org/10.1145/1930488.1930512>
- Noupe. (2009). Discussing PHP Frameworks: What, When, Why and Which? Obtido 14 de Fevereiro de 2017, de <https://www.noupe.com/development/discussing-php-frameworks.html>
- Ochi, D., Kunita, Y., Fujii, K., Kojima, A., Iwaki, S., & Hirose, J. (2014). HMD Viewing Spherical Video Streaming System. *Proceedings of the 22Nd ACM International Conference on Multimedia*, 763–764. <https://doi.org/10.1145/2647868.2654870>
- Ochi, D., Niwa, K., Kameda, A., Kunita, Y., & Kojima, A. (2015). Dive into Remote Events: Omnidirectional Video Streaming with Acoustic Immersion. *Proceedings of the 23rd ACM International Conference on Multimedia*, 737–738. <https://doi.org/10.1145/2733373.2807963>
- Oxford. (2017). framework noun. Obtido 13 de Fevereiro de 2017, de <http://www.oxfordlearnersdictionaries.com/definition/english/framework>
- Perez, S. (2016). YouTube rolls out support for 360-degree live streams and spatial audi. Obtido 29 de Abril de 2017, de <https://techcrunch.com/2016/04/18/youtube-rolls-out-support-for-360-degree-live-streams-and-spatial-audio/>
- PHP. (2017). O que é o PHP? Obtido 27 de Março de 2017, de https://secure.php.net/manual/pt_BR/intro-what-is.php

- Pop, D.-P., & Altar, A. (2014). Designing an MVC Model for Rapid Web Application Development. *Procedia Engineering*, 69, 1172–1179.
<https://doi.org/http://dx.doi.org/10.1016/j.proeng.2014.03.106>
- potro777. (2016). Use desktop as a viewfinder. Obtido 20 de Março de 2017, de <https://forums.360fly.com/t/use-desktop-as-a-viewfinder/1418/7>
- Proença, L. (2017, Março 18). Vimeo investe na realidade virtual. Obtido 2 de Maio de 2017, de <http://expresso.sapo.pt/sociedade/2017-03-18-Vimeo-investe-na-realidade-virtual>
- Quax, P., Liesenborgs, J., Issaris, P., Lamotte, W., & Claes, J. (2013). A Practical and Scalable Method for Streaming Omni-directional Video to Web Users. *Proceedings of the 2013 ACM International Workshop on Immersive Media Experiences*, 57–60.
<https://doi.org/10.1145/2512142.2512149>
- Rhinow, F., Veloso, P. P., Puyelo, C., Barret, S., & Nuallain, E. (2014). P2P live video streaming in WebRTC. *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, 1–6. <https://doi.org/10.1109/WCCAIS.2014.6916588>
- Salas, F. Ó., & Kalva, H. (2013). Architecting social TV. *Proceedings of the SPIE*, 8856.
<https://doi.org/10.1117/12.2026551>
- Samuel, A. (2015). Collaborating Online Is Sometimes Better than Face-to-Face. Obtido 7 de Junho de 2017, de <https://hbr.org/2015/04/collaborating-online-is-sometimes-better-than-face-to-face>
- Shin, D.-H. (2013). Defining sociability and social presence in Social TV. *Computers in Human Behavior*, 29(3), 939–947.
<https://doi.org/https://doi.org/10.1016/j.chb.2012.07.006>
- Statz, P. (2010). Get Started with Web Frameworks. Obtido 14 de Fevereiro de 2017, de https://www.wired.com/2010/02/get_started_with_web_frameworks/
- Swan, K., Shen, J., & hiltz, S. R. (2006). Assessment and Collaboration in Online Learning. Obtido de <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.475.408>
- Symfony. (2017). Symfony. Obtido 2 de Abril de 2017, de <http://symfony.com/>

Techterms. (2013). Framework. Obtido 13 de Fevereiro de 2017, de <https://techterms.com/definition/framework>

UEFA. (2017a). *UEFA Champions League Draw Re-Run in 360!* Obtido de <https://www.youtube.com/watch?v=-gnNNuTtybl>

UEFA. (2017b). *UEFA Europa League Draw Re-Run in 360!* Obtido de <https://www.youtube.com/watch?v=D9OmSzD5WvA>

Velasco, H. (2017). Vimeo debuts 360 offerings for 'immersive storytelling' opportunity. Obtido 2 de Maio de 2017, de <http://www.thedrum.com/news/2017/03/13/vimeo-debuts-360-offerings-immersive-storytelling-opportunity>

Vimeo. (2017a). Carregando vídeos 360. Obtido 2 de Maio de 2017, de <https://help.vimeo.com/hc/pt/articles/115001877167-Carregando-videos-360>

Vimeo. (2017b). Plataforma Vimeo. Obtido 2 de Maio de 2017, de <https://vimeo.com/about>

Wang, G. (2011). Application of lightweight MVC-like structure in PHP. *2011 International Conference on Business Management and Electronic Information*, 2, 74–77. <https://doi.org/10.1109/ICBMEI.2011.5917846>

Westrate, C. (2016). *Outdoor Theta S 360 Live stream test*. Obtido de <https://www.youtube.com/watch?v=XniRgOVSRW4&t=13902s>

World Web Technology. (2016). List Of Top 10 Rated PHP Frameworks Of 2016. Obtido 29 de Março de 2017, de <http://www.worldwebtechnology.com/list-top-10-rated-php-frameworks-2016/>

&yet. (2017). simpleWebRTC. Obtido 15 de Abril de 2017, de <https://simplewebrtc.com/>

Youtube. (2017a). Acerca do Youtube. Obtido 29 de Abril de 2017, de <https://www.youtube.com/intl/pt-PT/yt/about/>

Youtube. (2017b). Carregar vídeos de 360 graus. Obtido 29 de Abril de 2017, de <https://support.google.com/youtube/answer/6178631?hl=pt>

Youtube. (2017c). Definições do codificador para vídeos em direto de 360 graus. Obtido 29 de Abril de 2017, de <https://support.google.com/youtube/answer/6396222?hl=pt>

Youtube. (2017d). Imprensa Youtube. Obtido 29 de Abril de 2017, de
<https://www.youtube.com/intl/pt-PT/yt/about/press/>