



UNIVERSIDADE D
COIMBRA

Miguel Jorge Rasteiro Letra

ARCRITICAL
PLATFORM FOR 3D AND AUGMENTED REALITY PRODUCT
VISUALIZATION

VOLUME 1

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Engenharia de Software, orientada pelo Professor Doutor João Nuno Gonçalves Costa Cavaleiro Correia e apresentada à Faculdade de Ciências e Tecnologia / Departamento de Engenharia Informática.

Setembro de 2021

Faculdade de Ciências e Tecnologia
Departamento de Engenharia Informática

ARCRITICAL

Platform for 3D and Augmented Reality Product Visualization

Miguel Jorge Rasteiro Letra

Dissertação no âmbito do Mestrado em Engenharia Informática, especialização em Engenharia de Software, orientada pelo Professor Doutor João Nuno Gonçalves Costa Cavaleiro Correia e apresentada à Faculdade de Ciências e Tecnologia / Departamento de Engenharia Informática.

Setembro de 2021



UNIVERSIDADE D
COIMBRA

Resumo

De forma a promover a interação com os seus funcionários, a Critical Software adota várias iniciativas ao longo do ano nas quais, para além das diversas atividades organizadas é também habitual oferecerem pequenos packs ou kits com algumas surpresas no seu interior. O objetivo do projeto é tirar partido destas iniciativas, melhorando a experiência através da realidade aumentada. O projeto ARCritical visa o desenvolvimento de um sistema que permita aos funcionários da Critical Software, através de uma aplicação móvel, visualizar conteúdo de realidade aumentada preparado de acordo com as iniciativas. Para isso, é necessária a implementação de um sistema que possibilite, não só a visualização do conteúdo, mas também que permita, às equipas responsáveis pelas iniciativas, definir e customizar o que será mostrado. Um aspeto considerado fundamental para o sistema é o facto de possibilitar que este conteúdo seja preparável e atualizável sem a necessidade de uma nova compilação da aplicação móvel, e, ao mesmo, não ser necessário possuir qualquer conhecimento de realidade aumentada ou do funcionamento do sistema desenvolvido.

O projeto iniciou-se com o estudo do estado da arte de realidade aumentada de forma a perceber, como esta tem evoluído, onde é utilizada, investigar possíveis limitações, conceitos e as ferramentas de desenvolvimento existentes.

Posteriormente foram definidos os requisitos funcionais e atributos de qualidade do sistema, assim como alguns casos de uso que descrevem o funcionamento do mesmo. De destacar que, todos estes foram sofrendo alterações ao longo do desenvolvimento, com base no feedback recebido. Estas alterações foram possíveis graças à metodologia de desenvolvimento adotada.

A partir destes requisitos e do conhecimento obtido, através do estudo do estado da arte, foi proposta uma solução arquitetural, que não só define a estrutura do sistema, mas também especifica as tecnologias que foram utilizadas em cada uma das suas componentes.

Com base no planeamento do projeto e com toda a informação recolhida até aqui, foi feita uma análise de riscos. Nesta análise para além de identificar possíveis riscos e o respetivo impacto sobre o produto final, foi especificado como deve ser feita a sua gestão e monitorização, de forma a reduzir este impacto ou até mesmo mitigá-lo.

Antes de iniciar a implementação do projeto foi desenvolvido algum trabalho preliminar. Foi implementado um pequeno protótipo de uma aplicação móvel de realidade aumentada, de forma a introduzir as tecnologias, ferramentas, e metodologias de desenvolvimento adotadas. Foi também feita uma especificação inicial da API que faz parte do sistema.

Após este trabalho preliminar, iniciou-se a implementação do sistema de acordo com a arquitetura planeada. Nesta arquitetura o sistema é composto por 3 grandes componentes: Aplicação móvel, que permite aos utilizadores visualizar a experiência de realidade aumentada preparada, Backend, responsável por garantir a criação, customização, e modificabilidade destas experiências, e, por último, Interface de Gestão do Conteúdo que permite a qualquer pessoa, desde que tenha autorização para isso, independentemente do seu nível de conhecimento do sistema, interagir com o Backend para criar e editar as experiências.

Por último, para além dos vários testes unitários e de integração feitos para cada componente ao longo do desenvolvimento, após concluído todo o sistema, foram feitos testes *end-to-end* de forma a verificar e validar o comportamento do sistema como um todo.

Desta forma foram cumpridos os objetivos propostos, existindo um sistema funcional capaz de proporcionar as funcionalidades desejadas de criação, edição, e visualização das experiências de realidade aumentada, ultrapassando os desafios encontrados.

Palavras-Chave

Realidade Aumentada, AR Foundation, Aplicação móvel, Unity, .Net, Docker, Azure

Abstract

To improve engagement with its employees, Critical Software adopts several initiatives throughout the year in which, besides organizing several group activities, it is customary to offer small packs or kits with some surprises inside. The goal of the project is to take advantage of these initiatives, improving the experience through augmented reality. The ARCritical project aims to develop a system that allows Critical Software's employees to view, through a mobile application, augmented reality content prepared in accordance with these initiatives. For this, it is necessary to implement a system that not only enables the visualization of the content, but also allows the teams responsible for the initiatives to define and customize what will be shown. A fundamental aspect of the system is the fact that it allows this content to be preparable and updatable without the need for a new build of the mobile application and, at the same time, not being mandatory to have any knowledge about augmented reality or the developed system.

The project began with a state-of-the-art study about augmented reality to understand how it has evolved, where it is used, to investigate possible limitations, important concepts, and the existing development tools.

Afterwards, both system requirements and quality attributes of the system were defined, as well as some use cases that describe its operation. It is noteworthy that all of these were changed throughout the development, based on the feedback received. These changes were possible thanks to the development methodology adopted.

From these requirements and the knowledge acquired through the state-of-the-art study, an architectural solution was proposed, which not only defines the structure of the system but also specifies the technologies that were used in each of its components.

Based on the project scheduling and all the information collected so far, a risk analysis was made. In this analysis, besides identifying possible risks and their impact on the final product, it was specified how they should be managed and monitored, to reduce this impact or even mitigate it.

Before starting the implementation of the project, some preliminary work was developed. Starting with the implementation of a small prototype of an augmented reality mobile application, to introduce the technologies, tools and development methodologies adopted. It was also made an initial draft of the API specification.

With the completion of the preliminary work, began the implementation of the system according to the planned architecture. In this architecture the system consists of 3 major components: Mobile application, which allows users to view the prepared augmented reality experience, Backend, responsible for ensuring the creation, customization, and modifiability of these experiences, and, finally, Content Management Interface that allows anyone, provided they have the necessary authorization, regardless of their level of knowledge of the system, to interact with the Backend to create and edit the experiences.

Finally, in addition to the various unit and integration tests made for each component throughout the development, after the whole system was finished, *end-to-end* tests were conducted to verify and validate the behavior of the entire system.

In the end, the proposed objectives were met, with a full functioning system being available and capable of providing the desired functionalities for creating, editing, and visualizing augmented reality experiences, overcoming the encountered challenges.

Keywords

Augmented Reality, AR Foundation, Mobile application, Unity, .Net, Docker, Azure

Agradecimentos

Com o terminar do estágio, não posso deixar de agradecer às pessoas que, direta ou indiretamente, me ajudaram nesta última fase do meu Mestrado, permitindo-me desenvolver um produto do qual me sinto orgulhoso, sabendo que abre inúmeras oportunidades futuras para o uso de Realidade Aumentada.

Aos orientadores da Critical Software, Álvaro Menezes e Rui Gonçalves, agradeço não só por me terem dado a oportunidade de participar neste estágio, mas também pelo constante apoio e disponibilidade mostrada ao longo de todo o caminho.

Agradeço ao Brian Rodrigues, que não só me ajudou em aspetos técnicos e me ensinou bastante, mas também rapidamente se tornou num amigo que esteve sempre presente e pronto para ajudar.

Ao Professor Doutor João Nuno Gonçalves Costa Cavaleiro Correia, orientador do estágio, que provou estar sempre disponível para ajudar ou tirar dúvidas e ao mesmo tempo mostrou interesse em acompanhar e melhorar o projeto partilhando as suas ideias.

À minha família, por estarem sempre presentes e por todo o apoio e carinho dado, não só ao longo desta minha jornada académica, mas toda a minha vida.

Aos meus amigos, por estarem sempre presentes e partilharem comigo momentos especiais indispensáveis.

Índice

Capítulo 1	Introdução	1
1.1	Estrutura do documento	2
Capítulo 2	Realidade Aumentada: Estado da arte	4
2.1	História	4
2.2	Utilização	9
2.3	Limites, oportunidades e futuro	16
2.4	Desenvolvimento de realidade aumentada	18
	Ferramentas de Desenvolvimento	19
	ARCore	21
	ARKit	23
	AR Foundation	26
	Vuforia	27
	Wikitude	30
	Opiniões externas e Conclusões	33
Capítulo 3	Levantamento de requisitos	34
3.1	Nomenclatura e Convenções	34
3.2	Contextualização	34
3.3	Análise de requisitos funcionais	35
	Requisitos do Backend e da Interface de Gestão do Conteúdo	35
	Requisitos da Aplicação móvel:	39
	Casos de uso	40
3.4	Requisitos não-funcionais / Atributos de Qualidade	48
Capítulo 4	Planeamento	50
4.1	Metodologia de desenvolvimento	50
4.2	Planeamento	52
Capítulo 5	Análise de Riscos	55
Capítulo 6	Trabalho Preliminar	57
6.1	Protótipo	57
6.2	Especificação da API	59
Capítulo 7	Arquitetura Implementada	63
7.1	Aplicação móvel	64
	Ferramenta de desenvolvimento	64
	Fases de funcionamento	66
7.2	Backend	68
	Api Privada	69
	Api Pública	72
	Container Unity	72
	Container FFmpeg	73
	Base de dados	74
7.3	Interface de Gestão do Conteúdo	75
Capítulo 8	Funcionamento	76
Capítulo 9	Testes	83

9.1 Triggers	83
9.2 Ações	84
Capítulo 10 Conclusão e Oportunidades Futuras	87
Referências	89

Acrónimos

Acrónimo	Descrição
API	<i>Application Programming Interface</i>
HMD	<i>Head Mounted Display</i>
MARS	<i>Mobile Augmented Reality System</i>
TAR	<i>Tactical Augmented Reality</i>
RA / AR	<i>Realidade Aumentada / Augmented Reality</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
SDK	<i>Software Development Kit</i>
LiDAR	<i>Light Detection And Ranging</i>
VISLAM	<i>Visual-Inertial Simultaneous Localization and Mapping</i>
VIO	<i>Visual-Inertial Odometry</i>
RF	Requisito Funcional
CPU	<i>Central Processing Unit</i>
GPU	<i>Graphics Processing Unit</i>
VPN	<i>Virtual Private Network</i>
CI / CD	<i>Continuous Integration / Continuous Delivery</i>
DTO	<i>Data Transfer Object</i>

Lista de Figuras

Figura 1 - Sword of Damocles. Fonte: [1].....	4
Figura 2 - Exemplo de interação através do Videoplace: escrever utilizando o teclado virtual. Fonte: [5].....	5
Figura 3 - Ilustração do funcionamento do Videoplace. Fonte: [5]	5
Figura 4 - Teste do sistema Virtual Fixtures. Fonte: [1].....	6
Figura 5 - MARS, Sistema de informação do campus. Fonte: [8]	6
Figura 6 - Uso da linha amarela numa transmissão desportiva. Fonte: [9].....	6
Figura 7 - Jogo ARQuake, Fonte: [10].....	7
Figura 8 - Jogo AR Tennis desenvolvido pela Nokia. Fonte: [11].....	7
Figura 9 - Campanha de Marketing da BMW. Fonte: [12]	7
Figura 10 - Google Glasses lançados em 2015. Fonte:[13].....	8
Figura 11 - Microsoft HoloLens (1ª gen), lançados em 2015. Fonte: [14]	8
Figura 12 - Microsoft HoloLens 2 lançados em 2019. Fonte: [15]	8
Figura 13 - Uso dos Microsoft HoloLens na preparação de uma operação. Fonte: [18]	9
Figura 14 - AccuVein. Fonte: [20].....	9
Figura 15 - Aplicação Anatomy4D utilizada no estudo mencionado. Fonte: [22].	10
Figura 16 - Aplicação criada pela Wikitude que permite combinar elementos químicos. Fonte: [24].....	10
Figura 17 - Aplicação IKEA Place. Fonte: [26]	11
Figura 18 - Aplicação Nike Fit. Fonte: [27]	11
Figura 19 - Aplicação Warby Parker. Fonte: [28].....	11
Figura 20 - Pokemon Go Fonte: [29]	12
Figura 21- Minecraft Earth, outro exemplo de aplicação. Fonte: [30]	12
Figura 22 - Interface de suporte, com informações sobre o elevador (Thyssenkrupp). Fonte:[31].....	12
Figura 23 - Videochamada remota com especialista (Thyssenkrupp). Fonte:[31]	12
Figura 24 - Uso dos Google Glass pela Boeing. Fonte: [32]	13
Figura 25 - Visualização das falhas através dos HoloLens. Fonte: [34]	13
Figura 26 - Visualização das falhas através do tablet. Fonte:[33]	13
Figura 27 - Funcionalidade de realidade aumentada do Google Maps para utilizadores pedestres. Fonte: [35].....	14
Figura 28 - Exemplos de utilização de uma interface para navegação no para-brisas. Fonte, 1ª imagem:[36],2ª imagem: [35]	14
Figura 29 - GuideBot, aplicação de navegação para interior de edifícios. Fonte: [38].....	14

Figura 30 - História da Jim Beam contada através de realidade aumentada. Fonte: [40].....	15
Figura 31 - Uso de Realidade aumentada no mural do Fonte:[41].....	15
Figura 32 - Uso de realidade aumentada numa revista turística. Fonte:[42].....	16
Figura 33 - Tarefas que eram desempenhadas nos dispositivos utilizados atualmente passam a ser feitas usando AR ou VR. Fonte: [45]	17
Figura 34 - Ilustração de uns possíveis futuros óculos de realidade aumentada. Fonte: [45]	17
Figura 35 - Possível separação do processamento, graças aos benefícios da rede 5G. Fonte: [45].....	18
Figura 36 - Popularidade dos SDKs com base no número de repositórios no Github (29/10/2020).....	20
Figura 37 - Imagem da esquerda não tira partido das funcionalidades de profundidade obtendo resultados menos realísticos do que da direita. Fonte: [50]	22
Figura 38 - Exemplo de aplicação de um filtro sobre a imagem detetada. Fonte:[53].....	24
Figura 39 - Scan do objeto para posterior identificação. Fonte:[54].....	24
Figura 40 - Esquerda reflete a falta de oclusão. Na direita ocorre a oclusão dos livros virtuais quando a pessoa está mais próxima da câmara. Fonte:[55].....	25
Figura 41 - Lista das funcionalidades suportadas pelo AR Foundation. Fonte:[57].....	26
Figura 42 - Exemplo de aplicação da funcionalidade Multi-Targets. Fonte:[59].....	28
Figura 43 - Exemplo de aplicação da funcionalidade Cylinder Targets. Fonte:[60]	28
Figura 44 - Exemplos de VuMarks. Fonte:[61].....	29
Figura 45 - Prioridade de escolha de acordo com a capacidade do dispositivo. Fonte:[62]..	29
Figura 46 - Diferentes licenças e respetivas funcionalidades. Fonte:[63]	30
Figura 47 - Arquitetura do Wikitude. Fonte: [65].....	31
Figura 48 - Dispositivos suportados através da API do Wikitude para Unity. Fonte: [66]...	32
Figura 49 - Preços das opções disponibilizadas do Wikitude. Fonte:[67]	32
Figura 50: Diagrama de Gantt do trabalho realizado e planeado (1º Semestre).	53
Figura 51 - Diagrama de Gantt do trabalho realizado (2º Semestre).....	54
Figura 52 - - Swagger Preview das operações definidas para os Triggers.....	60
Figura 53 - Swagger Preview das operações definidas para as Ações.....	60
Figura 54 - Swagger Preview das operações definidas para as Campanhas.	60
Figura 55 - Swagger Preview das operações relacionadas com o download de recursos.	61
Figura 56 - Exemplo de especificação de um pedido à API presente no Backend.....	62
Figura 57 - Arquitetura de alto nível do sistema.....	63
Figura 58 - Estatística de distribuição das versões de Android. Fonte: Android Studio.	65
Figura 59 -Exemplo de sistema de partículas (Branças) em torno do objeto selecionado.	67

Figura 60 - Arquitetura e Tecnologias usadas na componente do sistema: Backend.....	68
Figura 61 – Camadas utilizadas pela Web API seguindo a Onion Architecture.	69
Figura 62 – Diferença de acoplamento das camadas entre a N-Layer Architecture (esquerda) e Onion Architecture (direita).	70
Figura 63 - Exemplo de comando de inicialização de uma instância do Unity Editor, com plataforma alvo Android.....	72
Figura 64 – Tabela com os formatos suportados pelo Unity Editor para cada plataforma. Fonte: [73].....	73
Figura 65 - Separador dos Triggers da Interface de Gestão do Conteúdo.	75
Figura 66 - Adicionar um novo Trigger (<i>SummerCampTrigger</i>) através da Interface de Gestão do Conteúdo.....	76
Figura 67 - Separador que contém todos os Triggers já adicionados. O Backend está ainda a processar o novo Trigger para que possa ser utilizado pela Aplicação.	77
Figura 68 - Adicionar uma nova Ação (<i>WelcomeVideoCritical</i>) através da Interface de Gestão do Conteúdo.....	77
Figura 69 - Separador que contém todas as Ações já adicionadas. O Backend está ainda a processar a nova Ação para que possa ser utilizada pela Aplicação.	78
Figura 70 - Criação de uma nova Campanha, através da Interface de Gestão do Conteúdo. Nome dado à campanha: <i>SummerCampCampaign</i> , Data de início: 27/07/2021, Data de fim: 30/07/2021.....	78
Figura 71 - Associar Trigger (<i>SummerCampTrigger</i>) à Campanha criada (<i>SummerCampCampaign</i>).	79
Figura 72 - Associar uma Ação (<i>WelcomeVideoCritical</i>) a um determinado Trigger (<i>SummerCampTrigger</i>) de uma Campanha (<i>SummerCampCampaign</i>).....	79
Figura 73 - Triggers e respetivas Ações de cada uma das Campanhas já criadas.	80
Figura 74 – Customização de cada Ação de acordo com as propriedades disponíveis.....	80
Figura 75 - Utilizador pode optar por fazer o Download agora ou mais tarde.	81
Figura 76 - Progresso do Download de novos conteúdos.	81
Figura 77 – Aplicação indica quantas imagens são reconhecíveis, ou seja, quantos Triggers existem.....	81
Figura 78 - Ao ser reconhecido o Trigger a Aplicação mostra as respetivas Ações sobrepostas a este.	82
Figura 79 - Vídeo de demonstração do funcionamento da Aplicação.....	82

Lista de Tabelas

Tabela 1 - Tabela de comparação das ferramentas de desenvolvimento de realidade aumentada.	21
Tabela 2 - Tabela de requisitos funcionais do Backend e da Interface de Gestão do Conteúdo, e respectivas prioridades.....	36
Tabela 3 - Parâmetros de customização aplicáveis para cada tipo de Ação.	38
Tabela 4 - Tabela de requisitos funcionais, da aplicação móvel, e respectivas prioridades....	39
Tabela 5 - Tabela modelo da definição de casos de uso utilizada.	40
Tabela 6 - Caso de uso (app): Fase de inicialização (Atualizar Campanhas).....	40
Tabela 7 - Caso de uso (app): Verificação das permissões de uso da câmara.....	41
Tabela 8 - Caso de uso (app): Despoletar Ação	42
Tabela 9 - Caso de uso (app): Tirar Fotografia.....	42
Tabela 10 - Caso de uso (app): Gravar vídeo	43
Tabela 11 – Caso de Uso (Interface): Criar uma Campanha.....	44
Tabela 12 – Caso de Uso (Interface): Adicionar novo Trigger.....	45
Tabela 13 – Caso de Uso (Interface): Adicionar nova Ação.....	45
Tabela 14 - Caso de uso (Interface): Associar Trigger a Campanha	46
Tabela 15 - Caso de uso (Interface): Associar Ação a Trigger de uma Campanha	47
Tabela 16 - Análise de Riscos.....	55
Tabela 17 - Versões Android e iOS suportadas pela AR Foundation, Vuforia e Wikitude...65	
Tabela 18- Resultados dos testes de Triggers.....	84
Tabela 19 - Uso de memória com múltiplos Triggers em simultâneo.	84
Tabela 20 - Resultados dos testes de Ações (Vídeos).....	85
Tabela 21 - Resultados dos testes de Ações (Áudio).....	85
Tabela 22 - Resultados dos testes de Ações (Modelos 3D).....	86
Tabela 23 - Resultados dos testes de Ações (Imagens).....	86

Capítulo 1

Introdução

A realidade aumentada é uma tecnologia que, quando utilizada corretamente, permite alterar a maneira como resolvemos problemas e como visualizamos e interagimos com o nosso mundo. Saúde, entretenimento, indústria e marketing são algumas das muitas áreas nas quais esta tecnologia tem tido um grande impacto, contribuindo para um aumento da produtividade, eficiência e segurança em várias atividades. Além disso, possibilita a criação de experiências memoráveis que geram respostas emocionais, melhorando a experiência dos utilizadores. Através da aplicação desta tecnologia, é possível integrar elementos virtuais na nossa realidade, criando assim inúmeras possibilidades de expandir o mundo real através do virtual.

Com o intuito de promover a interação com os seus funcionários, a Critical Software adota várias iniciativas ao longo do ano, nas quais, para além das diversas atividades organizadas, é habitual oferecerem pequenos packs ou kits com várias surpresas no seu interior. Funcionários que estão a entrar pela primeira vez na empresa são recebidos através da iniciativa *Onboarding*, constituída por várias atividades e onde, para além de receberem um kit com as várias ferramentas necessárias para trabalhar, são presenteados com algumas surpresas. O mesmo acontece para atuais funcionários da empresa, que em determinadas épocas especiais, como o Natal ou o aniversário da Critical, recebem pequenos packs que simbolizam a ocasião.

O objetivo do estágio é tirar partido destas iniciativas ou de outras que possam surgir, melhorando as experiências através de realidade aumentada. Ao apontarem a câmara dos seus dispositivos móveis para determinados conteúdos, os funcionários serão capazes de visualizar a experiência de realidade aumentada preparada pela equipa responsável pela iniciativa. Para isso, decidimos construir um sistema que possibilite: definir os elementos que serão utilizados como estímulo para despoletar cada experiência, definir e customizar o conteúdo que será mostrado, tendo em conta que, as equipas responsáveis pelas iniciativas poderão não ter conhecimentos técnicos de realidade aumentada ou de qualquer outra tecnologia inerente, e, por último, através de uma aplicação móvel permita aos funcionários visualizar a experiência preparada para cada estímulo reconhecido. Outra característica acordada como fundamental para a nossa solução, é a necessidade de garantir que estas experiências possam ser criadas ou atualizadas sem ser necessária uma nova compilação e respetiva instalação da aplicação. Para isso definimos um sistema composto por 3 componentes:

- **Aplicação móvel:** responsável por fornecer as experiências de realidade aumentada definidas. Desenvolvida para ambas as plataformas, Android e iOS, é capaz de, ao reconhecer estímulos, despoletar experiências de realidade aumentada.

Um dos maiores desafios deste projeto foi o facto de ser fundamental que esta Aplicação fosse capaz de mostrar novas experiências sem a necessidade de fazer uma nova compilação da mesma. Surgindo assim a componente Backend, para solucionar este desafio.

- **Backend:** esta é o coração de todo o sistema, sendo responsável por garantir a criação, customização e modificabilidade das experiências de realidade aumentada. É nesta

componente onde acontece o processamento de novos conteúdos, para que possam ser utilizados pela Aplicação móvel.

Outro desafio foi possibilitar a criação destas experiências de realidade aumentada a qualquer pessoa, independentemente do seu nível de conhecimento. Surgindo assim a componente Interface de Gestão do Conteúdo.

- **Interface de Gestão do Conteúdo:** interface, acedida através de um browser, que permite criar de forma simples e intuitiva as experiências de realidade aumentada. É através desta que se especificam os estímulos que estão a ser utilizados, e para cada um desses estímulos se define a experiência de realidade aumentada que deve ser mostrada.

Para além dos desafios mencionados, existiram certos obstáculos que tiveram impacto na quantidade de funcionalidades implementadas, alguns destes previstos graças ao estudo do estado da arte e ao planeamento efetuado na fase inicial do estágio. A grande variedade de hardware e software existente nos dispositivos móveis associado ao facto de suportarem diferentes funcionalidades de realidade aumentada torna difícil o desenvolvimento de aplicações que tirem partido desta tecnologia. Para além disso, tratando-se de uma aplicação de realidade aumentada, a realização de testes torna-se complexa graças à necessidade de utilização de imagens recolhidas pela câmara do dispositivo. Uma vez que a utilização de emuladores é bastante limitada, foi necessário recorrer a dispositivos reais para a realização de testes. No total foram disponibilizados 3 dispositivos diferentes, para auxiliar o desenvolvimento do trabalho, estando assim limitado à implementação de funcionalidades de realidade aumentada suportadas por estes.

1.1 Estrutura do documento

O restante documento está estruturado da seguinte forma:

- **Capítulo 2 Estudo do estado da arte:** É feito um resumo de toda a informação recolhida sobre realidade aumentada, desde a sua história, onde é utilizada, possíveis limitações e avanços futuros, conceitos importantes, e por último uma análise das ferramentas de desenvolvimento existentes.
- **Capítulo 3 Levantamento de requisitos:** São definidas nomenclaturas que serão utilizadas ao longo de todo o projeto, é especificada a técnica de priorização de requisitos utilizada, é feita uma contextualização mais detalhada do objetivo do projeto, e por últimos são especificados os requisitos funcionais, atributos de qualidade e definidos alguns casos de uso.
- **Capítulo 4 Planeamento:** É especificada a metodologia de desenvolvimento adotada e é feita uma comparação entre o que estava planeado no início do estágio face ao que realmente aconteceu.
- **Capítulo 5 Análise de Riscos:** São especificados os vários riscos identificados para este projeto, assim como as técnicas utilizadas de forma fazer a sua monitorização, diminuir a sua probabilidade e severidade.

- **Capítulo 6 Trabalho Preliminar:** É detalhado o trabalho desenvolvido antes de iniciar o desenvolvimento do sistema, mais especificamente, o protótipo de uma aplicação móvel de realidade aumentada, e uma especificação inicial da API.
- **Capítulo 7 Arquitetura Implementada:** É explorada a solução arquitetural adotada que, não só define a estrutura do sistema, mas também especifica as tecnologias que foram utilizadas em cada uma das suas componentes.
- **Capítulo 8 Funcionamento:** É feita uma demonstração do funcionamento do sistema como um todo, através de exemplos.
- **Capítulo 9 Testes:** são especificados alguns dos testes realizados ao sistema como um todo.
- **Capítulo 10 Conclusão e Oportunidades Futuras:** São apresentadas as conclusões do trabalho realizado e são exploradas possíveis oportunidades futuras.

Capítulo 2

Realidade Aumentada: Estado da arte

Realidade aumentada é uma tecnologia que permite expandir (aumentar) a nossa realidade através da integração de elementos virtuais no mundo real. Este conceito é muitas vezes confundido com o de realidade virtual, apesar de retratarem situações distintas. Enquanto que a realidade virtual consiste num mundo ou ambiente virtual gerado pelo computador, completamente isolado do mundo real, no qual o utilizador é inserido, a realidade aumentada é uma maneira de expandir o mundo real trazendo componentes virtuais para este. No entanto a definição de realidade aumentada tem vindo a ser adaptada nos últimos anos para incluir um híbrido, realidade mista (*mixed reality*), na qual é possível a interação dos elementos virtuais com o mundo real. Ao longo deste documento, quando o termo realidade aumentada é utilizado, refere-se à integração de elementos digitais, possíveis ou não de interagir, a qualquer ambiente real.

A tecnologia utilizada na realidade aumentada evoluiu bastante ao longo dos anos. A próxima secção irá abordar um pouco desta evolução, especialmente a evolução do hardware utilizado.

2.1 História

Foi nos anos 60 que surgiu o primeiro exemplo de combinação de elementos virtuais com o mundo real. Apesar de na altura ainda não existir o conceito de realidade aumentada [1], foi em 1968 que Ivan Sutherland, um informático e professor de Harvard, criou um dispositivo para colocar na cabeça chamado “The Sword of Damocles”. Este dava a possibilidade de observar imagens geradas pelo computador de várias perspetivas [2], [3]. Este dispositivo era parcialmente transparente, de modo que, os utilizadores não ficavam completamente isolados dos seus arredores, sendo por isso considerado como um precursor da realidade aumentada [4].

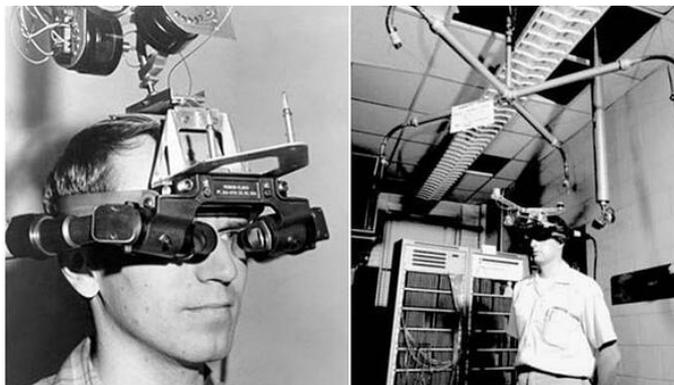


Figura 1 - Sword of Damocles. Fonte: [1]

Com a possibilidade de visualizar elementos virtuais no mundo real o próximo passo seria arranjar novas formas de interagir com estes elementos. Foi em 1975 que Myron Krueger criou o 'Videoplace', um laboratório de "realidade artificial" (termo usado na altura), no qual o utilizador consegue interagir com objetos virtuais através dos movimentos ou ações do participante [5], [6].



Figura 2 - Exemplo de interação através do Videoplace: escrever utilizando o teclado virtual. Fonte: [5]

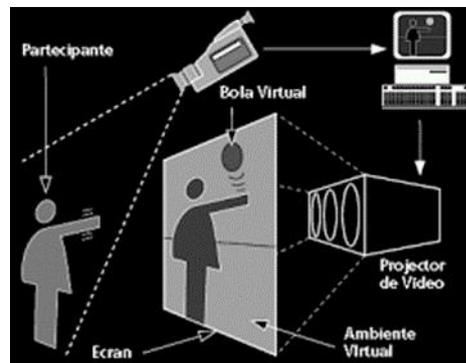


Figura 3 - Ilustração do funcionamento do Videoplace. Fonte: [5]

Apenas em 1990 foi criado o termo de realidade aumentada atribuído por Thomas P. Caudell e David Mizell, antigos investigadores da Boeing [1]. Ambos trabalharam em simplificar o processo de transmitir instruções de montagem dos fios aos funcionários da empresa, sendo a solução referida a sobreposição do material no mundo real (realidade aumentada), através de um '*Head-mounted display*' (HMD) [4].

Foi então a partir dos anos 90 que a realidade aumentada começou a ser aplicada em várias indústrias e a alcançar o público comum. Em 1992 um investigador da força aérea dos Estados Unidos, Louis Rosenburg, criou 'Virtual Fixtures', um sistema de realidade aumentada que permitia auxiliar, guiar o controlo remoto de maquinaria no desempenho de determinadas tarefas, sendo considerado um dos primeiros sistemas de realidade aumentada funcionais [1], [4], [7].



Figura 4 - Teste do sistema Virtual Fixtures. Fonte: [1]

Em 1997 foi desenvolvido o primeiro sistema de realidade aumentada móvel (MARS – *Mobile Augmented Reality System*), *Touring Machine*, que combinava um HMD ('Head Mounted Display'), um tablet e uma mochila com um computador, GPS e ligação à internet. Este fornecia informação sobre o campus da Universidade de Columbia [4], [8].



Figura 5 - MARS, Sistema de informação do campus. Fonte: [8]

Em 1998 o uso realidade aumentada foi introduzido numa grande variedade de sobreposições gráficas usadas em transmissões desportivas. Um dos exemplos mais conhecido é o uso da linha amarela no Rugby de forma a tornar mais fácil para os telespectadores acompanharem o jogo [4].



Figura 6 - Uso da linha amarela numa transmissão desportiva. Fonte: [9]

Em 1999 surgiu o primeiro kit de desenvolvimento de software de realidade aumentada ARToolkit, tornado open-source em 2001 [4].

No ano 2000 surgiu o primeiro jogo chamado ARQuake. Para além do display na cabeça, os utilizadores tinham de usar uma mochila que continha um computador, giroscópios e sistema GPS [10].



Figura 7 - Jogo ARQuake, Fonte: [10]

Graças a evolução no hardware dos smartphones, foi também a partir deste ano que começaram a aparecer as primeiras aplicações de realidade aumentada nestes dispositivos.

Em 2005 a Nokia desenvolveu o jogo de realidade aumentada, para 2 jogadores, AR Tennis [11].



Figura 8 - Jogo AR Tennis desenvolvido pela Nokia. Fonte: [11]

Em 2008 a BMW utilizou pela primeira vez realidade aumentada para marketing através do browser e da câmara do computador [12].



Figura 9 - Campanha de Marketing da BMW. Fonte: [12]

A partir de 2011 começaram a surgir os *smartglasses* e *headsets* como o Google Glass (lançados em 2013) e os HoloLens 1ª geração da Microsoft (lançados em 2015). Avanços recentes na miniaturização de dispositivos eletrônicos tornaram possível a integração de câmaras, sensores e ecrãs em pequenos produtos inteligentes, como os *smartglasses* e smartphones, aumentando a sua portabilidade bem como as aplicações possíveis da realidade aumentada [3].



Figura 10 - Google Glasses lançados em 2015. Fonte:[13]



Figura 11 - Microsoft HoloLens (1ª gen), lançados em 2015. Fonte: [14]



Figura 12 - Microsoft HoloLens 2 lançados em 2019. Fonte: [15]

Em 2017, verificou-se o interesse de grandes empresas em realidade aumentada, com a Google e a Apple a lançar os seus kits de desenvolvimento de software, ARCore e ARKit respetivamente, e a Amazon a lançar uma aplicação que permite ver os seus produtos em realidade aumentada, Amazon AR View [16].

2.2 Utilização

O facto de grande parte dos dispositivos móveis atuais suportarem esta tecnologia, proporcionou uma grande evolução da mesma nos últimos tempos. Áreas como a de E-Commerce e de entretenimento foram as que mais beneficiaram, no entanto, com o aumento da acessibilidade de dispositivos próprios para realidade aumentada, como os Microsoft HoloLens, abrem-se portas para as mais diversas áreas.

Na **Saúde**, através da utilização de equipamento especializado, a realidade aumentada pode dar suporte aos profissionais de saúde, aumentando a segurança e proficiência das suas atividades. A utilização dos Microsoft HoloLens para guiar o cirurgião durante uma operação é um dos exemplos de utilização nesta área. Em Portugal os HoloLens estão a ser utilizados para permitir que cirurgião possa ver e localizar o tumor dentro do corpo do doente durante a cirurgia, substituindo técnicas invasivas de localização deste [17], [18]. Outro exemplo é o AccuVein, um aparelho que permite ver um mapa das veias periféricas na superfície da pele [19], [20].



Figura 13 - Uso dos Microsoft HoloLens na preparação de uma operação. Fonte: [18]



Figura 14 - AccuVein. Fonte: [20]

No **Exército** verifica-se o uso da tecnologia como suporte tanto para situações de treino como para situações reais. TAR, *Tactical Augmented Reality*, ou HUD 3.0, são exemplos de tecnologias que fornecem informações ou suporte extra aos militares durante situações reais através de uma interface fornecida pelo capacete [21].

Com o intuito de melhorar a experiência de aprendizagem, a **Educação** é outra área na qual é possível aplicação desta tecnologia. De acordo com o estudo feito por Tasneem Khan, Kevin Johnston e Jacques Ophoff utilizando a aplicação Anatomy4D, o uso de realidade aumentada durante a aprendizagem aumenta a motivação, atenção, confiança e a satisfação dos estudantes [22].

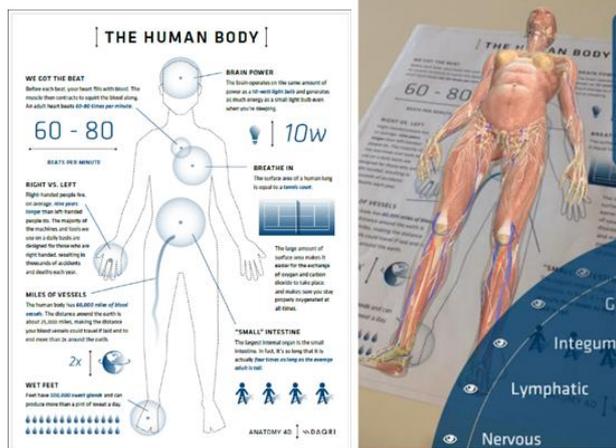


Figura 15 - Aplicação Anatomy4D utilizada no estudo mencionado. Fonte: [22].

A Wikitude [23] é uma empresa provedora de tecnologia de realidade aumentada que num vídeo [24] a demonstrar a capacidade de rastreamento de múltiplos objetos do seu SDK, criou um bom exemplo de aplicação da tecnologia para a educação. Esta permite combinar vários elementos químicos e obter o resultado dessa combinação.

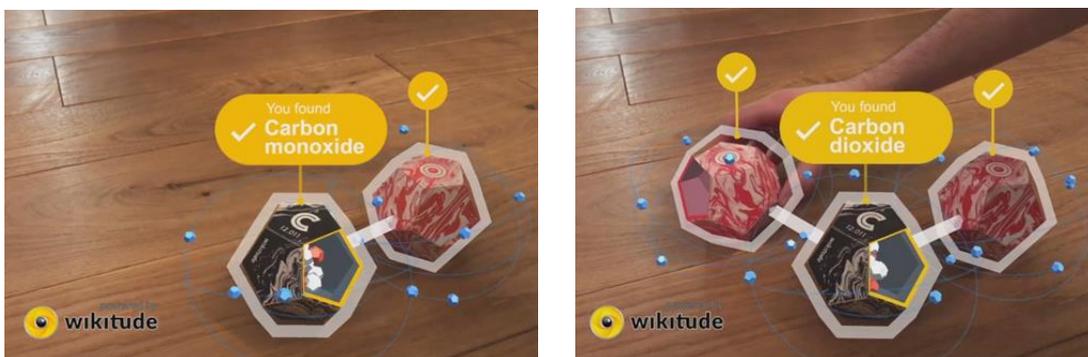


Figura 16 - Aplicação criada pela Wikitude que permite combinar elementos químicos. Fonte: [24]

Atualmente as pessoas cada vez menos têm de sair de casa quando pretendem adquirir algum produto, sendo a aplicação de realidade aumentada em **E-Commerce** um dos fatores que tem influenciado esta realidade. Através desta tecnologia é possível enriquecer a experiência de compras online permitindo por exemplo experimentar calçado, roupa ou acessórios [25]. Seguem-se alguns exemplos de aplicações de entre as inúmeras que tornam mais fácil a compra de produtos online.

IKEA Place permite colocar modelos 3D de mobiliário de escala real num determinado espaço.



Figura 17 - Aplicação IKEA Place. Fonte: [26]

Nike Fit, é uma aplicação para smartphone que para cada uma das sapatilhas fornece o tamanho ideal após examinar os pés do utilizador.

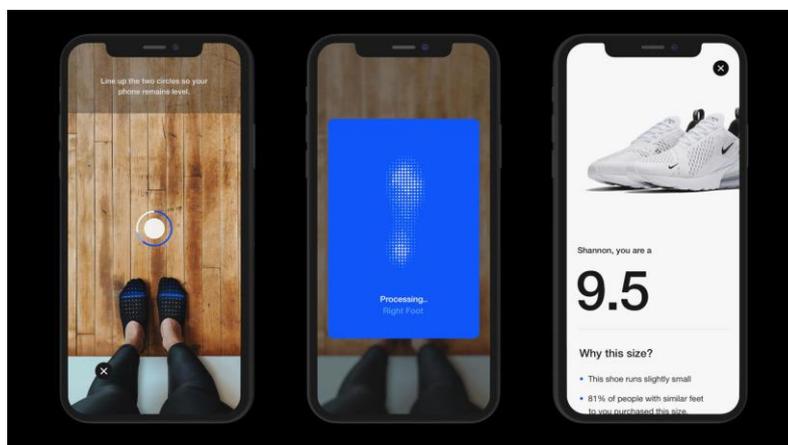


Figura 18 - Aplicação Nike Fit. Fonte: [27]

Warby Parker, uma empresa de venda online de óculos de sol e de prescrição, possui uma aplicação que permite utilizar o smartphone para experimentar os seus óculos.



Figura 19 - Aplicação Warby Parker. Fonte: [28]

O **Entretenimento** é uma das áreas que mais beneficia do acesso à realidade aumentada em dispositivos móveis, tendo por isso crescido bastante nos últimos anos. O enorme sucesso da aplicação Pokemon Go é um bom exemplo deste facto, e desde o seu lançamento em 2016 o número de aplicações a apostar nesta tecnologia cresceu substancialmente [2].



Figura 20 - Pokemon Go
Fonte: [29]



Figura 21- Minecraft Earth, outro exemplo de aplicação. Fonte: [30]

Na **Indústria**, cada vez mais é tirado partido da realidade aumentada de forma a facilitar ou tornar mais rápido, eficiente e seguro os processos de produção e fabrico [2].

A empresa Thyssenkrupp utiliza os Microsoft HoloLens para fornecer uma interface de suporte aos seus profissionais durante a manutenção de elevadores, permitindo ainda a opção de suporte remoto de um especialista em caso de necessidade.



Figura 22 - Interface de suporte, com informações sobre o elevador (Thyssenkrupp). Fonte:[31]



Figura 23 - Videochamada remota com especialista (Thyssenkrupp). Fonte:[31]

A Boeing, empresa de desenvolvimento aeroespacial, utiliza os Google Glass de forma a tornar menos complexo o processo de montagem de fios, fornecendo um guia, através dos óculos, controlável por comandos de voz [32].

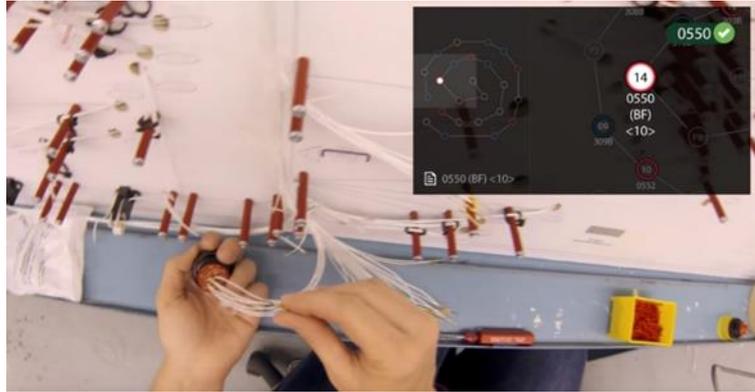


Figura 24 - Uso dos Google Glass pela Boeing. Fonte: [32]

A Porsche utiliza a realidade aumentada como ferramenta de controlo de qualidade dos seus veículos. Através de um tablet ou dos HoloLens conseguem tornar as falhas detetadas visíveis [33].



Figura 25 - Visualização das falhas através dos HoloLens. Fonte: [34]

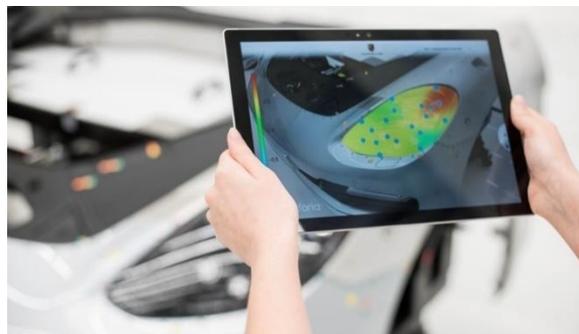


Figura 26 - Visualização das falhas através do tablet. Fonte:[33]

Aplicações na área de **Navegação** têm como objetivo serem fáceis e convenientes de utilizar, proporcionando: uma navegação precisa, identificação de edifícios e portas, e auxílio a alcançar o destino desejado. Podemos dividir a navegação em 2 categorias: navegação no exterior e navegação no interior.

Para navegação no exterior, existem já aplicações como o Google Maps que integram uma componente de realidade aumentada de suporte a utilizadores pedestres. O facto desta aplicação utilizar a informação recolhida pela câmara, de forma a identificar pontos de referência, torna a localização mais precisa do que usar apenas o GPS [35].

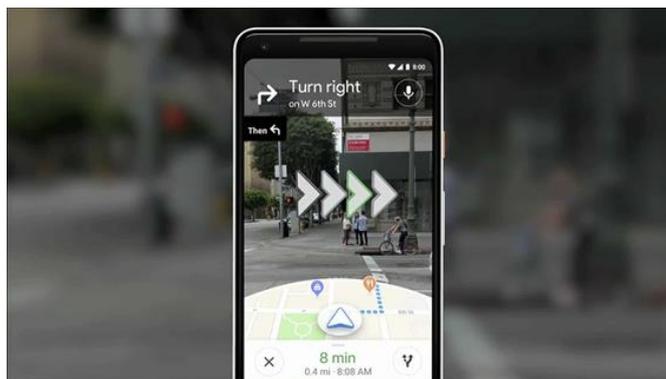


Figura 27 - Funcionalidade de realidade aumentada do Google Maps para utilizadores pedestres. Fonte: [35]

Outro exemplo de navegação no exterior que está a ser explorado, é a utilização dos parabrisas de veículos para fornecer uma interface de navegação ao condutor [35].

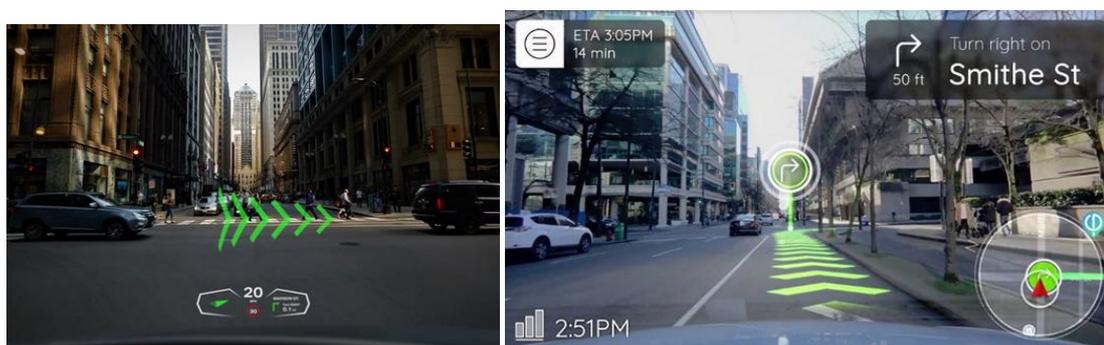


Figura 28 - Exemplos de utilização de uma interface para navegação no para-brisas. Fonte, 1ª imagem:[36],2ª imagem: [35]

Realidade aumentada pode também ser usado para navegação no interior de edifícios. Aqui o GPS poderá não ter a precisão necessária sendo possível a utilização de técnicas adicionais como: Sistema de posicionamento baseado em Wi-Fi, Bluetooth e *Fingerprinting*, bem como técnicas de reconhecimento de imagens de referência de forma a detetar a localização. Para além disso é necessário possuir a planta do edifício [35], [37]. Um exemplo de aplicação é o Guide Bot, que permite criar pontos de interesse para os quais os utilizadores poderão ser direcionados [38].



Figura 29 - GuideBot, aplicação de navegação para interior de edifícios. Fonte: [38]

O uso de realidade aumentada em **Marketing** não está limitado à demonstração ou visualização de produtos como foi visto anteriormente para a área de E-Commerce. Esta possibilita a empresas a oportunidade de criarem experiências memoráveis que geram respostas emocionais, aumentando assim envolvimento que os clientes têm com a marca.

A Jim Beam, criou uma experiência de realidade aumentada diretamente no seu produto, permitindo aos clientes envolverem-se com a história da marca enquanto aproveitam a sua bebida [39].



Figura 30 - História da Jim Beam contada através de realidade aumentada. Fonte: [40]

O Philadelphia Insectarium and Butterfly Pavilion permite aos visitantes posicionarem o seu dispositivo sobre um mural para observar como uma borboleta emerge da sua crisálida em realidade aumentada. Assim conseguiram aumentar a interação e satisfação dos visitantes, e ter uma maior visibilidade e número de visitas graças à partilha nas redes sociais [39].



Figura 31 - Uso de Realidade aumentada no mural do Fonte:[41]

Uma das maiores revistas de viagens da Alemanha, Abenteuer und Reisen, proporciona aos seus clientes experiências de realidade aumentada através das suas revistas, notando-se um aumento dos seus clientes e permitindo a destinos, operadores turísticos, companhias aéreas e hotelarias parceiras, inovar as suas publicidades publicadas diretamente na revista [39].



Figura 32 - Uso de realidade aumentada numa revista turística. Fonte:[42]

2.3 Limites, oportunidades e futuro

Apesar da realidade aumentada ter evoluído bastante nos últimos anos, o facto de ser uma tecnologia relativamente recente leva a que existam algumas limitações ou desafios.

Um dos maiores desafios da tecnologia é o realismo dos elementos virtuais. Este realismo é algo difícil de alcançar, muitas vezes devido à falta de experiência, mas também a limitações impostas pelo hardware presente na maioria dos dispositivos. Existe uma grande variedade de dispositivos com diferentes câmaras, sensores, processadores etc. o que leva a uma grande dispersão nas funcionalidades que cada um é capaz de suportar. Durante a análise das ferramentas de desenvolvimento destinadas à realidade aumentada esta limitação é explorada em maior detalhe [2].

O uso de smartphones para tirar partido da tecnologia é pouco imersivo, especialmente quando comparado com o uso dos óculos. No entanto, apesar destes serem cada vez mais acessíveis, ainda não alcançamos avanços suficientes que justifiquem consumidores comuns a investir nestes dispositivos.

Existe um perigo associado à tecnologia, especialmente quando utilizada nos smartphones, podendo ser uma distração para o que está ao redor do utilizador, ou seja, retira a atenção do utilizador do mundo real [2].

Em algumas das aplicações possíveis, pode ser difícil manter as pessoas a utilizar a aplicação. Um bom exemplo deste facto é uma aplicação com o objetivo de fornecer serviços de navegação no interior de um edifício, da qual o utilizador tira partido até conhecer o mesmo.

Por último, ainda não existe a certeza de como se deve interagir com esta tecnologia. Por exemplo, para computadores e telemóveis existem os teclados *qwerty*, que todos sabem utilizar. No caso da realidade alimentada, sendo esta uma tecnologia bastante recente, não existe nenhum precedente ou contexto de como os utilizadores podem e devem interagir com as oportunidades digitais que realidade aumentada pode oferecer. Existem bastantes possibilidades, toques, gestos, comandos de voz, expressões faciais, ..., sendo que nenhuma destas está bem definida [43].

No entanto apesar destes desafios o futuro desta tecnologia é promissor, com grandes empresas como a Google, Apple e Qualcomm a apostar na tecnologia, fornecendo a capacidade de melhorar não só as experiências exploradas na secção anterior, mas também diversas ações ou atividades do dia a dia. Numa apresentação sobre o futuro da realidade aumentada, a

Qualcomm, uma empresa americana conhecida principalmente pelo desenvolvimento de *chipsets* de smartphones [44], referiram que a aplicação de realidade aumentada iria ter impacto em todos os aspetos da nossa vida. De acordo com a empresa, veremos um aumento na produtividade, eficiência e segurança nas indústrias e empresas graças à aplicação desta tecnologia. Para além disso, o uso que damos a aparelhos como smartphones, computadores, televisões, consolas vai passar a ser feito através de uns óculos de realidade aumentada e/ou virtual.

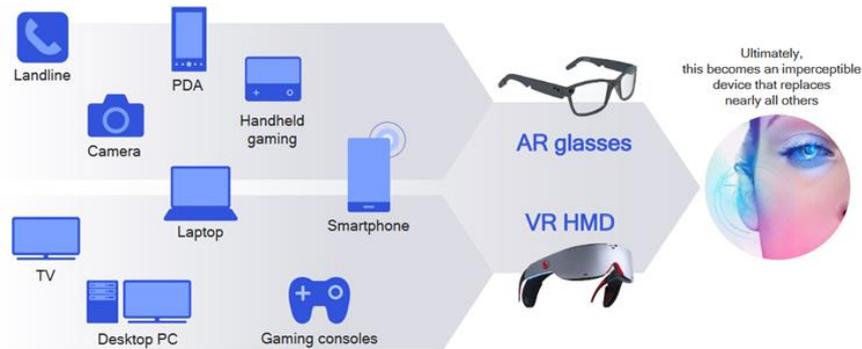


Figura 33 - Tarefas que eram desempenhadas nos dispositivos utilizados atualmente passam a ser feitas usando AR ou VR. Fonte: [45]

Assim espera-se uma evolução de soluções como os óculos, tornando-os leves, confortáveis, móveis, podendo ser utilizados no dia a dia e sendo mais acessíveis.



Figura 34 - Ilustração de uns possíveis futuros óculos de realidade aumentada. Fonte: [45]

Para além da evolução no hardware, esperam-se melhorias que aumentem a imersão durante a utilização. Melhorando o realismo de elementos virtuais, tendo em conta condições de luminosidade, oclusão de objetos, som e a interação, não só com o utilizador, mas também com outros elementos presentes no espaço, tanto reais como virtuais.

Outro aspeto que poderá ter impacto no futuro da realidade aumentada é a chegada de redes 5G. Através destas conseguimos alcançar maiores larguras de banda e menores atrasos na rede, tornando possível a capacidade de recorrer à computação na nuvem, deixando assim de estar totalmente dependente do hardware presente nos óculos ou smartphones.



Figura 35 - Possível separação do processamento, graças aos benefícios da rede 5G. Fonte: [45]

Com a chegada do 5G poderá ser passado parte do processamento da aplicação para o servidor (por exemplo, fazer a análise e reconhecimento de imagem no servidor), deixando assim de estar totalmente dependente do hardware existente nos dispositivos móveis. No entanto isto é algo que não será tido em conta durante o desenvolvimento deste projeto.

2.4 Desenvolvimento de realidade aumentada

Geralmente, existem duas abordagens na aplicação de realidade aumentada: realidade aumentada com base em marcadores (**marker-based AR**) ou sem marcadores (**markerless AR**). Na **marker-based AR** existe um marcador, pode ser uma imagem ou um objeto que tenha características/padrões facilmente distinguíveis, que quando reconhecido desencadeia alguma ação. Enquanto que a abordagem **markerless AR** tira partido de outras técnicas para despoletar eventos [46][47]. Uma componente crítica desta abordagem (**markerless AR**) é saber onde se encontra o dispositivo em relação ao espaço. Para isso existem técnicas como o **SLAM** (*Simultaneous Localization and Mapping*) que permitem ao dispositivo criar um mapa do espaço onde se encontra, e ao mesmo tempo fazer o rastreamento da sua posição nesse espaço. À medida que se obtém mais informações através da câmara do dispositivo o mapa vai sendo atualizado e refinado. Através desta técnica, ao analisar o espaço é possível identificar objetos ou superfícies como o chão ou a parede nos quais é possível, por exemplo, colocar elementos virtuais aos quais fica associado uma coordenada nesse espaço.[4]

Existe ainda outro tipo de abordagem **Location-Based AR**, normalmente considerada um subtipo de **markerless AR**, uma vez que não necessita de marcadores apenas utiliza sensores como o GPS e o giroscópio para determinar a localização e orientação do utilizador e demonstrar conteúdo associado a estes parâmetros[46]. Esta é técnica utilizada por algumas das funcionalidades de realidade aumentada da aplicação Pokemon Go.[4]

É de salientar que a aplicação das várias abordagens não é mutuamente exclusiva, podendo uma aplicação móvel aplicar as várias abordagens ao mesmo tempo. Por exemplo, é possível desenvolver uma aplicação que despolete uma ação perante um marcador ou perante uma localização.

Na próxima secção é feita uma análise dos vários kits de desenvolvimento de software, SDKs, mas antes é importante explorar alguns termos de realidade aumentada que são partilhados entre a maioria destas ferramentas:

- **Feature points**- *Feature points* ou pontos de referência, são zonas ou objetos, num espaço, com características facilmente distinguíveis. Qualquer coisa que seja suscetível de se manter visível e consistente no local onde encontrado, como por exemplo a borda de uma cadeira, um interruptor de luz na parede ou o canto de um tapete. Qualquer componente visual com alto contraste ou textura pode também ser um potencial ponto de referência, como vasos, pratos, copos, estátuas, papel de parede [48].
- **Pose** – Posição e orientação de um objeto num espaço [48].
- **Anchor** – São pontos de interesse, num espaço, definidos pelo utilizador aos quais podem ser associados elementos virtuais. O objetivo destes pontos é de atualizar e rastrear a *pose* dos objetos virtuais associados, à medida que se obtém mais informação sobre o espaço. Por exemplo, digamos que se pretende colocar um objeto digital em cima de uma mesa. Colocando uma *Anchor* nessa mesa, objetos colocados nesta ficam associados à *Anchor*. Desta forma, a posição do objeto vai sendo atualizada caso haja alterações na *pose* da mesa (a posição e orientação de objetos *Trackable*, como a mesa, é atualizada à medida que se obtém mais informação sobre o espaço) [43], [48].
- **Trackable** – *Trackable* ou rastreável, é uma característica de objetos, planos ou pontos que são rastreados ao longo do tempo desde que foram detetados, ou seja, têm sempre uma *pose* definida e atualizada de acordo com a informação já recolhida através de *computer vision*. É normalmente sobre este tipo de objetos que são definidas *Anchor*s [48].
- **Computer vision** – Termo dado ao hardware, software e processos que permitem aos computadores ver e compreender o mundo físico. Por outras palavras é a mistura de inteligência artificial e ciência da computação que visa possibilitar aos computadores a compreender visualmente o mundo tal como um humano, permitindo por exemplo, identificar objetos, condições de luminosidade ou a noção profundidade [43].
- **Occlusion** – É o processo de esconder um objeto atrás de algum elemento, podendo estes ser reais ou virtuais [48].

Ferramentas de Desenvolvimento

Um *Software Development Kit* (SDK) de realidade aumentada fornece as ferramentas e funcionalidades necessárias para desenvolver aplicações que apliquem esta tecnologia, sendo por isso fundamental fazer uma análise das soluções disponíveis de forma a escolher a mais adequada no contexto do projeto do estágio. Tendo em conta a grande quantidade de SDKs existentes, e não sendo possível comparar todas, foi adotada uma metodologia de seleção com base na popularidade/reputação das mesmas, bem como no suporte fornecido e possíveis funcionalidades futuras.

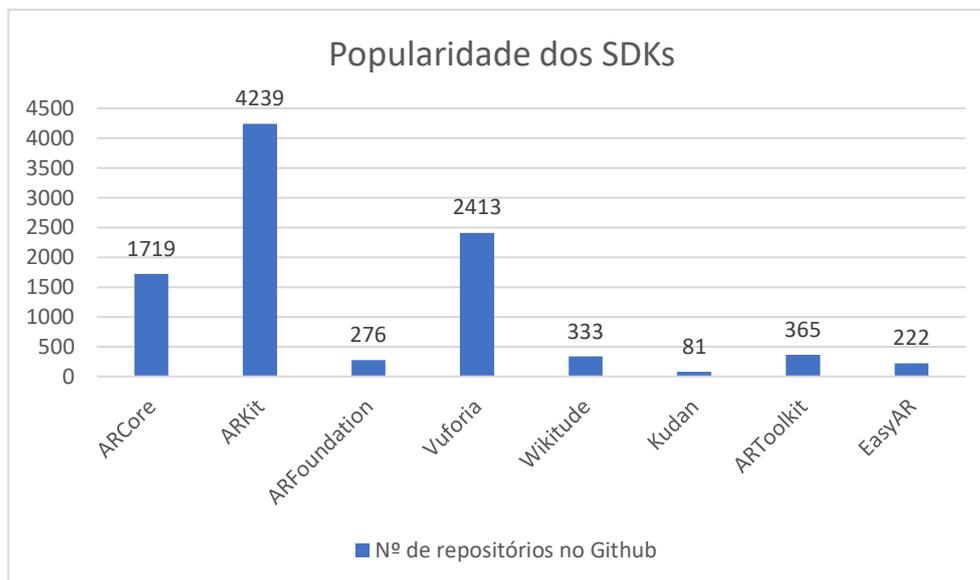


Figura 36 - Popularidade dos SDKs com base no número de repositórios no Github (29/10/2020)

No total foram selecionados 5 SDKs: ARCore, ARKit, ARFoundation, Vuforia e Wikitude para serem analisadas, e foram aplicados os seguintes critérios de comparação:

- **Preço** - O Objetivo é gastar o menos possível alcançando os requisitos do estágio.
- **Plataformas suportadas** - Versões do Android e/ou iOS suportadas.
- **Marker-based AR** - Se fornece funcionalidades que abordem o uso de marcadores.
 - Suporta o reconhecimento e rastreamento de imagens 2D.
 - Suporta o reconhecimento e rastreamento de objetos 3D.
- **Markerless AR** - Se fornece funcionalidades que tirem partido de técnicas como o SLAM, permitindo a aplicação de realidade aumentada sem marcadores.
- **Location-based AR** - Se fornece alguma funcionalidade que tira partido da localização para fornecer conteúdo de realidade aumentada.
- **Integração com o Unity** - Unity é uma plataforma que permite o desenvolvimento de aplicações para múltiplas plataformas em simultâneo, que possui uma vasta documentação, tutoriais e recursos fundamentais. Esta plataforma facilita o trabalho com modelos 3D, assim como a análise e edição de animações para estes, caso necessário.
- **Recursos disponibilizados** - A quantidade e qualidade dos recursos disponibilizados, como documentação, exemplos de implementação, repositórios, são um fator importante, principalmente para quem está a conhecer uma nova tecnologia com a qual nunca teve contacto. Na tabela será fornecida uma nota que indica a quantidade de recursos disponibilizados dentre: documentação com explicação de conceitos, demonstração das funcionalidades e melhores práticas (0-3), exemplos de implementação ou repositórios (0-1), cursos ou guias para quem está a começar (0-1). Infelizmente sem utilizar cada um dos recursos disponibilizados durante a implementação/aplicação das funcionalidades é difícil avaliá-los qualitativamente.
- **Outros** - Na tabela não será exposto, mas durante a análise individual de cada um dos SDKs serão analisadas as várias funcionalidades disponibilizadas por estes.

Tabela 1 - Tabela de comparação das ferramentas de desenvolvimento de realidade aumentada.

	ARCore	ARKit	ARFoundation	Vuforia	Wikitude
Preço ¹	Grátis	Grátis	Grátis	≅424€/ano	2490€
Android	>= 7.0	Não	>=7.0	>=5.1.1	>= 5.0
iOS	>= 11.0 ²	>=11.0	>=11.0	>=12.0	>=11.0
<i>Marker-based</i>	2D	2D e 3D	2D e 3D	2D e 3D	2D e 3D
<i>Markerless</i>	Sim (SLAM)	Sim (SLAM)	Sim (SLAM)	Sim (SLAM)	Sim (SLAM)
<i>Location-based</i>	Não	Sim	Sim	Não	Sim
Unity	Sim	Sim	Sim	Sim	Sim
Documentação	5	4	5	5	5

ARCore

O ARCore [48] é o SDK de realidade aumentada desenvolvido pela Google. É gratuito e funciona com base em 3 capacidades:

- **Motion Tracking:** capacidade de perceber e rastrear a posição do dispositivo relativamente ao espaço. Através da identificação e rastreamento de pontos de recursos visuais distintos (*feature points*), obtidos pela imagem da câmara, em conjunto com informações recolhidas pelos sensores do dispositivo, obtém a posição e orientação (*pose*) da câmara relativamente ao espaço.
- **Environmental understanding:** o ARCore está constantemente a analisar o espaço através da deteção de pontos de recurso (*feature points*) e planos. Estes planos são identificados através dos *features points* que aparentam estar em superfícies horizontais ou verticais comuns. Assim o ARCore consegue detetar o tamanho e localização de todos os tipos de superfícies.
- **Light estimation:** capacidade de estimar as condições de iluminação do ambiente a partir das quais podemos adaptar os elementos virtuais colocados neste.

Para além destas capacidades o ARCore disponibiliza outras funcionalidades:

- **Augmented Faces:** Deteção e rastreamento automático de diferentes regiões da cara, e uso dessas regiões para sobrepor filtros, modelos ou texturas (idêntico a filtros do Instagram).
- **Augmented Images:** Reconhecimento e rastreamento de imagens 2D no ambiente, às quais pode estar associada uma ação como mostrar um objeto 3D, um vídeo, texto, uma animação Deteta até um máximo de 20 imagens em simultâneo.

¹ Preços indicados são referentes ao pacote mais básico que poderá não conter todas as funcionalidades. Esta nota não se aplica a SDKs gratuitas.

² Apenas suporta algumas funcionalidades. Será analisado com maior detalhe na secção seguinte.

As imagens a ser reconhecidas têm de estar presentes numa base de dados local, com uma capacidade de até 1000 imagens, sendo possível a alteração da base de dados durante a execução da aplicação.

O ARCore consegue reconhecer e rastrear tanto imagens paradas como em movimento.

- **Instant Placement:** Permite a colocação instantânea de objetos mesmo antes do ARCore analisar o espaço. A posição e orientação (*pose*) do objeto é recalculada e ajustada à medida que o ARCore obtém mais informação do espaço através da câmara.
- **Depth API:** Consegue estimar a profundidade de objetos e superfícies, permitindo por exemplo que objetos virtuais apareçam realisticamente à frente ou atrás de objetos reais. Não está disponível em todos os dispositivos que suportam o ARCore, para uma lista detalhada e atualizada visitar a referência [49]).

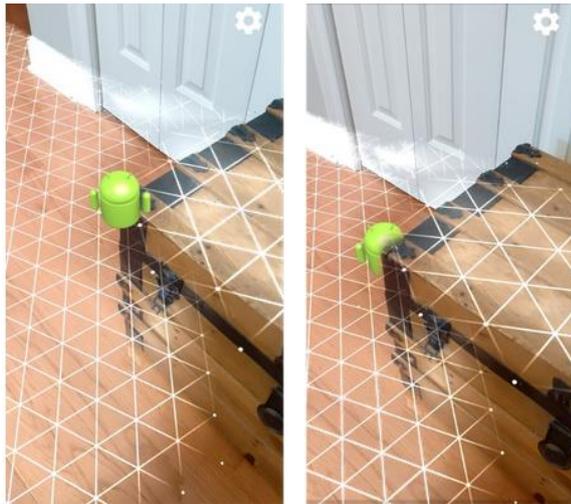


Figura 37 - Imagem da esquerda não tira partido das funcionalidades de profundidade obtendo resultados menos realísticos do que da direita. Fonte: [50]

- **Cloud Anchors:** Permite que múltiplos dispositivos diferentes que se encontrem no mesmo espaço consigam ver e interagir com os mesmos objetos virtuais em simultâneo. Esta funcionalidade é alcançável através da partilha de *Anchors*, e dos objetos associados a estas, entre os utilizadores. Figura 21 é um bom exemplo que demonstra esta funcionalidade. Outro benefício desta funcionalidade é o facto ser possível guardar as *Anchors*, e respetivos objetos associados, na *Nuvem* até 365 dias.
- **User interaction:** A interação do utilizador é implementada através de *hit testing*, em que através da coordenada (x,y) correspondente ao ecrã do telemóvel, projeta um raio (*raycasting*) pela perspetiva da câmara, devolvendo quaisquer planos, pontos de referência (*feature points*), ou elementos virtuais que o raio intersecta e a posição e orientação (*pose*) da interseção. Isto permite aos utilizadores selecionar ou interagir com os elementos virtuais no ambiente.
- **Oriented points:** Permite colocar objetos virtuais em superfícies angulares. Perante um *hit test* que devolve um ponto de referência (*feature point*), são analisados outros pontos de referência perto deste, de forma a estimar o ângulo da superfície no ponto devolvido, obtendo assim uma posição e orientação (*pose*) que tenha esse ângulo em conta.

O ARCore não possui nenhuma funcionalidade de reconhecimento e rastreamento de objetos 3D, mas não impede que esta seja desenvolvida através da informação recolhida pela câmara.

Durante o desenvolvimento deste relatório a Google anunciou uma nova funcionalidade *Earth Cloud Anchors* que se encontra em fase de testes. De acordo com a publicação no blog da Google [51], esta funcionalidade tira partido da *Cloud Anchors* e do GPS para guiar os utilizadores ao conteúdo de realidade aumentada, passando assim o ARCore a fornecer uma funcionalidade de *Location-based AR*. Anteriormente a esta atualização não existia nenhuma funcionalidade que tirasse partido do GPS.

Quanto às plataformas suportadas, esta SDK funciona em dispositivos com uma versão de Android igual ou superior a 7.0 e iOS igual ou superior a 11.0, no entanto existem algumas restrições[49]:

- Apenas as funcionalidades *Augmented Faces* e *Cloud Anchors* estão disponíveis nos dispositivos com iOS.
- No android certas funcionalidades como a *Depth API* só estão disponíveis em determinados dispositivos.

Por fim, em relação aos recursos fornecidos, a documentação aparenta estar bastante completa na qual explicam os conceitos, demonstram as funcionalidades possíveis de implementar e quais as melhores práticas. É disponibilizado um repositório com exemplos de implementação, um guia para iniciantes, e ainda um curso gratuito de introdução ao ARCore feito pela própria Google disponível no Coursera.

ARKit

ARKit [52] é o SDK de realidade aumentada desenvolvida pela Apple, é gratuito, tal como o ARCore, consegue estimar as condições de luminosidade e utiliza técnicas de SLAM para detetar e rastrear a posição e orientação do dispositivo no espaço. Para além disso disponibiliza as seguintes funcionalidades:

- **Image Tracking:** Reconhecimento de imagens e rastreamento da sua posição e orientação. Esta funcionalidade é dividida em duas:

Idêntico à funcionalidade *Augmented Images* do ARCore é possível fazer o reconhecimento de imagens, presentes na base de dados da aplicação, às quais está associada uma ação que é realizada assim que a imagem é detetada. Deteta até 100 imagens em simultâneo e é necessária uma versão do iOS 11.3+.

Outra possibilidade é a deteção e rastreamento de imagens, com forma retangular, no ambiente, sendo possível a aplicação de transformações sobre a imagem detetada. Quando a aplicação deteta uma forma retangular, é possível extrair os dados dos pixels definidos por esta forma, criando a imagem detetada à qual podem ser aplicadas alterações/transformações. Para esta funcionalidade é necessária uma versão do iOS 13.0+.

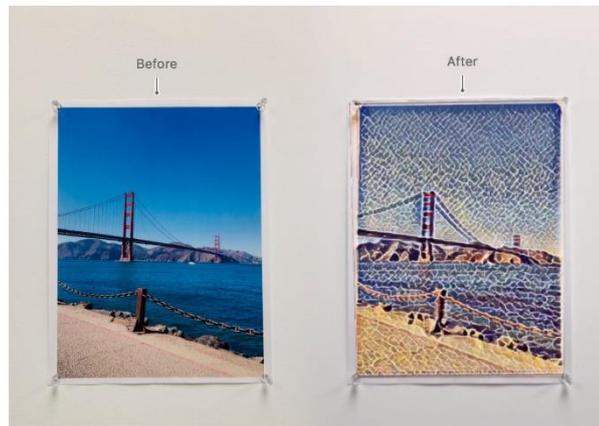


Figura 38 - Exemplo de aplicação de um filtro sobre a imagem detetada. Fonte:[53]

- **Object Detection:** Através do scan de objetos, o ARKit cria pontos de referência desses objetos através dos quais os consegue identificar. Para esta funcionalidade é necessária uma versão do iOS 12.0+.

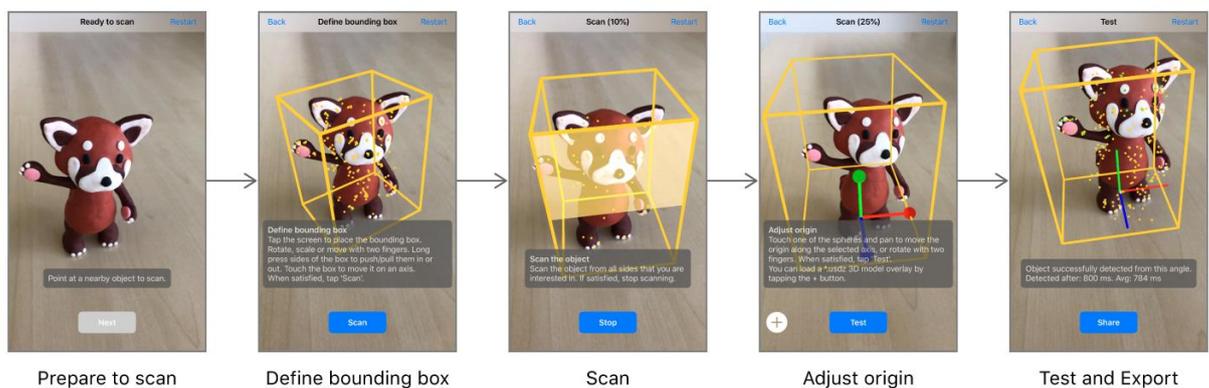


Figura 39 - Scan do objeto para posterior identificação. Fonte:[54]

- **Scene Geometry:** Permite criar um mapa topológico do espaço, através do qual torna possível a oclusão de objetos (objetos virtuais aparecem à frente ou atrás de objetos reais) e a aplicação de físicas a objetos virtuais. Por exemplo, perante um plano inclinado se colocarmos uma bola virtual à qual é aplicada física, esta irá descer ao longo do plano.
- **Depth API -** Utiliza as capacidades do scanner LiDAR para obter informações de profundidade do ambiente. Quando combinado com o *Scene Geometry* torna a oclusões de objetos virtuais mais realista e exata. Esta funcionalidade está disponível apenas em dispositivos com o sensor LiDAR (iPad Pro 11-inch (2nd generation), iPad Pro 12.9-inch (4th generation), iPhone 12 Pro, iPhone 12 Pro Max)).
- **Meshing:** Consegue estimar a forma de elementos físicos, permitindo identificar determinados objetos como mesas, cadeiras ou sofás....
- **People Occlusion:** O ARKit consegue identificar regiões onde se encontram pessoas, prevenindo que elementos virtuais sejam desenhados nestas regiões quando a pessoa se encontra mais perto da câmara do que o elemento virtual. Esta funcionalidade está disponível apenas em dispositivos com iOS 13.0+ com um processador A12 ou mais recente.

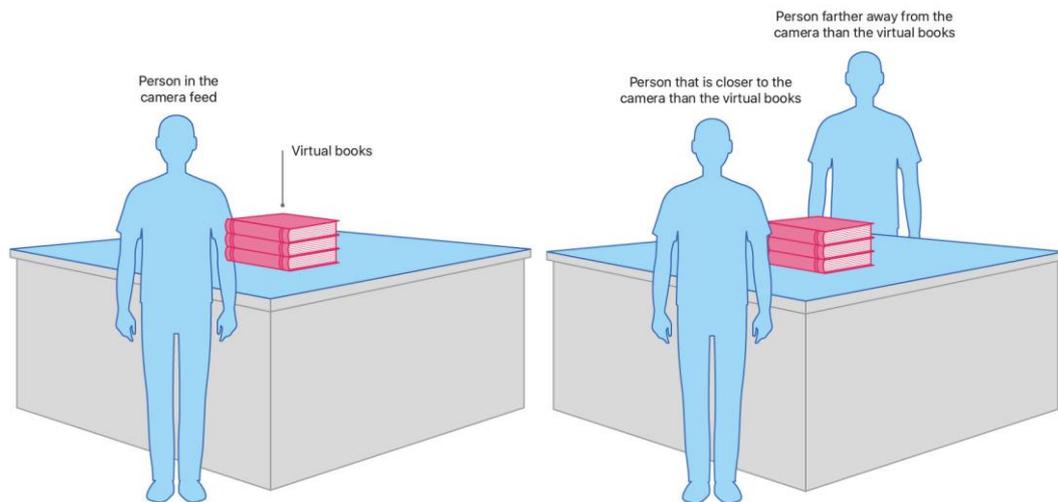


Figura 40 - Esquerda reflete a falta de oclusão. Na direita ocorre a oclusão dos livros virtuais quando a pessoa está mais próxima da câmara. Fonte:[55]

- **Body Tracking (Motion Capture):** Para além de detetar pessoas, o ARKit permite capturar e rastrear os seus movimentos em tempo real. Através desta funcionalidade é possível, por exemplo, fazer a sobreposição do corpo da pessoa com algum efeito, usar os movimentos capturados para interagir com os elementos virtuais, ou, para animar alguma personagem em tempo real imitando estes movimentos. Esta funcionalidade está disponível apenas em dispositivos com iOS 13.0+ com um processador A12 ou mais recente.
- **Simultaneous Front and Back Camera:** É possível a utilização da câmara frontal e traseira em simultâneo. Permitindo por exemplo que o utilizador interaja com o conteúdo de realidade aumentada usando apenas a sua cara. Apenas é mostrado o conteúdo de uma das câmaras ao utilizador. Esta funcionalidade está disponível apenas em dispositivos com iOS 13.0+ com um processador A12 ou mais recente.
- **Face Tracking:** Consegue detetar e rastrear regiões da cara e expressões, permitindo sobrepor as regiões com efeitos, ou usar as expressões para animar personagens em tempo real. Esta funcionalidade está disponível apenas em dispositivos com iOS 11.0+.
- **Collaborative Sessions:** Idêntica ao *Cloud Anchors* do ARCore, permite que múltiplos dispositivos diferentes consigam ver e interagir com os mesmos elementos virtuais. A disponibilidade desta funcionalidade depende do tipo de implementação, se peer-to-peer requer iOS 13.0+, ou host-guest requer iOS 12.0+.
- **Geotracking:** Combina o GPS e outros sensores do dispositivo como o compasso para fornecer experiências de realidade aumentada associadas à localização (*Location-based AR*). Esta funcionalidade está disponível apenas em dispositivos com iOS 14.0+.

Em relação às plataformas suportadas, o ARKit apenas suporta dispositivos com versão do iOS igual ou superior a 11.0 e um processador A9 ou mais recente. No entanto existem algumas das funcionalidades exploradas acima que possuem restrições adicionais. Ao longo da exposição das funcionalidades foram incluídas estas restrições adicionais.

Um dos problemas desta SDK é para além de suportar apenas dispositivos com iOS, é necessário hardware específico da Apple para desenvolver aplicações destinadas ao seu sistema operativo. O Xcode é um programa que disponibiliza um conjunto de ferramentas necessárias para o

desenvolvimento e *deploy* de aplicações para o iOS, sendo necessário um Mac para ter acesso a este programa.

Por último, sobre os recursos disponibilizados, fornecem uma documentação onde explicam conceitos, demonstram as funcionalidades, mas nem sempre referem as melhores práticas para cada uma destas e não fornecem nenhum exemplo de implementação ou repositório. Possuem uma grande biblioteca de vídeos e apresentações, não só sobre o ARKit mas também outras ferramentas de realidade aumentada [56].

AR Foundation

AR Foundation[57] é uma *framework* gratuita, desenvolvida pela Unity Technologies, construída acima do ARKit e do ARCore funcionando como uma interface que permite desenvolver para ambas as plataformas em simultâneo. Esta não fornece nenhuma funcionalidade extra para além das já existentes no ARKit e ARCore, sendo as plataformas suportadas idênticas às respetivas SDKs.

Poderão existir funcionalidades no ARCore ou ARKit que ainda não são suportadas no AR Foundation, como é o caso da funcionalidade Cloud Anchors do ARCore, sendo para estes casos possível a utilização de extensões. Sabendo a plataforma em que aplicação está a ser executada, através das extensões do ARCore e do ARKit, é possível implementar as funcionalidades específicas da respetiva SDK que ainda não estão disponíveis através do AR Foundation

Segue-se uma tabela com as funcionalidades do ARKit e do ARCore já suportadas:

	ARCore	ARKit
Device tracking	✓	✓
Plane tracking	✓	✓
Point clouds	✓	✓
Anchors	✓	✓
Light estimation	✓	✓
Environment probes	✓	✓
Face tracking	✓	✓
2D Image tracking	✓	✓
3D Object tracking		✓
Meshing		✓
2D & 3D body tracking		✓
Collaborative participants		✓
Human segmentation and occlusion		✓
Raycast	✓	✓
Pass-through video	✓	✓
Session management	✓	✓

Figura 41 - Lista das funcionalidades suportadas pelo AR Foundation. Fonte:[57]

Por último, sobre os recursos fornecidos, a Unity Technologies é especialmente conhecida pelos recursos fornecidos aos seus desenvolvedores e para o AR Foundation não é diferente. Existe uma documentação com explicação dos conceitos, demonstração das funcionalidades e melhores práticas de realidade aumentada, possui ainda um repositório com exemplos de implementação, bem como guias e cursos.

Vuforia

Vuforia[58] é uma das SDKs mais populares quando se fala em *marker-based AR*. Grande parte das suas funcionalidades vão de encontro a este tipo de realidade aumentada, sendo por isso focada essencialmente na deteção e rastreamento de imagens e objetos. Recentemente esta SDK tem sido atualizada, proporcionando outras funcionalidades onde não é necessário o uso de marcadores, como por exemplo o reconhecimento de planos. Seguem-se as funcionalidades disponíveis:

- **Model Targets:** Permite reconhecer e rastrear objetos através da sua forma usando modelos 3D pré-existentes. Para garantir o funcionamento adequado é necessário que o objeto seja rígido, ou seja, não pode ser maleável ou deformável, e apresente características de superfície estáveis (não suporta superfícies brilhantes).

Para gerar os modelos 3D é fornecida uma ferramenta junto com o SDK, **Model Target Generator (MTG)**.

- **Area Targets:** Permite rastrear e associar elementos virtuais a áreas e espaços. Para isso é necessário um scan 3D de todo o espaço e associar os elementos virtuais a objetos estacionários pertencentes a esse espaço.

O scan tem de ser feito usando um dos seguintes aparelhos, um tablet iPad Pro com scanner LiDAR ou um Matterport™ Pro2 3D Camera. O Modelo digital gerado por estes aparelhos é depois importado para uma ferramenta fornecida junto com o SDK, **Area Target Generator**, que gera os ficheiros necessários.

- **Image Targets:** Reconhecimento e rastreamento de imagens. Para criar a base de dados de imagens é necessário utilizar a ferramenta Vuforia Target Manager. As imagens têm de estar no formato JPG ou PNG, devem ser menores do que 2.25MB e possuir um comprimento superior a 320 pixéis.
- **Object Targets:** Reconhecimento de objetos 3D de pequenas dimensões. O objeto deve ser opaco, rígido e conter nenhuma ou poucas peças móveis (caso tenha zonas móveis ou deformáveis é possível apenas usar as zonas do objeto que não o são para tentar identificar). A superfície do objeto deve ter características com contraste e textura.

É necessário fazer o scan do objeto usando a aplicação Object Scanner fornecida.

- **Multi-Targets:** Multi-Targets é uma coleção de Image Targets combinadas numa disposição geométrica definida, como por exemplo uma caixa. Esta funcionalidade permite rastrear e detetar a partir de qualquer um dos lados geométricos. Igual à *Image Targets* as imagens têm restrições de tamanho, formato e devem ser guardadas na base de dados. Para além disso, as dimensões de cada imagem e o formato geométrico final têm de estar definidos.



Figura 42 - Exemplo de aplicação da funcionalidade Multi-Targets. Fonte:[59]

- **Cylinder Targets:** Consegue reconhecer e rastrear imagens enroladas em formas cilíndricas ou cónicas. É necessário utilizar a ferramenta Vuforia Target Manager para criar a base de dados de imagens sendo que estas têm de estar no formato JPG ou PNG e devem ser menores do que 2.25MB.

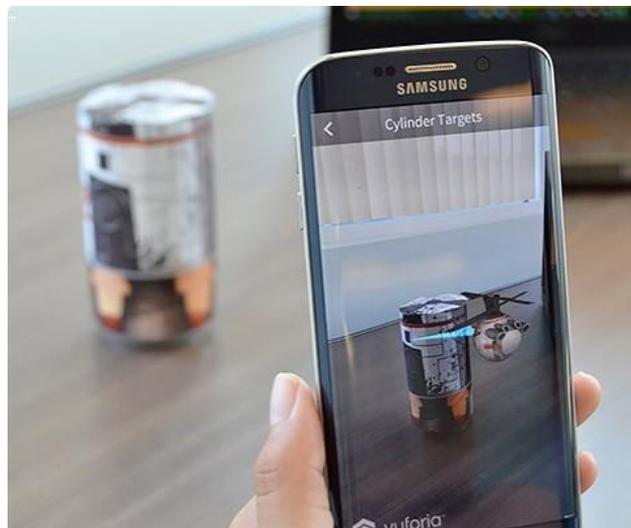


Figura 43 - Exemplo de aplicação da funcionalidade Cylinder Targets. Fonte:[60]

- **VuMarks:** Opção de criar marcadores usando as ferramentas fornecidas, como o VuMark Designer, em vez de utilizar uma imagem. O mesmo design, com pequenas alterações, pode ser utilizado para guardar dados ou para despoletar ações de realidade aumentada diferentes, idêntico a um QRCode.



Figura 44 - Exemplos de VuMarks. Fonte:[61]

- **External Camera:** Permite utilizar uma câmara externa, que não pertença ao smartphone ou tablet que está a ser utilizado. Algumas funcionalidades poderão não funcionar, como é o caso da Gound Plane, e outras funcionam de forma menos robusta. É necessário que a câmara externa implemente APIs que o Vuforia usa para aceder ao sistema externo.
- **Ground Plane:** Permite detetar e colocar elementos virtuais em superfícies horizontais como o chão ou uma mesa. Esta funcionalidade tira partido do *Vuforia Fusion*.

Vuforia Fusion é um conjunto de tecnologias que tem como objetivo oferecer a melhor experiência de realidade aumentada de acordo com a capacidade do dispositivo que está a ser utilizado. Funciona como uma API que tira partido de tecnologias como o ARKit ou ARCore, VISLAM ou SLAM de acordo com as capacidades do dispositivo.

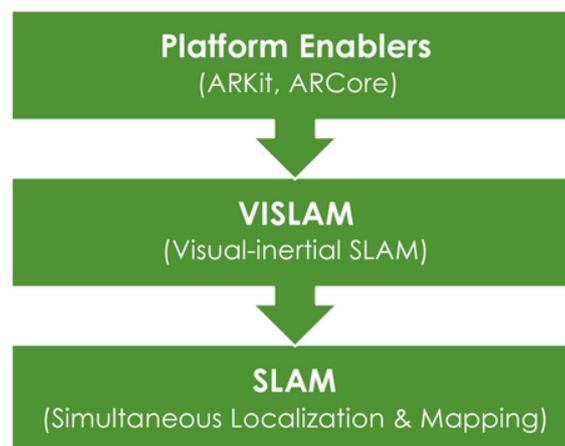


Figura 45 - Prioridade de escolha de acordo com a capacidade do dispositivo. Fonte:[62]

O *Vuforia Fusion* tenta sempre usar a tecnologia do topo e vai descendo de acordo com o que está disponível no dispositivo. Por exemplo perante uma aplicação a ser executada num iPhone X, irá tirar partido automaticamente do ARKit para todas as funcionalidades dependentes do *Vuforia Fusion*. A mesma aplicação a correr num dispositivo iOS mais antigo que não suporte o ARKit mas que suporte VISLAM, usará esta tecnologia para as funcionalidades dependentes.

VISLAM - *Visual-Inertial Simultaneous Localization and Mapping* é um algoritmo implementado pela Vuforia que combina os benefícios de *Visual-Inertial Odometry* (VIO, o uso de dados, fornecidos por sensores de movimento, para estimar alterações de posição ao longo do tempo) e *Simultaneous Localization and Mapping* (SLAM). Este algoritmo é suportado em dispositivos com os sensores necessários e calibrados pela Vuforia.

- **Cloud Recognition:** Permite guardar e gerir *Image Targets* online, deixando de existir uma base de dados local no dispositivo utilizado.

Quanto ao preço, o desenvolvimento de uma aplicação que tire partido do Vuforia é gratuito, apenas se paga uma licença a partir da publicação da aplicação (cada aplicação tem a sua própria licença). De acordo com as funcionalidades disponibilizadas, estão disponíveis várias licenças com preços diferentes.

Features	Pro	Agency Package	Basic + Cloud	Basic	Lab Pack	University Pack	Developer
Publishing for Commercial Use	✓	✓	✓	✓	✓ 50 Licenses	✓ 500 Licenses	Can't Publish
Image Targets VuMarks Multi-Targets Cylinder Targets	✓	✓	✓	✓	✓	✓	✓ 100 VuMarks
Object Targets	✓	✓	✓	✓	✓	✓	✓
Model Targets	✓	Add on	—	—	✓	✓	✓ 20
Area Targets	✓	Add on	—	—	✓	✓	✓ 10
Advanced and External Camera APIs	✓	Add on	—	—	✓	✓	✓
Cloud Database	✓ Custom	✓ Custom	✓ 10,000 recos per month 100,000 Images	—	✓	—	✓ 1,000 recos per month 1,000 Images
Production Support	✓	✓	—	—	—	—	—

Figura 46 - Diferentes licenças e respetivas funcionalidades. Fonte:[63]

Em relação aos dispositivos suportados, o Vuforia suporta dispositivos com um sistema operativo Android 6.0 ou superior (com uma arquitetura de 32 ou 64 bits) e iOS 12 ou superior (com uma arquitetura de 64 bits). A funcionalidade *Vuforia Fusion* tal como dito anteriormente funciona em dispositivos com ARCore e ARKit ou dispositivos que suportem o VISLAM.

O Vuforia pode ser utilizado em conjunto com o AR Foundation no Unity.

Por último, sobre os recursos disponibilizados, possuem uma documentação que explica os conceitos, demonstra as funcionalidades e as melhores práticas. Possui exemplos de implementação e guias para iniciantes.

Wikitude

O SDK Wikitude[64] foi dos primeiros kits de desenvolvimento de realidade aumentada para Android e iOS. Inicialmente era focado em fornecer funcionalidades de *location-based AR*, mas

atualmente as suas funcionalidades abrangem os dois tipos de realidade aumentada, *markerless AR* e *marker-based AR*.

Começando pelas funcionalidades que se baseiam no uso de marcadores, temos:

- **Object Tracking:** Permite reconhecer e rastrear múltiplos objetos 3D em simultâneo. Cada um destes objetos deve ser previamente definido criando um *object target* para cada. Existem duas formas de gerar *object targets*, através de um vídeo do objeto ou de um conjunto de imagens desse objeto.
- **Scene Tracking:** Permite rastrear e associar elementos virtuais a áreas, espaços ou objetos de grandes dimensões. Para isso é necessário um scan 3D do espaço, e que haja pouca variação entre o mapa gerado e o espaço digitalizado.
- **Image Tracking:** Permite reconhecer e rastrear várias imagens em simultâneo. Para isso é necessária uma base de dados das várias imagens podendo esta ser local (até 1000 imagens) ou tirar partido da funcionalidade *Cloud Recognition*.
- **Cloud Recognition:** Permite fazer o reconhecimento de mais de 1000 imagens online.
- **Cylinder Tracking:** Permite reconhecer e rastrear imagens enroladas em formas cilíndricas, sendo possível o rastreamento de vários cilindros em simultâneo.

Passando para as funcionalidades que não necessitam de marcadores, temos:

- **Instant Tracking:** Permite rastrear a posição do dispositivo ao analisar o ambiente em que está inserido, detetando pontos de referência (*feature points*) e os respetivos planos (SLAM). Esta funcionalidade tira partido da funcionalidade Wikitude SMART.

A **Wikitude SMART** é idêntica ao Vuforia Fision, no sentido de ser uma API que tira partido do ARCore e do ARKit quando suportados, para aplicar técnicas de SLAM. Quando não são suportados usa a implantação de SLAM da Wikitude.

- **Geo-AR:** Esta é a funcionalidade de *location-based AR* fornecida. Permite associar o conteúdo de realidade aumentada a pontos geográficos de interesse. Infelizmente esta funcionalidade não é ainda suportada oficialmente pela API da Wikitude para Unity.

O SDK Wikitude está dividido em várias componentes possibilitando várias abordagens para a criação de aplicações de realidade aumentada.

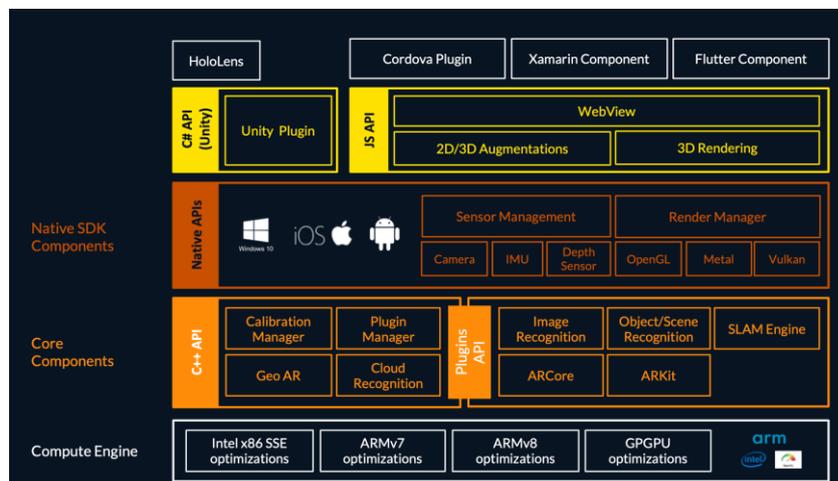


Figura 47 - Arquitetura do Wikitude. Fonte: [65]

Para além da API disponibilizada para o desenvolvimento em Unity, o Wikitude disponibiliza extensões para o desenvolvimento em Flutter, Xamarin e Cordova. Estas extensões estão construídas acima de uma API em JavaScript. Neste momento apenas aplicações que são desenvolvidas através de *frameworks* que tirem partido desta API de JavaScript conseguem fornecer a funcionalidade Geo-AR.

Existe a possibilidade deste SDK ser usado em conjunto com o AR Foundation.

Em relação aos dispositivos suportados, para aplicações desenvolvidas em Unity que tiram partido das componentes nativas temos:

	Sensor-based AR (Geo-AR)	Image recognition and tracking	Instant tracking & Object and Scene tracking
ANDROID (NATIVE API)	<ul style="list-style-type: none"> not available 	<ul style="list-style-type: none"> Android 5.0 (API Level 21) or newer High resolution devices (hdpi) Camera Devices with a capable CPU (armv7a with NEON support or armv8a) e.g. 	<ul style="list-style-type: none"> Android 5.0 (API Level 21) or newer High resolution devices (hdpi) Camera Devices with a quad-core CPU (armv7a with NEON support or armv8a) e.g. <p>Additional for Instant Tracking:</p> <ul style="list-style-type: none"> Compass Accelerometer
IOS (NATIVE API)	<ul style="list-style-type: none"> not available 	<ul style="list-style-type: none"> iOS 12.0 or newer Camera Devices with a 64-bit CPU <ul style="list-style-type: none"> iPhone 5s or newer iPad Air (2013) or newer 	<ul style="list-style-type: none"> iOS 12.0 or newer Camera Devices with a 64-bit CPU <ul style="list-style-type: none"> iPhone 5s or newer iPad Air (2013) or newer

Figura 48 - Dispositivos suportados através da API do Wikitude para Unity. Fonte: [66]

Sobre os preços, existem várias opções, de taxa única ou subscrições anuais, que diferem em 2 fatores: se incluem atualizações futuras do SDK e se incluem o serviço de *Cloud Recognition*. Existe ainda a possibilidade de fazer um pacote personalização para empresas.

	PRO 3D	PRO 3D	CLOUD	ENTERPRISE
	<p>ONE-TIME FEE SDK updates not included</p>	<p>SUBSCRIPTION SDK updates included</p>	<p>SUBSCRIPTION Cloud Hosting</p>	<p>Custom built package, required for smart glasses, enterprise apps, and 1 million+ installs</p>
	<p>2490 €</p>	<p>2990 € / yr</p>	<p>4490 € / yr</p>	<p>custom</p>
	<p>VIEW PRODUCT</p>	<p>VIEW PRODUCT</p>	<p>VIEW PRODUCT</p>	<p>VIEW PRODUCT</p>

Figura 49 - Preços das opções disponibilizadas do Wikitude. Fonte:[67]

Por último, sobre os recursos disponibilizados: possuem uma documentação que explica os conceitos, demonstra as funcionalidades e as melhores práticas, possui exemplos de implementação, guias para iniciantes e vários tutoriais.

Opiniões externas e Conclusões

Para terminar o estudo do estado da arte, foi feita uma pesquisa por referências e análises/opiniões externas sobre as ferramentas de desenvolvimento de realidade aumentada. De notar que, muitos dos estudos encontrados estão já desatualizadas, existindo por isso certas funcionalidades que na altura não eram suportadas, mas que atualmente foram atualizadas ou adicionadas. Para além disso, determinadas ferramentas mais recentes, como o ARCore ou o ARKit, não foram incluídas em alguns destes estudos. No entanto, esta pesquisa permitiu perceber quais seriam os fatores mais importantes, para a escolha da ferramenta de desenvolvimento a utilizar, tendo conta os requisitos do nosso sistema.

O primeiro estudo encontrado consiste num comparativo das ferramentas quando o objetivo é desenvolver aplicações de realidade aumentada com intuito educacional [68]. Neste estudo concluíram que um dos fatores mais importantes na ferramenta de desenvolvimento seria a facilidade de uso das mesmas sem a necessidade de programar, uma vez que os educadores seriam os responsáveis por criar as experiências, sendo que, estes poderiam não ter conhecimentos prévios da tecnologia. Para a nossa solução este não seria dos aspetos mais importantes, uma vez que foi criada uma componente que facilite a criação de experiências sem ser necessário conhecer ou interagir com as ferramentas ou tecnologias utilizadas pelo sistema.

Noutro estudo [69], os autores limitaram-se a avaliar as várias ferramentas escolhidas, sem qualquer contexto ou objetivo. Algumas das conclusões tiradas, neste estudo, sobre o Vuforia e o Wikitude foram idênticas às recolhidas durante o estudo do estado da arte.

Por último, temos um estudo com o objetivo de perceber qual das ferramentas de realidade aumentada seria a ideal para desenvolver uma aplicação educacional destinada à área de química [70]. Tratando-se de uma investigação mais recente, ferramentas como o ARCore e o ARKit já foram incluídas, no entanto existem determinados aspetos que já se encontram desatualizados. Este estudo determinou que o Vuforia seria a escolha ideal devido aos seguintes fatores:

- Permitir o desenvolvimento para múltiplas plataformas em simultâneo - É verdade que tanto o ARCore como o ARKit apenas permitem o desenvolvimento para Android e iOS respetivamente (salve certas funcionalidades do ARCore que funcionam em iOS), no entanto o AR Foundation, tratando-se de uma interface construída acima destas, permite também desenvolver para ambas as plataformas em simultâneo tirando partido da ferramenta de acordo com o sistema onde está a ser utilizado.
- Permitir manter o rastreamento dos alvos, mesmo estando estes fora de visão – atualmente qualquer uma das ferramentas estudadas possui esta funcionalidade.
- Permitir rastrear múltiplos alvos em simultâneo – atualmente qualquer uma das ferramentas disponibiliza esta funcionalidade.
- Oferecer a funcionalidade *Cloud Database* – esta é uma das funcionalidades que apenas o Wikitude e Vuforia possuem.

Capítulo 3

Levantamento de requisitos

Ao longo deste capítulo são explorados o âmbito e objetivo do projeto, e é feita a especificação e análise tanto dos requisitos funcionais, incluindo a sua priorização, como dos atributos de qualidade. Por último, ainda neste capítulo, são especificados vários casos de uso.

3.1 Nomenclatura e Convenções

De forma a definir os requisitos de forma concisa, foram definidas nomenclaturas e convenções utilizadas não só ao longo desta secção, mas também em toda a scope do projeto:

- **Trigger:** Imagem que ao ser reconhecida pela aplicação despoleta uma ou mais Ações.
- **Ação:** Componente de realidade aumentada que é gerada ou mostrada após o reconhecimento de um Trigger. Pode ser uma imagem, áudio, vídeo, ou um modelo 3D.
- **Campanha:** Conjunto de Triggers, e respetivas Ações, agrupados de acordo com a iniciativa que está a decorrer, durante um determinado período definido. Dando um exemplo, a Campanha natalícia é um conjunto de Triggers e respetivas Ações utilizados durante a época natalícia de 1 a 31 de dezembro. Uma Campanha diz-se ativa durante o seu período definido.

Quanto à priorização de requisitos, optámos por utilizar uma técnica semelhante à *MoSCoW*, apenas alterando a última categoria:

- **Must have:** requisito mandatório para o funcionamento da solução proposta.
- **Should have:** requisito importante, mas não essencial para o funcionamento da solução.
- **Could have:** requisito desejado, menos importante que o *Should have*
- **Wish list:** requisito acordado como bónus, apenas implementado se houver tempo suficiente, podendo constituir uma oportunidade futura.

3.2 Contextualização

Tal como referido na introdução, o objetivo do projeto é tirar partido das várias iniciativas preparadas pela Critical Software, melhorando as experiências através de realidade aumentada. Para isso, a equipa responsável por estas iniciativas, deverá conseguir definir:

- As imagens, que serão utilizadas como Trigger e que, ao serem reconhecidas, despoletam Ações.
- As Ações que serão despoletadas após o reconhecimento de cada Trigger, sendo que, diferentes Triggers podem despoletar Ações diferentes. Estas ações devem ter várias opções de customização, de forma a dar a liberdade necessária para criar as

experiências de realidade aumentada desejadas. Alguns destes parâmetros de customização são especificados na secção seguinte de análise de requisitos.

- As Campanhas que desejarem, com uma data de início e de fim, sendo estas editáveis.

É através de uma aplicação móvel que os utilizadores são capazes de visualizar as Campanhas preparadas e que se encontram ativas. Esta aplicação deve ser capaz de reconhecer os Triggers destas Campanhas e mostrar as respetivas Ações associadas.

A criação e visualização destas Campanhas, deve ser possível, sem haver a necessidade de um nova compilação e respetiva instalação da aplicação, tornando o sistema dinâmico, no qual pode ser utilizado qualquer conteúdo desejado. Desta forma, não existe a limitação de apenas ser possível escolher conteúdo (Triggers e Ações) que seja compilado juntamente com a aplicação, podendo novos conteúdos ser adicionados a qualquer momento.

Ao mesmo tempo o sistema deve estar construído de forma que não seja necessário ter conhecimento sobre o seu funcionamento ou sobre realidade aumentada, para que seja possível tirar partido das funcionalidades descritas.

3.3 Análise de requisitos funcionais

Requisitos funcionais definem as funcionalidades que o sistema deve disponibilizar. O sistema deste projeto é composto por 3 grandes componentes:

- **Backend:** responsável por garantir o dinamismo, a modificabilidade e customização do sistema, evitando a necessidade de novas compilações da aplicação móvel.
- **Interface de Gestão do Conteúdo:** responsável por facilitar, às equipas responsáveis pelas iniciativas, a interação com o Backend para fazer a gestão do conteúdo.
- **Aplicação móvel:** responsável por fornecer o conteúdo de realidade aumentada definido.

De forma a diferir os vários utilizadores e respetivas permissões e privilégios sobre o sistema como um todo, foram definidos 2 níveis de acesso ao sistema:

- **Gestor:** Capaz de realizar todas as operações disponibilizadas pelo Backend através da Interface de Gestão do Conteúdo.
- **Utilizador:** Qualquer utilizador que possua a aplicação instalada no seu dispositivo móvel e pretenda usufruir das experiências de realidade aumentada definidas.

Os requisitos definidos foram agrupados de acordo com as componentes a que se destinam.

Requisitos do Backend e da Interface de Gestão do Conteúdo

Na tabela 2 são especificados os requisitos funcionais do Backend e da Interface de Gestão do Conteúdo, e respetivas prioridades. Para uma melhor visualização verificar [apêndice A](#).

Tabela 2 - Tabela de requisitos funcionais do Backend e da Interface de Gestão do Conteúdo, e respetivas prioridades.

Requisitos Funcionais				
Componente	ID	Nome	Descrição	Prioridade
Backend + Interface de Gestão do Conteúdo	RF-1	Adicionar novos Triggers	Através destas Componentes, qualquer Gestor deve ser capaz de adicionar novos Triggers.	Must
	RF-2	Remover Triggers	Através destas Componentes, qualquer Gestor deve ser capaz de remover Triggers previamente adicionados.	Must
	RF-3	Avaliar uma imagem	Através destas Componentes, qualquer Gestor deve ser capaz de avaliar se uma imagem pode ser utilizada como Trigger e se possui boa qualidade para	Should
	RF-4	Adicionar novas Ações	Através destas Componentes, qualquer Gestor deve ser capaz de adicionar novas Ações.	Must
	RF-5	Remover Ações	Através destas Componentes, qualquer Gestor deve ser capaz de remover Ações previamente adicionadas.	Must
	RF-6	Criar Campanhas	Através destas Componentes, qualquer Gestor deve ser capaz de criar Campanhas com uma data de início e de fim.	Must
	RF-7	Editar Campanhas	Através destas Componentes, qualquer Gestor deve ser capaz de editar as Campanhas.	Should
	RF-8	Remover Campanhas	Através destas Componentes, qualquer Gestor deve ser capaz de remover Campanhas criadas (mesmo estando ativas).	Must
	RF-9	Associar Triggers a Campanhas	Através destas Componentes, qualquer Gestor deve ser capaz de associar um ou mais Triggers a Campanhas existentes.	Must
	RF-10	Associar Ações a Triggers	Através destas Componentes, qualquer Gestor deve ser capaz de associar uma ou mais Ações a Triggers existentes numa determinada Campanha.	Must
	RF-11	Remover um Trigger de uma Campanha	Através destas Componentes, qualquer Gestor deve ser capaz de remover um Trigger de uma Campanha.	Must
	RF-12	Remover uma Ação associada a um Trigger	Através destas Componentes, qualquer Gestor deve ser capaz de remover uma Ação de um Trigger existente numa determinada Campanha.	Must
	RF-13	Ações customizáveis	As Ações devem possuir vários parâmetros de customização.	Must
	RF-14	Alterar a customização de cada Ação	Através destas Componentes, qualquer Gestor deve ser capaz de alterar os vários parâmetros de customização de cada Ação.	Must
	RF-15	Customizações diferentes para a mesma Ação	A mesma Ação quando associada a Triggers diferentes pode ter parâmetros de customização diferentes.	Should
	RF-16	Obter todas as Campanhas ativas	Através do Backend deve ser possível obter todas as Campanhas ativas, e fazer o download dos respetivos Triggers e Ações.	Must
	RF-17	Testar Campanha	Qualquer Gestor deve ser capaz de testar as Campanhas criadas antes de as tornar públicas.	Could
	RF-18	Processar conteúdo	O Backend deve ser capaz de preparar / processar o conteúdo (Triggers e Ações) para que este possa ser utilizado pela aplicação.	Must

Seguem-se anotações para alguns dos requisitos, de forma a detalhar determinados aspetos, especificar o comportamento do sistema, ou apontar certas restrições/limitações associadas a cada um.

RF-1: Adicionar novos Triggers

Tal como referido anteriormente Triggers são imagens que podem ser adicionadas através do Backend e/ou da Interface de Gestão do Conteúdo. São aceites imagens com as seguintes características:

- **Formatos:** PNG, JPEG ou JPG
- **Resolução**
 - **Min:** 300x300

Após adicionado um Trigger, este deve ser processado de forma que possa ser utilizado pela aplicação, sendo que, só pode ser associado a uma Campanha após este processo estar concluído.

RF-2: Remover Triggers

Após removido um Trigger, este deve ser removido de todas as Campanhas às quais se encontra associado.

RF-3: Avaliar uma Imagem

Ao ser adicionado um novo Trigger, o sistema automaticamente deve fazer uma avaliação da imagem indicando não só se esta pode ser utilizada como Trigger, mas também o quão “reconhecível” ela é.

RF-4: Adicionar novas Ações

Tal como os Triggers as Ações podem ser adicionadas através do Backend e/ou da Interface de Gestão do Conteúdo, sendo aceites os seguintes formatos de ficheiros:

- **Áudio**
 - **Formatos:** Mp3

- **Vídeo** (com ou sem transparência)
 - **Formatos:** Mp4 ou Webm
- **Imagem** (com ou sem transparência)
 - **Formatos:** PNG, JPEG ou JPG.
- **Modelo 3D** (com ou sem animações)
 - **Formatos:** fbx.

Após adicionada uma Ação, esta deve ser processada de forma que possa ser utilizada pela aplicação, sendo que, só pode ser utilizada após este processo estar concluído.

RF-5: Remover Ações

Após removida uma Ação, esta deve ser removida de todos os Triggers, aos quais foi associada.

RF-9 Associar Triggers a Campanhas

Uma Campanha pode possuir um ou mais Triggers, no entanto antes de associar um Trigger é necessário validar se este já está a ser utilizado, não só pela Campanha onde está a ser adicionado, mas também por outras Campanhas que possam estar ativas durante o mesmo período.

RF-10 Associar Ações a Triggers

Um Trigger pode possuir uma ou mais Ações associadas, no entanto, existe uma limitação que deve ser validada: o mesmo Trigger não pode ter mais do que uma Ação do mesmo formato (por exemplo, não pode mostrar dois vídeos ao mesmo tempo, no entanto pode mostrar um Modelo 3D juntamente com Áudio).

RF-13: Ações customizáveis

Cada Ação terá parâmetros de customização de acordo com o seu tipo/formato. Seguem-se alguns parâmetros definidos, juntamente com uma tabela que indica a que tipo de Ações estes parâmetros serão aplicáveis:

- **Editável / Transformável:** Propriedade que indica se o utilizador da aplicação móvel pode rotacionar ou alterar a escala da Ação.
- **Clone:** Propriedade que indica se a Ação pode ser clonada e colocada em qualquer lugar à escolha do utilizador, deixando assim de estar dependente da imagem.
- **Loop:** Indica se o vídeo / áudio deve ou não ficar em loop.
- **Painel de controlo:** Permite definir se, o Vídeo utilizado como Ação, deve ou não ter um painel de controlo que permita: pausar/continuar a reprodução e verificar o progresso do vídeo
- **Transparência:** Permite definir se a Ação deve manter a Transparência (caso tenha).
- **Manter Proporção:** Permite definir se a Ação deve manter a sua proporção original, dentro dos limites do Trigger, ou se deve adaptar-se de forma a ocupar todo o tamanho do Trigger.

Tabela 3 - Parâmetros de customização aplicáveis para cada tipo de Ação.

Parâmetros de Customização	Imagem	Vídeo	Áudio	Modelo 3D
Editável / Transformável	✓	✓	×	✓
Clone	✓	✓	×	✓
Loop	×	✓	✓	×
Painel de controlo	×	✓	×	×
Transparência	✓	✓	×	×
Manter Proporção	×	✓	×	×

Foram removidos os seguintes parâmetros:

Tempo início: Permite definir o tempo a partir do qual se iniciará a reprodução do áudio ou vídeo.

Tempo de fim: Permite definir o tempo até ao qual o vídeo ou áudio serão reproduzidos.

Estes foram removidos de forma a obrigar a que o conteúdo seja cortado antes de ser enviado para o Backend, consumindo assim menos recursos quer do servidor quer da aplicação móvel.

O sistema deve ser implementado de forma que facilmente sejam adicionados novos parâmetros de customização. Alguns exemplos de parâmetros que poderiam ser implementados no futuro, seria a possibilidade de definir a posição e rotação da Ação em relação ao Trigger.

RF-15: Customização diferentes para a mesma Ação

Este requisito pretende que a mesma Ação quando associada a Triggers diferentes possa ter parâmetros de customização diferentes. Explicando através de um exemplo, temos 2 Triggers diferentes que possuem a mesma Ação associada:

Trigger_1: Logotipo da Critical Software

Ação_1: Vídeo da Parceria Critical&BMW (Propriedades: Manter Proporção = *true*)

Trigger_2: Logotipo da BMW

Ação_1: Vídeo da Parceria Critical&BMW (Propriedades: Manter Proporção = *false*)

Quando a Ação é despoletada a partir do Trigger_1 o vídeo deve manter a sua proporção original, enquanto que, quando é despoletada a partir do Trigger_2 não deve manter a proporção, adaptando-se ao tamanho do Trigger_2.

RF-18: Preparar Conteúdo

O Backend é responsável por processar tanto os Triggers como as Ações de forma que estas possam ser utilizadas pela aplicação móvel, para ambas as plataformas Android e iOS, sem haver a necessidade de uma nova compilação da aplicação.

Requisitos da Aplicação móvel:

Na tabela 4 são especificados os requisitos funcionais, da Aplicação móvel, e respetivas prioridades. Para uma melhor visualização verificar [apêndice B](#).

Tabela 4 - Tabela de requisitos funcionais, da aplicação móvel, e respetivas prioridades.

Requisitos Funcionais				
Componente	ID	Nome	Descrição	Prioridade
Aplicação Móvel	RF-19	Atualizar Campanhas	Aplicação deve ser capaz de obter as novas Campanhas ou atualizar existentes sem necessitar de uma nova <i>build</i>	Must
	RF-20	Reconhecimento de Triggers	Aplicação deve ser capaz de reconhecer Triggers através da imagem recolhida pela câmara do dispositivo	Must
	RF-21	Despoletar Ações	Ao reconhecer um Trigger a aplicação deve despoletar as respetivas Ações com os parâmetros de customização definidos.	Must
	RF-22	Tirar fotografia	Através da aplicação deve ser possível tirar fotografias que incluam o conteúdo de realidade aumentada gerado	Could
	RF-23	Gravar vídeo	Através da aplicação deve ser possível gravar vídeos que incluam o conteúdo de realidade aumentada gerado	Could

RF-19: Atualizar Campanhas

A aplicação deve ser capaz de atualizar as Campanhas que se encontram ativas através do Backend, fazendo download dos Triggers e Ações que estão a ser utilizados por estas, assim como das propriedades definidas para cada uma das Ações.

De forma a diminuir a quantidade/tempo deste download é necessário que seja implementado um sistema de *caching* na aplicação, para que não se faça o download de conteúdo que já esteja disponível na mesma.

RF-20: Reconhecimento de Triggers

A aplicação deve ser capaz de reconhecer os Triggers, pertencentes a Campanha ativas, através da imagem recolhida pela câmara do dispositivo

RF-21: Despoletar Ações

Ao reconhecer um Trigger a aplicação deve ser capaz de mostrar as Ações associadas a este, colocando-as sobrepostas à imagem (Trigger). As Ações devem respeitar os parâmetros de customização definidos.

Casos de uso

Os casos de uso permitem detalhar o funcionamento do sistema perante a execução das funcionalidades descritas nos requisitos. Desta forma é possível compreender como o sistema se deve comportar perante a interação com o utilizador ou entre as suas componentes, ajudando a perceber como os requisitos deverão ser implementados. A definição de casos de uso foi feita com recurso a uma tabela idêntica à seguinte:

Tabela 5 - Tabela modelo da definição de casos de uso utilizada.

Nome	Nome dado ao caso de uso
Âmbito	Descreve onde o utilizador se encontra e o que pretende fazer.
Atores	Pessoas ou Componentes do sistema envolventes neste caso de uso.
Estímulo	O que foi feito para chegar a este caso.
Pré-Condições	Condições necessárias para que o caso de uso possa acontecer.
Pós-Condições	Condições que se verificam após finalizar este caso de uso.
Caminho de sucesso	Caminho, etapas, para desempenhar este caso de uso.
Caminhos alternativos	Outros caminhos possíveis.
Exceções	Possíveis falhas, erros ou exceções que possam acontecer e como resolvê-las.

Os casos estão divididos em duas secções, uma para a aplicação móvel, que demonstra o comportamento da aplicação perante a interação com Utilizadores ou com outras componentes do sistema, e outra para a Interface de Gestão do Conteúdo que demonstra o comportamento da mesma perante interação com Gestores e o Backend.

Casos de uso: Aplicação móvel

Ao abrir a aplicação, é iniciada a fase de inicialização. Esta fase consiste em verificar se existem novas Campanhas ativas ou se é preciso atualizar existentes, fazendo o download do conteúdo necessário. Para além disso a aplicação deve remover os Triggers e Ações, de Campanhas anteriores, que deixaram de ser necessários, ou seja, que não façam parte de nenhuma das Campanhas ativas. Assim foi definido o seguinte caso de uso:

Tabela 6 - Caso de uso (app): Fase de inicialização (Atualizar Campanhas)

Nome	Fase de inicialização (Atualizar Campanhas) (RF-16, RF-19)
Âmbito	A aplicação está a ser inicializada e é necessário verificar se existem alterações nas Campanhas ativas.
Atores	Utilizador, Aplicação móvel e Backend.
Estímulo	Aplicação está a ser inicializada (fase de inicialização).
Pré-Condições	Aplicação estar instalada no dispositivo móvel. Aplicação tem acesso à internet. (Ex1 e Ex2)

Pós-Condições	Aplicação termina a fase de inicialização e passa para a fase de funcionamento normal.
Caminho de sucesso	<ol style="list-style-type: none"> 1. Aplicação carrega as Campanhas guardadas no dispositivo. 2. Aplicação faz pedido ao Backend para obter as Campanhas ativas. (Ex1) 3. Backend responde devolvendo todas as Campanhas que se encontram ativas. 4. Aplicação verifica se é necessário fazer o download de novos Triggers ou Ações 5. É necessário fazer download. 6. Aplicação dá escolha ao Utilizador para fazer download agora ou mais tarde. 7. Utilizador escolhe fazer download agora. 8. Aplicação faz Download dos Triggers e Ações necessários (que ainda não tem). (Ex2) 9. Aplicação remove todos os Triggers e Ações que deixaram de ser necessários. 10. Aplicação guarda as Campanhas no dispositivo 11. Aplicação mostra um botão para continuar para a fase seguinte. 12. Utilizador clica no botão. 13. Aplicação passa para a fase de funcionamento normal.
Caminhos alternativos	<p>4A: Não é necessário fazer download:</p> <ol style="list-style-type: none"> 1. Saltar para Caminho de sucesso (11)
Exceções	<p>Ex1: Aplicação não tem acesso à internet ou não consegue conectar com o Backend:</p> <ol style="list-style-type: none"> 1. Aplicação avisa o Utilizador que não conseguiu conectar 2. Aplicação dá duas opções: para continuar e usar as Campanhas guardadas no dispositivo (ir para Caminho de Sucesso (13)) ou tentar novamente. 3. Utilizador escolhe tentar novamente 4. Se a ligação com servidor for estabelecida ir para: Caminho de Sucesso (3), caso contrário voltar passo 1 <p>Ex2: Aplicação perde ligação ao Backend durante o download</p> <ol style="list-style-type: none"> 1. Aplicação mantém apenas os Triggers e Ações que conseguiu fazer download ou que já estavam presentes no dispositivo. 2. Aplicação avisa que perdeu a ligação. 3. Aplicação dá duas opções: para continuar e usar as Campanhas guardadas no dispositivo ou tentar novamente (ir para Caminho de Sucesso (8)) 4. Utilizador escolhe continuar 5. Aplicação passa para a fase de funcionamento normal usando apenas os Triggers e Ações disponíveis.

Passando agora para a fase de funcionamento normal, nesta fase a aplicação mostra ao utilizador a imagem recolhida pela câmara juntamente com as Ações dos Triggers que reconhece. Para esta fase é necessário que a aplicação tenha permissões para o uso da câmara. Desta forma, caso esta ainda não tenha essas permissões, surge o seguinte caso de uso:

Tabela 7 - Caso de uso (app): Verificação das permissões de uso da câmara.

Nome	Verificação das permissões de uso da câmara
Âmbito	A aplicação está a iniciar a fase de funcionamento normal, mas não tem permissões para o uso da câmara.
Atores	Utilizador e Aplicação móvel.
Estímulo	O Utilizador clicou no botão para continuar para a fase de funcionamento normal. (Início da fase de funcionamento normal)

Pré-Condições	Aplicação passou pela fase de inicialização Aplicação não tem permissões necessárias para utilizar a câmara do dispositivo
Pós-Condições	Aplicação continua na fase de funcionamento normal
Caminho de sucesso	1. Aplicação verifica se tem permissões para o uso da câmara. 2. Aplicação tem permissões.
Caminhos alternativos	2A: Aplicação não tem permissões de uso da câmara: 1. Aplicação pede permissões de uso da câmara. 2. Utilizador aceita permissões (ver Ex1).
Exceções	Ex1: Utilizador recusa as permissões: 1. Aplicação fica com ecrã preto.

Já com as permissões necessárias, a aplicação está pronta para reconhecer Triggers, através da imagem recolhida pela câmara, e despoletar as respetivas Ações da seguinte forma:

Tabela 8 - Caso de uso (app): Despoletar Ação

Nome	Despoletar Ação (RF-20, RF-21)
Âmbito	O Utilizador tem a aplicação aberta, inicializada e com as permissões necessárias, e pretende visualizar o conteúdo de realidade aumentada.
Atores	Utilizador e Aplicação móvel.
Estímulo	Aplicação reconhece um Trigger.
Pré-Condições	Aplicação aberta já com permissões da câmara. Aplicação inicializada. Aplicação encontra-se na fase de funcionamento normal.
Pós-Condições	Aplicação mostra o conteúdo de realidade aumentada de acordo com o Trigger.
Caminho de sucesso	1. Utilizador aponta a câmara para um Trigger. 2. Aplicação reconhece o Trigger. 3. Aplicação despoleta as Ações (com as características definidas) correspondentes a esse Trigger. As Ações devem aparecer sobrepostas ao Trigger e, caso este se movimente, a Ação deve seguir o mesmo.
Caminhos alternativos	1. Utilizador aponta a câmara para um ou mais Triggers em simultâneo. 2. Aplicação reconhece os Triggers. 3. Aplicação despoleta todas as Ações (com as características definidas) correspondentes a esse Trigger.
Exceções	

Durante a utilização da aplicação é ainda possível tirar fotografias ou gravar vídeos do conteúdo de realidade aumentada que está a ser mostrado. Seguem-se os casos de uso a descrever estas funcionalidades.

Tabela 9 - Caso de uso (app): Tirar Fotografia

Nome	Tirar Fotografia (RF-22)
Âmbito	O Utilizador está a ver o conteúdo de realidade aumentada e quer tirar uma fotografia.

Atores	Utilizador, Aplicação móvel.
Estímulo	Utilizador pressiona o botão de tirar fotografia.
Pré-Condições	Aplicação aberta e inicializada. Aplicação tem permissões para o uso da câmara.
Pós-Condições	Fotografia fica gravada na galeria, aplicação continua normalmente a mostrar o conteúdo.
Caminho de sucesso	<ol style="list-style-type: none"> 1. Utilizador clica no botão de tirar fotografia. 2. Aplicação tira fotografia que inclui o conteúdo de realidade aumentada. 3. Aplicação verifica se possui permissões necessárias para guardar na galeria. 4. Aplicação tem permissões necessárias. 5. Aplicação guarda foto na galeria. (ver Ex1)
Caminhos alternativos	<p>4A: Aplicação não tem permissões:</p> <ol style="list-style-type: none"> 1. Aplicação pede permissões ao Utilizador (opções: permitir, recusar, recusar e não perguntar novamente). 2. Utilizador escolhe a opção: permitir. 3. Aplicação guarda foto na galeria. <p>4B: Aplicação não tem permissões:</p> <ol style="list-style-type: none"> 1. Aplicação pede permissões ao Utilizador (opções: permitir, recusar, recusar e não perguntar novamente). 2. Utilizador escolhe a opção: recusar ou não perguntar novamente. 3. Aplicação avisa que a foto não foi guardada. <p>4C: Aplicação não tem permissões e não quer ser perguntado novamente:</p> <ol style="list-style-type: none"> 1. Aplicação avisa que necessita de permissões para guardar a foto. 2. Aplicação remete utilizador para as definições da aplicação. 3. Aplicação, quando retomada, continua normalmente.
Exceções	<p>Ex1: Falta de memória no dispositivo para guardar a foto:</p> <ol style="list-style-type: none"> 1. Aplicação avisa o utilizador. 2. Aplicação continua normalmente.

Tabela 10 - Caso de uso (app): Gravar vídeo

Nome	Gravar vídeo (RF-23)
Âmbito	O Utilizador está a ver o conteúdo de realidade aumentada e quer gravar um vídeo.
Atores	Utilizador, Aplicação móvel.
Estímulo	Utilizador pressiona o mesmo botão para tirar fotografias durante 2 ou mais segundos.
Pré-Condições	Aplicação aberta e inicializada. Aplicação tem permissões para o uso da câmara.
Pós-Condições	Vídeo fica gravado na galeria, aplicação continua normalmente a mostrar o conteúdo.
Caminho de sucesso	<ol style="list-style-type: none"> 1. Utilizador pressiona o mesmo botão para tirar fotografias durante 2 segundos. 2. Aplicação grava o conteúdo. 3. Utilizador pressiona o botão para parar de gravar. 4. Aplicação verifica se possui permissões necessárias para guardar na galeria. 5. Aplicação tem permissões necessárias. 6. Aplicação guarda o vídeo na galeria. (ver Ex1)
Caminhos alternativos	<p>4A: Aplicação não tem permissões:</p> <ol style="list-style-type: none"> 1. Aplicação pede permissões ao Utilizador (opções: permitir, recusar, recusar e não perguntar novamente). 2. Utilizador escolhe a opção: permitir.

	<p>3. Aplicação guarda vídeo na galeria.</p> <p>4B: Aplicação não tem permissões:</p> <ol style="list-style-type: none"> 1. Aplicação pede permissões ao Utilizador (opções: permitir, recusar, recusar e não perguntar novamente). 2. Utilizador escolhe a opção: recusar ou não perguntar novamente. 3. Aplicação avisa que o vídeo não foi gravado. <p>4C: Aplicação não tem permissões e não quer ser perguntado novamente:</p> <ol style="list-style-type: none"> 1. Aplicação avisa que necessita de permissões para guardar o vídeo. 2. Aplicação remete utilizador para as definições da aplicação. 3. Aplicação, quando retomada, continua normalmente.
Exceções	<p>Ex1: Falta de memória no dispositivo para guardar o vídeo:</p> <ol style="list-style-type: none"> 1. Aplicação avisa o utilizador. 2. Aplicação continua normalmente.

Casos de uso: Interface de Gestão do Conteúdo

Para criar uma Campanha é necessário um nome e definir o período de tempo durante o qual esta se encontrará ativa, surgindo assim o seguinte caso de uso:

Tabela 11 – Caso de Uso (Interface): Criar uma Campanha

Nome	Criar uma Campanha (RF-6)
Âmbito	Um Gestor pretende criar uma Campanha
Atores	Gestor, Interface de Gestão do Conteúdo, Backend
Estímulo	Gestor clica no botão para criar uma nova Campanha
Pré-Condições	Acesso à internet Nível de acesso de Gestor
Pós-Condições	É criada a Campanha à qual podem associar Triggers e Ações, sendo esta visível através da interface.
Caminho de sucesso	<ol style="list-style-type: none"> 1. Gestor pressiona o botão para criar Campanha 2. Interface mostra o formulário no qual o Gestor deve inserir um nome, dado à Campanha, uma data de início e uma data de fim (todos estes obrigatórios). 3. Gestor preenche os dados e clica no botão para submeter. 4. Interface envia o pedido para o Backend criar uma Campanha usando os dados fornecidos 5. Backend faz a validação desses dados, sendo a Campanha criada caso estes sejam validos. Backend responde com Sucesso. (ver Ex1) 3. Interface fecha o formulário, sendo a nova Campanha visível através da interface
Exceções	<p>Ex1: Dados inválidos:</p> <ol style="list-style-type: none"> 1. Backend devolve uma resposta que contém informação sobre os dados inválidos. 2. Interface indica ao utilizador quais dados não são válidos e o porquê. 2. Ir para Caminho de Sucesso (3)

O caso de uso para edição de uma Campanha (RF-7) será idêntico a este, com apenas algumas alterações, o utilizador deve selecionar qual a Campanha que pretende editar, o formulário já vem com os dados preenchidos de acordo com a Campanha selecionada, e o pedido para o Backend será para a edição da Campanha usando os novos dados.

Após a criação de Campanhas, para que se consiga associar Triggers e Ações a estas, é necessário adicionar/dar upload destes (Triggers e Ações) antecipadamente, de forma que sejam preparados para serem utilizados na aplicação móvel. Apenas após esta fase de preparação, e caso não ocorra nenhum erro, é que os Triggers e as Ações poderão ser associados a Campanhas. Surgindo assim os seguintes casos de uso:

Tabela 12 – Caso de Uso (Interface): Adicionar novo Trigger

Nome	Adicionar novo Trigger (RF-1)
Âmbito	Um Gestor pretende adicionar um novo Trigger para que este possa ser utilizado numa Campanha.
Atores	Gestor, Interface de Gestão do Conteúdo, Backend
Estímulo	Gestor clica no botão para adicionar um novo Trigger
Pré-Condições	Acesso à internet Nível de acesso de Gestor
Pós-Condições	O Trigger é adicionado, é visível através da interface, e pode ser associado a uma Campanha assim que estiver pronto para ser utilizado.
Caminho de sucesso	<ol style="list-style-type: none"> 1. Gestor pressiona o botão para adicionar um novo Trigger 2. Interface mostra o formulário no qual o Gestor deve inserir um nome, dado ao Trigger, e um ficheiro (imagem), ambos obrigatórios. 3. Gestor preenche o formulário e clica no botão para submeter. 4. Interface envia pedido para o Backend adicionar o novo Trigger usando os dados fornecidos 5. Backend faz a validação desses dados, sendo o Trigger adicionado caso estes sejam validos. Nesse caso o Backend responde com Sucesso. (ver Ex1) 6. Backend faz o processamento do ficheiro adicionado para que possa ser utilizado pela aplicação móvel. 7. Interface fecha o formulário, sendo o novo Trigger visível através da interface. No entanto este só poderá ser utilizado quando o Backend terminar o processamento do mesmo.
Exceções	Ex1: Dados inválidos: <ol style="list-style-type: none"> 1. Backend devolve uma resposta que contém informação sobre os dados inválidos. 2. Interface indica ao utilizador quais dados não são válidos e o porquê. 3. Ir para Caminho de Sucesso (3)

O Backend deve validar se a imagem que está a ser usada como Trigger já existe, devolvendo um erro caso isso se verifique.

Tabela 13 – Caso de Uso (Interface): Adicionar nova Ação

Nome	Adicionar nova Ação (RF-4)
Âmbito	Um Gestor pretende adicionar uma nova Ação para que esta possa ser utilizado numa Campanha.
Atores	Gestor, Interface de Gestão do Conteúdo, Backend
Estímulo	Gestor clica no botão para adicionar uma nova Ação

Pré-Condições	Acesso à internet Nível de acesso de Gestor
Pós-Condições	A Ação é adicionada, é visível através da interface, e pode ser associada a um Trigger existente numa Campanha assim que estiver pronto para ser utilizada.
Caminho de sucesso	<ol style="list-style-type: none"> 1. Gestor pressiona o botão para adicionar uma nova Ação 2. Interface mostra o formulário no qual o Gestor deve inserir um nome, dado à Ação, e um ficheiro (formatos especificados na secção de requisitos), ambos obrigatórios. 3. Gestor preenche o formulário e clica no botão para submeter. 4. Interface envia pedido para o Backend adicionar a nova Ação usando os dados fornecidos 5. Backend faz a validação desses dados, sendo a Ação adicionada caso estes sejam validos. Nesse caso o Backend responde com Sucesso. (ver Ex1) 6. Backend faz o processamento do ficheiro adicionado para que possa ser utilizado pela aplicação móvel. 7. Interface fecha o formulário, sendo a nova Ação visível através da interface. No entanto esta só poderá ser utilizada quando o Backend terminar o processamento da mesma.
Exceções	<p>Ex1: Dados inválidos:</p> <ol style="list-style-type: none"> 1. Backend devolve uma resposta que contém informação sobre os dados inválidos. 2. Interface indica ao utilizador quais dados não são válidos e o porquê. 3. Ir para Caminho de Sucesso (3)

Assim que o Backend terminar de processar um Trigger ou Ação adicionado, estes podem ser associados a Campanhas para que sejam utilizados pela Aplicação móvel. Surge assim os seguintes casos de uso:

Tabela 14 - Caso de uso (Interface): Associar Trigger a Campanha

Nome	Associar Trigger a Campanha (RF-9)
Âmbito	Um Gestor pretende associar um Trigger a uma Campanha
Atores	Gestor, Interface de Gestão do Conteúdo, Backend
Estímulo	Gestor clica num botão para associar um Trigger a uma Campanha
Pré-Condições	Acesso à internet Nível de acesso de Gestor Trigger está preparado para ser utilizado
Pós-Condições	A Campanha escolhida fica com o Trigger associado, sendo possível visualizar através da interface quais os Triggers que já foram associados a cada Campanha
Caminho de sucesso	<ol style="list-style-type: none"> 1. Gestor clica no botão para editar uma das Campanhas já criadas 2. Interface mostra um formulário no qual é possível selecionar Triggers (já preparados) 3. Gestor seleciona o Trigger que pretende adicionar 4. Gestor clica no botão para adicionar 5. Interface envia pedido para o Backend associar o Trigger escolhido à Campanha. Backend faz a validação do pedido e responde conformemente. 6. Interface dá feedback ao utilizador, mantendo o formulário aberto

Caminhos alternativos	<p>1A: Gestor clicar no botão para usar um dos Triggers já preparados</p> <ol style="list-style-type: none"> 1. Interface mostra um formulário no qual é possível selecionar uma das Campanhas já criadas 2. Gestor seleciona a Campanha à qual pretende associar o Trigger 3. Gestor clicar no botão para adicionar 4. Interface envia pedido para o Backend associar o Trigger e a Campanha escolhidos. Backend faz a validação do pedido e responde conformemente 5. Interface dá feedback ao utilizador e fecha o formulário
------------------------------	---

Tabela 15 - Caso de uso (Interface): Associar Ação a Trigger de uma Campanha

Nome	Associar Ação a Trigger de uma Campanha (RF-10)
Âmbito	Um Gestor pretende associar uma Ação a um Trigger pertence a determinada Campanha
Atores	Gestor, Interface de Gestão do Conteúdo, Backend
Estímulo	Gestor clica num botão para associar uma Ação a um Trigger
Pré-Condições	<p>Acesso à internet</p> <p>Nível de acesso de Gestor</p> <p>Ação está preparada para ser utilizada</p> <p>Trigger ao qual se pretende associar a Ação já pertence a uma Campanha</p>
Pós-Condições	O Trigger pertencente a determinada Campanha fica com a Ação associada, sendo possível visualizar através da interface as Ações associadas a cada Trigger de cada Campanha.
Caminho de sucesso	<ol style="list-style-type: none"> 1. Gestor clica no botão para editar uma das Campanhas já criadas (Página das Campanhas) 2. Interface mostra um formulário no qual é possível selecionar Triggers, já pertencentes a essa Campanha, e Ações (já preparadas) 3. Gestor seleciona o Trigger ao qual pretende associar uma Ação 4. Gestor seleciona a Ação que pretende associar ao Trigger selecionado 5. Gestor clica no botão para adicionar 6. Interface envia pedido para o Backend associar a Ação escolhida ao Trigger selecionado na Campanha escolhida. Backend faz a validação do pedido e responde conformemente. 7. Interface dá feedback ao utilizador, mantendo o formulário aberto
Caminhos alternativos	<p>1A: Gestor clicar no botão para usar uma das Ações já preparadas (Página da Ações)</p> <ol style="list-style-type: none"> 1. Interface mostra um formulário no qual é possível selecionar uma das Campanhas já criadas e um Trigger pertencente à Campanha selecionada 2. Gestor seleciona uma das Campanhas já criadas 3. Gestor seleciona o Trigger, pertencente a essa Campanha, ao qual pretende associar a Ação 4. Gestor clicar no botão para adicionar 5. Interface envia pedido para o Backend associar a Ação escolhida ao Trigger selecionado na Campanha escolhida. Backend faz a validação do pedido e responde conformemente 6. Interface dá feedback ao utilizador e fecha o formulário

3.4 Requisitos não-funcionais / Atributos de Qualidade

Os atributos de qualidade, muitas vezes reconhecidos como requisitos não-funcionais, especificam o modo como o sistema deve desempenhar os requisitos funcionais, influenciando a sua implementação e a arquitetura do sistema. Aqui, ao contrário dos requisitos funcionais, os atributos de qualidade não serão separados por componentes, uma vez que alguns deles afetam o sistema como um todo.

Availability (Disponibilidade)

Sabendo que o nosso sistema será apenas utilizado por funcionários da Critical Software, não se tratando de um sistema de risco ou de grande escala, não necessitamos de garantir elevados valores de *availability*. No entanto isso não significa que o sistema possa deixar de estar disponível durante grandes períodos. O ideal seria garantir que o sistema está a funcionar corretamente enquanto existirem Campanhas ativas. No entanto concluímos que, este não é um requisito que deva afetar ou aumentar a complexidade da arquitetura do nosso sistema, e caso ocorra algum problema este poderá ser corrigido de forma manual.

Performance (Desempenho)

Em relação à *performance*, este é um atributo importante que influenciou algumas limitações impostas tanto na Aplicação como no Backend. Parâmetros como a resolução, o tamanho, o *bitrate*, o nº de polígonos dos modelos 3D, entre outros, terão impacto:

- **No Backend:**
 - Tempo que demora a preparar/processar Trigger e Ações para que estes possam ser utilizados pela aplicação.
 - Espaço de disco utilizado pelo conteúdo, no servidor onde está alocada esta componente.
- **Na Aplicação móvel:**
 - Consumo de recursos do dispositivo (memória, espaço disco, CPU, GPU)

Uma vez que não temos acesso a uma grande variedade de dispositivos para testes, não conseguimos garantir que quaisquer requisitos de performance definidos serão uma realidade para todos os dispositivos compatíveis de executar a Aplicação móvel.

No entanto, foram implementadas determinadas limitações de forma a diminuir o impacto no desempenho do sistema:

- **Backend:**
 - Limitar o tamanho dos Triggers e Ações aceites
 - Limitar a qualidade durante a conversão de vídeos
- **Aplicação:**
 - Limitar a quantidade de clones (cópias das Ações) possíveis de se colocar

Deixamos assim ao encargo dos Gestores a escolha do conteúdo que pretendem utilizar, alertando-os para as possíveis consequências de utilizar conteúdo de elevada qualidade:

1. O Backend levará mais tempo a processar o conteúdo para que possa ser utilizado pela aplicação

2. Poderá não funcionar corretamente em alguns dispositivos móveis, especialmente nos mais antigos

No capítulo de Testes encontram-se alguns testes realizados com vários tipos de conteúdo, que exploram este *trade-off* entre desempenho e qualidade.

Em relação ao tempo de reconhecimento do Trigger, para além deste depender bastante da imagem utilizada, o algoritmo de reconhecimento utilizado foi o da ferramenta de desenvolvimento de AR escolhida, não fazendo sentido impor requisitos em algo sobre o qual não temos controlo, sabendo que não pretendemos implementar o nosso próprio algoritmo de reconhecimento de imagem.

Robustness (Robustez)

Caso não seja possível atualizar as Campanhas, por indisponibilidade do Backend ou por falta de acesso à internet, a aplicação deve utilizar as Campanhas guardadas da sessão anterior, caso existam.

Configurability (Configurabilidade)

Campanhas, Triggers e Ações devem ser configuráveis através da Interface de Gestão do Conteúdo.

Maintainability (Modificabilidade)

O sistema deve possibilitar a adição/alteração de propriedades de configuração das Ações mantendo um funcionamento normal. Para que estas alterações tenham efeito, pode ser necessária uma atualização da Aplicação, sendo que, caso esta não seja instalada o sistema deve funcionar normalmente descartando as novas propriedades.

Supportability

Em caso de erros, ou qualquer problema, estes devem ser expostos de forma clara, para que, no caso dos Gestores, consigam resolvê-los quando relacionados com a Interface de Gestão do Conteúdo/Backend ou, no caso dos Utilizadores, quando relacionados com a Aplicação móvel.

O sistema deve possuir medidas de *debug* e *logging* de forma que, caso existam problemas, estes consigam ser detetados e corrigidos.

Authentication e Authorization (Autenticação e Autorização)

O Backend deve garantir que apenas Gestores são capazes de tirar partido de algumas das funcionalidades descritas nos requisitos funcionais, sendo que só é possível garantir que as regras de *authorization* são cumpridas através da *authentication* dos utilizadores.

Capítulo 4

Planeamento

Neste capítulo, é apresentada a metodologia de trabalho adotada e são identificados alguns desvios que aconteceram face ao que estava planeado no início do estágio.

4.1 Metodologia de desenvolvimento

A metodologia de desenvolvimento adotada foi **Agile Kanban**. Agile é uma abordagem estruturada e iterativa de gestão e desenvolvimento de projetos, que reconhece a volatilidade do desenvolvimento dos mesmos, e fornece uma metodologia de organização e resposta a mudanças [71]. O projeto desenvolvido reúne algumas características que nos levaram a optar por uma abordagem deste tipo: o facto de querermos garantir um alto nível de liberdade na criação de experiências de realidade aumentada poderia levar a alterações ao longo do desenvolvimento (o que acabou mesmo por acontecer, por exemplo na adição de novas propriedades de customização das Ações). Para além disso a possibilidade de adaptar ou modificar o sistema com base no feedback obtido ao longo do desenvolvimento, foi outra característica que justifica a escolha de uma metodologia deste tipo.

Duas das metodologias Agile mais conhecidas são Scrum e Kanban, sendo que não são mutuamente exclusivas, podendo alguns dos seus aspetos ser adaptados ou agregados.

Scrum é uma *framework* baseada em ciclos de desenvolvimento, denominados *sprints*. Estas sprints duram um definido período de tempo, durante o qual devem ser concluídas um conjunto de tarefas selecionadas com base em estimativas. No final de cada sprint são avaliadas as tarefas desenvolvidas, recebendo feedback sobre estas e sendo possível a ocorrência de alterações (Em Scrum, normalmente, são feitas alterações apenas no final das *sprints* e nunca durante).

Kanban é uma *framework* baseada num fluxo e entrega contínuo. Ao contrário do Scrum não existem sprints, existe apenas um conjunto de tarefas definidas. Quando finalizada uma tarefa, esta é entregue e é iniciada uma nova, mantendo assim um fluxo. Este fluxo é alcançado através de um *Kanban board*. Normalmente um *Kanban board* é constituído por 3 colunas, uma que contém as tarefas a fazer (*To Do*), outra com as que estão em progresso (*In Progress*), e outra com as completadas (*Done*). Quando é utilizado Kanban a equipa define um valor limite (*WIP Limit – Work In Progress Limit*) que corresponde ao número máximo de tarefas que uma coluna pode ter. Assim, apenas são colocadas ou movidas tarefas entre colunas quando este limite permitir. Em relação à ocorrência de alterações, estas podem acontecer em qualquer altura, ou seja, em qualquer ponto do desenvolvimento pode ser adicionada uma nova tarefa ao *backlog* (conjunto de todas as tarefas necessárias para o projeto) ou adicionadas/alteradas/removidas tarefas existentes nas colunas.

Para este projeto optámos por Agile Kanban, por ser uma metodologia mais simples e com menor *overhead* em termos de documentação e processos necessários. Não achámos que havia necessidade de estar a definir sprints e a interromper o fluxo do trabalho, sabendo que se trata de uma equipa pequena (apenas o estagiário foi responsável pela implementação), sendo que,

Kanban promove um desenvolvimento mais livre e contínuo dando uma maior liberdade para fazer alterações, em qualquer momento, quer sejam de requisitos quer sejam das prioridades destes.

Um bom exemplo de feedback obtido que levou a alterações, foi a possibilidade de usar imagens ou vídeos com transparência como Ações, sendo que, estes deviam manter a transparência quando mostrados em Realidade Aumentada. Graças ao tipo de abordagem adotada (Agile Kanban) facilmente conseguimos adaptar o sistema de forma a possibilitar não só esta, mas também outras alterações ou melhorias.

As Tarefas relacionadas com novas funcionalidades foram descritas através de *user stories*. Uma *user story* é uma explicação informal de uma funcionalidade da perspetiva de um utilizador, cujo propósito é indicar como a funcionalidade irá fornecer valor ao cliente. A descrição é estruturada da seguinte forma: “Como um [tipo de utilizador], eu quero ser capaz de [descrever o que o utilizador quer fazer], [com que objetivo] “. Para além desta descrição, uma *user story* é composta por um grau de prioridade, critério de aceitação, testes realizados e qualquer outra informação considerada útil.

Quanto ao *Kanban Board* utilizado, este era constituído por 3 colunas:

- **To Do:** contém as *user stories* a fazer. Estas *user stories* eram criadas através de tarefas provenientes do backlog, após estarem bem definidas e especificadas. O *backlog* continha, para além de tarefas provenientes dos requisitos, ideias ou melhorias com base no feedback recebido ao longo do desenvolvimento.
- **In Progress:** contém as *user stories* que estão a ser feitas, em progresso. Apenas após o cumprimento do critério de aceitação, especificado na *story*, esta passa para a próxima coluna.
- **Done:** contém as *user stories* que foram completadas. Quando colocada nesta coluna, o trabalho realizado para a *user story* fica sujeito a um *code review*.

Em relação ao *WIP limit*, acabamos por não impor este limite, de forma a ter a liberdade de saltar entre tarefas ao longo de desenvolvimento de acordo com as necessidades.

Por último, as reuniões sofreram algumas alterações daquilo que estava planeado originalmente:

- **Reuniões semanais** acabaram por se tornar quase diárias. Às terças e quartas fez-se uma reunião de 10-15 min para mostrar o que está a ser feito e receber feedback (este feedback poderia provocar alterações ao Kanban Board diretamente ou ser adicionado ao *backlog*). Nos restantes dias da semana (segunda, quarta e sexta) foram reuniões de 15-20 min para discutir aspetos técnicos, tirar dúvidas técnicas ou fazer a análise do código, em caso de necessidade.
- **Reuniões extraordinárias** mantiveram-se e foram marcadas com base nas necessidades

4.2 Planeamento

Na sua maioria, a estimação/planeamento do estágio, feita no 1º Semestre, foi precisa, no entanto, existiram algumas exceções que valem a pena salientar.

A primeira seria a adição de novas tarefas com base no feedback dado na defesa intermédia. Para além de correções feitas ao relatório, foi também feita uma análise de riscos com base nos requisitos, no estudo do estado da arte e no planeamento.

Outra exceção foi o desenvolvimento, que não foi tão contínuo como se esperava, acabando por haver vários saltos entre o desenvolvimento da Aplicação móvel e o Backend à medida que foi sendo necessário. Desta forma no novo diagrama optei por dividir o desenvolvimento, de cada uma destas componentes, por partes, de maneira a representar estas situações.

Por último foi ainda adicionada a tarefa “Deploy” que consistiu em colocar o Backend a funcionar no servidor disponibilizado pelo serviço Azure. Esta tarefa não estava presente no diagrama de Gantt de estimativas, no entanto, acabou por demorar mais do que se esperava, devido a vários imprevistos técnicos que tiveram de ser resolvidos.

Seguem-se ambos os diagramas de Gantt, o primeiro que contém informação sobre o trabalho realizado no 1º semestre e com a estimação/planificação do segundo semestre. O segundo com o que realmente aconteceu. De salientar que algumas das componentes, como é o caso da análise de requisitos, o relatório de estágio final ou a especificação da API, poderão ter sofrido alterações ao longo de todo o desenvolvimento, no entanto não foram representadas no diagrama uma vez que o tempo gasto foi pouco significativo.

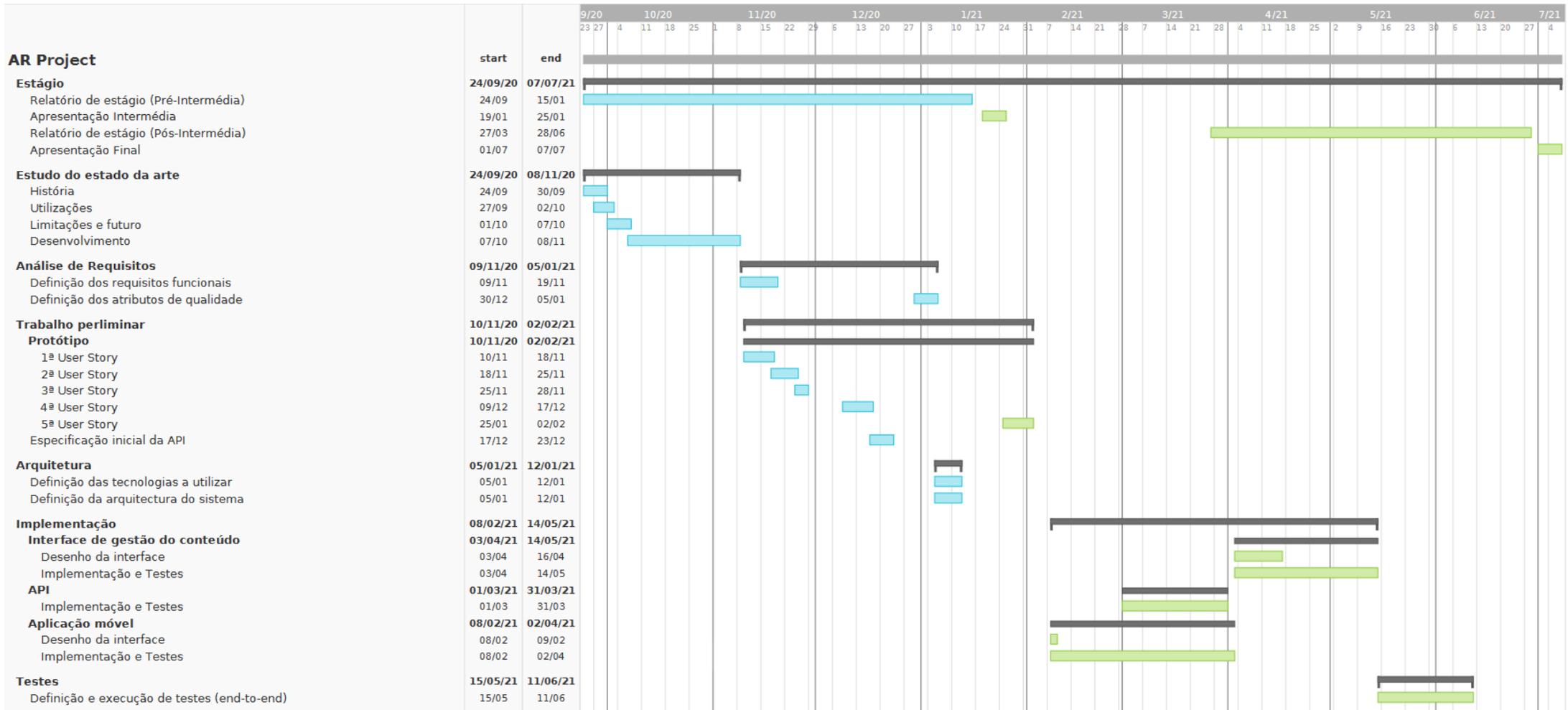


Figura 50: Diagrama de Gantt do trabalho realizado e planeado (1º Semestre).

- Duração de tarefas terminadas.
- Planeamento/Estimativa de tarefas por completar.

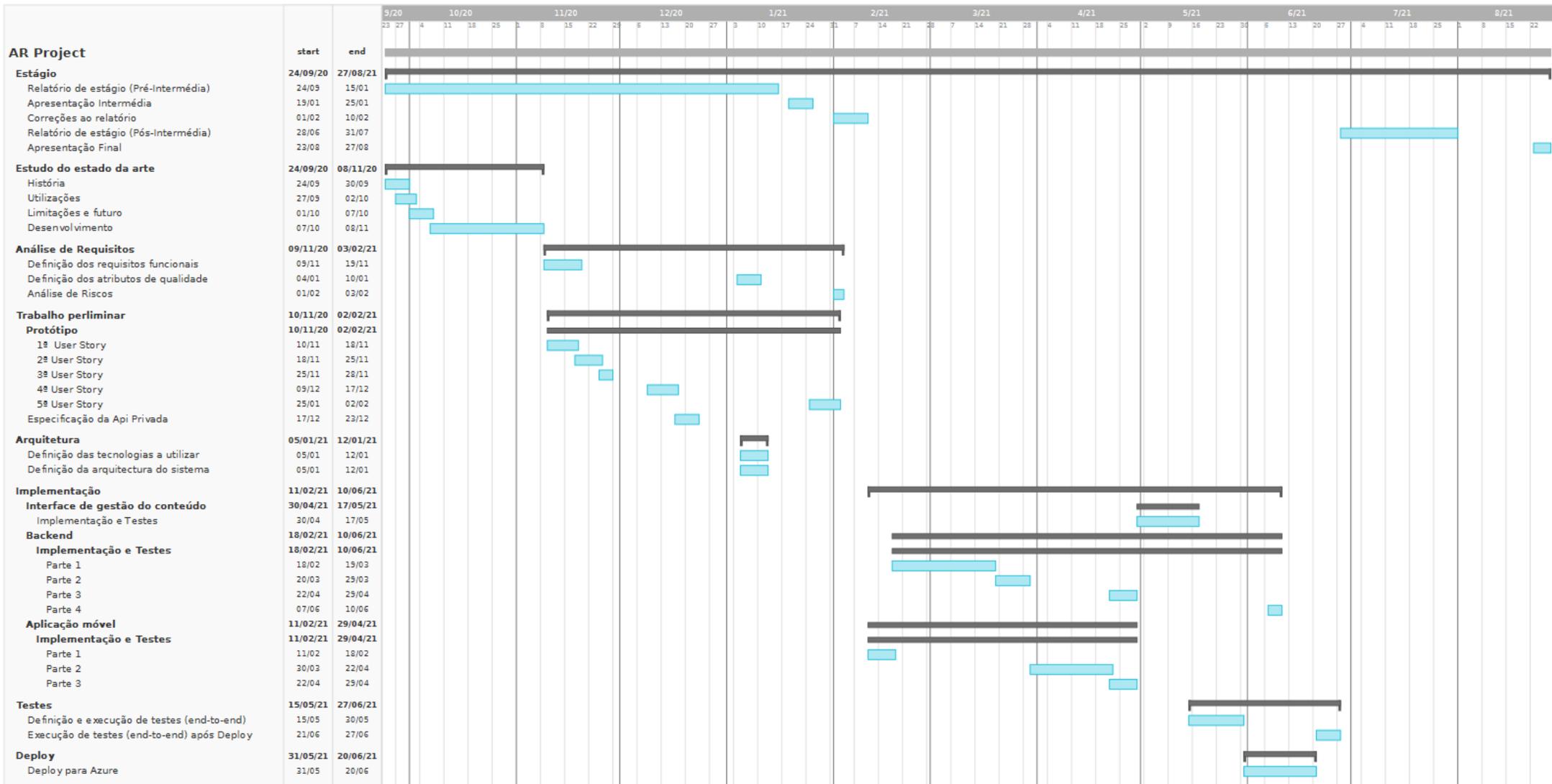


Figura 51 - Diagrama de Gantt do trabalho realizado (2º Semestre).

Capítulo 5

Análise de Riscos

Inerente ao desenvolvimento de qualquer projeto existem riscos que se poderão manifestar no decorrer do mesmo, tendo impacto no produto ou solução final. De forma a diminuir este impacto, ou até mesmo mitigá-lo por completo, é necessária uma análise dos possíveis riscos que poderão ocorrer, identificando-os e definindo como deve ser feita a gestão dos mesmos. Para isso, para cada um dos riscos identificados foram especificados os seguintes fatores:

- **Probabilidade:** a probabilidade de ocorrência do risco (Baixa: < 30%, Média: 30% _ 60%, Alta: 60% _ 80%, Muito Alta: ≥ 80%).
- **Severidade:** o grau de severidade / impacto da ocorrência do risco.
- **Consequências:** As consequências inerentes à ocorrência do risco.
- **Técnicas/Medidas para diminuir a probabilidade:** processos aplicados para diminuir a probabilidade de ocorrência do risco.
- **Técnicas/Medidas para diminuir a severidade:** processos aplicados para diminuir o impacto do risco sobre os objetivos do projeto.
- **Técnicas/Medidas para monitorizar:** processos utilizados para monitorizar e controlar o risco.

Os riscos identificados foram ainda organizados pelas seguintes categorias:

- **Riscos Técnicos:** riscos associados aos requisitos do projeto e às soluções propostas.
- **Riscos de Recursos:** riscos associados às dependências do projeto, recursos e orçamentos.
- **Riscos da Gestão do Projeto:** riscos associados à gestão do projeto, como: planificações, estimativas, comunicação/reuniões.

Segue-se uma tabela com os riscos identificados para o presente projeto, assim como as medidas de gestão definidas para os mesmos. Para uma melhor visualização verificar [apêndice C](#).

Tabela 16 - Análise de Riscos

		Análise de Riscos						
Categoria	ID	Risco	Probabilidade	Severidade	Consequências	Para diminuir a probabilidade	Para diminuir a severidade	Para monitorizar
Riscos Técnicos	1	Alterações de requisitos numa fase avançada do projeto	Baixa	Média	Poderá exigir alterações sobre funcionalidades que já foram implementadas, podendo atrasar o projeto, ou não existir tempo para aplicar estas alterações.	Realizar as reuniões planeadas, mantendo assim uma comunicação constante do que está a ser desenvolvido. Adotar uma metodologia de desenvolvimento adequada face à probabilidade de ocorrência de alterações de requisitos.	Fazer a análise das implicações impostas por estas alterações e possíveis conflitos que possam criar. Tendo em conta esta análise, fazer as alterações necessárias, caso factíveis, ou abandonar os requisitos com menor prioridade se necessário.	Manter uma constante comunicação com os orientadores sobre as tarefas que estão a ser desempenhadas.
	2	Solução proposta não permite cumprir com todos os requisitos ou possíveis alterações dos mesmos	Baixa	Muito-Alta	Não cumprimento de todos os objetivos / requisitos propostos para o sistema.	Fazer um bom estudo e planeamento do sistema tendo em conta possíveis oportunidades futuras. Desenvolver um sistema que tenha em conta estas oportunidades.	Exige reavaliar a arquitetura proposta para o sistema face aos requisitos existentes, e fazer as alterações necessárias caso sejam possíveis, tendo em conta o que já foi feito e o orçamento de tempo disponível.	Manter uma constante comunicação com os orientadores, não só sobre as tarefas que estão a ser desempenhadas, mas também possíveis oportunidades futuras que possam ser aplicadas e devem ser tidas em conta no desenho da arquitetura.
	3	Conflitos entre requisitos	Média	Média	Podem gerar quebras no desenvolvimento, perdendo tempo a resolver estes conflitos.	Descrever os requisitos de uma forma concisa e clara para que não existam dúvidas ao longo do desenvolvimento. Manter uma comunicação constante com os orientadores.	Fazer uma análise e reestruturação destes requisitos tendo em conta os conflitos encontrados.	Manter uma constante comunicação com os orientadores sobre os requisitos e as tarefas / user stories criadas a partir destes.
	4	Fraca ou incompleta documentação	Baixa	Média	Dificuldade de perceber o trabalho que foi feito, assim como as decisões tomadas ao longo do desenvolvimento.	Juntamente com os code reviews existir também uma análise da documentação produzida para cada tarefa/user story.	Adicionar uma nova tarefa / user story com o objetivo de complementar a documentação necessária.	Juntamente com os code reviews, existir também uma análise da documentação produzida.
	5	Recursos/Servidores disponibilizados para o Backend não são suficientes	Baixa	Média	Incumprimento dos atributos de qualidade definidos.	Planear a arquitetura do sistema tendo em conta o âmbito no qual o sistema será utilizado (apenas por funcionários da Critical +/- 1000)	Reestruturar o sistema para que existam várias instâncias das componentes do Backend. Aumentar os recursos da máquina utilizada.	Utilizar as ferramentas de monitorização providenciadas pelo serviço Microsoft Azure.
	6	Dependência de sistemas externos	Baixa	Muito-Alta	Na ocorrência de problemas ou alterações nestes sistemas externos, a nossa solução poderá ser afetada exigindo alterações de acordo com o sucedido		Diminuir ao máximo as dependências aos sistemas externos que estão a ser utilizados	Monitorizar possíveis alterações nestes sistemas externos.

Riscos de Recursos	7	Falta de acesso a outros dispositivos / smartphones para testes	Média	Alta	Impossibilidade de testar a aplicação em várias plataformas e hardware. Tratando-se de uma aplicação de realidade aumentada, requer o uso da câmara, não sendo possível testar com emuladores.	Constante comunicação com os orientadores alertando para a necessidade destes dispositivos.	Tirar partido dos dispositivos de familiares e amigos para realizar alguns testes.	Comunicação com os orientadores
	8	Falta de acesso a um Mac	Média	Muito-Alta	Impossibilidade de compilar a aplicação para plataformas IOS, uma vez que é necessário acesso ao programa Xcode	Constante comunicação com os orientadores alertando para a necessidade de um dispositivo que suporte o Xcode.		Comunicação com os orientadores
Riscos da Gestão do Projeto	9	Impossibilidade de realizar reuniões de gestão do Kanban Board (terças e quartas)	Baixa	Baixa	Não ocorre a revisão do Kanban Board juntamente com os orientadores da Critical, podendo não existir tarefas a desempenhar bem definidas e provocando atrasos. Pode ainda provocar o risco 1.	Constante comunicação com os orientadores	Caso não existam tarefas planeadas, terá de ser realizada uma reunião o mais cedo possível. O estagiário poderá também definir sozinho as tarefas/user stories que irá desempenhar, a partir dos requisitos.	Comunicação com os orientadores
	10	Erros de estimação e possíveis atrasos no desenvolvimento	Média	Alta	Poderá levar à propagação de atrasos e a um possível incumprimento da entrega final (Risco - 12) ou abandono de requisitos com uma menor prioridade.	Fazer estimações e planeamento juntamente com os orientadores.	Será necessária a reformulação das estimativas de acordo com estes atrasos e reestruturar o planeamento adequadamente.	Acompanhar o planeamento/estimativas de forma a verificar possíveis atrasos tendo em conta o backlog de tarefas por realizar.
	11	Dependência entre tarefas / user stories	Média	Baixa	Interrompe o fluxo de trabalho podendo criar atrasos. Exige colocar temporariamente uma tarefa em pausa enquanto se terminam outras necessárias.	Definir quais as próximas tarefas a desempenhar tendo em conta possíveis dependências entre as componentes do sistema.	Adaptar Kanban Board de acordo com as necessidades.	Tirar partido das funcionalidades disponibilizadas pela ferramenta Jira e o seu Kanban Board.
	12	Não cumprir com a entrega final do estágio estipulada	Média	Média	Pressão, sobre o estagiário. Incumprimento de determinados requisitos ou objetivos do sistema.	Cumprir com o planeamento definido, assim como, com os processos associados à metodologia de desenvolvimento escolhida.	Recorrer à época especial.	Acompanhar as estimativas de forma a verificar possíveis atrasos tendo em conta o backlog de tarefas por realizar.
	13	Surgimento de novas tarefas não planeadas	Média	Média	Poderá criar atrasos no projeto.	Constante comunicação com os orientadores	Exige reestruturar o planeamento adequadamente.	Comunicação com os orientadores

Capítulo 6

Trabalho Preliminar

Como trabalho preliminar foram desenvolvidos, uma aplicação protótipo de realidade aumentada e uma especificação inicial da API, sendo estes detalhados ao longo deste capítulo.

6.1 Protótipo

Apesar do estudo do estado da arte ser uma fase importante de qualquer projeto, por vezes não é o suficiente para perceber todas as capacidades e limitações de cada ferramenta. Assim optámos por realizar um pequeno protótipo, não só para confirmar se as ferramentas escolhidas (Unity + AR Foundation) seriam as ideais para realizar os requisitos propostos, mas também para começar a adquirir conhecimento sobre estas. Para além disso, o trabalho realizado, permitiu a familiarização com as ferramentas (BitBucket e Jira) e os processos utilizados pela Critical para o desenvolvimento de software.

O protótipo consiste numa aplicação simples que ao reconhecer uma imagem (Trigger) mostra um vídeo ou um objeto/modelo 3D sobreposto a esta. Sendo que, para além de explorar o funcionamento tanto do Unity como do AR Foundation, existiam algumas capacidades fundamentais de perceber se eram ou não factíveis, e as suas possíveis limitações:

- A capacidade/possibilidade de uma implementação flexível, ou seja, ser possível alterar os Triggers e respetivas Ações durante *runtime*, não sendo necessária uma nova compilação e atualização da aplicação.
 - No caso dos Triggers, através da documentação do AR Foundation, já sabíamos que é possível a alteração destes durante *runtime*.
 - No caso das Ações, existia alguma incerteza principalmente sobre modelos 3D, que normalmente requerem uma nova compilação da aplicação.
- Perceber que parâmetros de customização poderíamos oferecer de forma a personalizar cada tipo de Ação.

Para o desenvolvimento deste protótipo foi adotada uma metodologia Kanban idêntica à que foi seguida no desenvolvimento do projeto deste estágio. Em relação às ferramentas de apoio ao desenvolvimento utilizadas, através do JIRA era definido o *Kanban Board* com *user stories* e respetivos requisitos a cumprir, e o BitBucket foi utilizado como repositório partilhado da documentação e da implementação servindo também para controlo de versões.

De seguida verificam-se alguns detalhes sobre cada *user story*, incluindo a descrição, critério de aceitação, requisitos, e objetivos de cada uma.

1ª *User Story*

Descrição: Como utilizador da aplicação, quero ser capaz de ver um vídeo quando aponto a câmara do meu dispositivo para o logotipo da Critical Software.

Critério de aceitação: Dado um logótipo da Critical Software, consigo ver um vídeo a ser reproduzido em realidade aumentada.

Requisitos:

- RF-1: Aplicação deve ser capaz de reconhecer o logotipo da Critical Software.
- RF-2: Ao reconhecer o logotipo da Critical Software a aplicação deve despoletar uma Ação, sendo esta ação um vídeo a sobrepor a imagem.

Esta primeira *story* serviu principalmente como uma introdução ao Unity, de forma a perceber como funciona a *Engine* no geral e aprender conceitos básicos como *prefabs*, *game objects*, *scripts* etc... Serviu também para introduzir funcionalidades e conceitos básicos do AR Foundation, como *AR Session*, *ARSession Origin*, *Reference Images Library* e outras que foram úteis ao longo do projeto.

2ª User Story

Descrição: Como utilizador da aplicação quero poder ter botões de redes sociais que devem ser clicáveis, e ao clicar neles devo ser direcionado para as páginas das redes sociais da Critical Software.

Critério de aceitação: Consigo clicar em todos os botões das redes sociais, e todos eles direcionam-me para o endereço correto.

Requisitos:

- RF-3: Associado ao vídeo deve existir um painel de redes sociais. Este painel deve conter botões que remetem o utilizador para as redes sociais da Critical Software (Facebook e Instagram).
- RF-4: O painel referido em RF-3 deve ser uma componente opcional, ou seja, a decisão de ter ou não o painel deve ser customizável

3ª User Story

Descrição: Como utilizador da aplicação quero poder ter os botões de controlo de vídeo, que devem ser clicáveis, e ao clicar neles consiga parar, reproduzir ou silenciar o vídeo que está a ser mostrado.

Critério de aceitação: Consigo clicar em todos os botões, e todos eles fazem as operações pretendidas.

Requisitos:

- RF-5: Associado ao vídeo deve existir um painel de controlo, através do qual deve ser possível silenciar e pausar/continuar a reprodução do mesmo.
- RF-6: O painel referido em RF-5 deve ser uma componente opcional, ou seja, a decisão de ter ou não o painel deve ser customizável

Através da 2ª e 3ª *story* pretendíamos verificar não só se a customização das Ações seria possível, mas também explorar que tipo de parâmetros poderão ser customizáveis no futuro.

4ª User Story

Descrição: Como utilizador da aplicação ao apontar a câmara para uma imagem quero ser capaz de visualizar um qualquer modelo 3D a desempenhar uma animação.

Critério de aceitação: Conseguir visualizar o modelo e respetiva animação ao apontar a câmara para a imagem escolhida.

Requisitos:

- RF-7: Ao reconhecer a imagem escolhida, a aplicação deve colocar sobre esta um modelo 3D a desempenhar uma qualquer animação.

Esta *story* teve o intuito de explorar a implementação e o trabalho com modelos 3D e animações, perceber como funcionam, as suas propriedades, formatos e outros conceitos que poderiam vir a ser necessários ao longo do projeto.

5ª User Story

Requisitos:

- AQ-1: O modelo 3D utilizado em RF-5 deve ser obtido durante *runtime*, em vez de ser compilado com a aplicação (ou seja, não deve ir incluído na *build* da aplicação).

Esta última *story* do protótipo, não se trata de uma funcionalidade, apenas pretendíamos perceber quais as soluções possíveis para alcançar uma implementação flexível.

6.2 Especificação da API

Ainda na fase de trabalho preliminar, foi feita uma especificação inicial da API presente na componente Backend do sistema. Esta especificação foi sendo adaptada ou atualizada ao longo do projeto à medida que iam surgindo novos requisitos ou perante a alteração de existentes. Para criar esta documentação utilizámos a especificação OpenAPI 2.0 recorrendo ao Swagger e algumas das suas ferramentas, nomeadamente à Swagger Preview.

OpenAPI corresponde a um conjunto de regras (especificação) para descrever uma *Web API*, facilitando a criação de documentação que permita a um humano rapidamente perceber os métodos que API disponibiliza, respetivos pedidos e respostas.

Swagger é um conjunto de ferramentas para auxiliar na construção, desenho e documentação de APIs utilizando a especificação OpenAPI.

Segue-se a especificação final da API (Api Privada do Capítulo 7: Arquitetura Implementada), que disponibiliza informação sobre os pedidos aceites pela *Web API* e possíveis respostas a estes. A especificação foi dividida em 4 categorias:

- **Trigger:** Agrega operações relacionadas com os Triggers

Trigger		Operations related with Triggers	^
GET	/Triggers	Get a list with all the Triggers available in the server	∨
POST	/Triggers	Add/Upload a new Trigger	∨
GET	/Triggers/{triggerId}	Get a specific Trigger by id	∨
DELETE	/Triggers/{triggerId}	Remove a Trigger by id	∨

Figura 52 - - Swagger Preview das operações definidas para os Triggers.

- **Action:** Agrega operações relacionadas com as Ações

Action		Operations related with Actions	^
GET	/Actions	Get a list with all the Actions available in the server	∨
POST	/Actions	Add/Upload a new Action	∨
GET	/Actions/{actionId}	Get a specific Action by id	∨
DELETE	/Actions/{actionId}	Remove an Action by id	∨

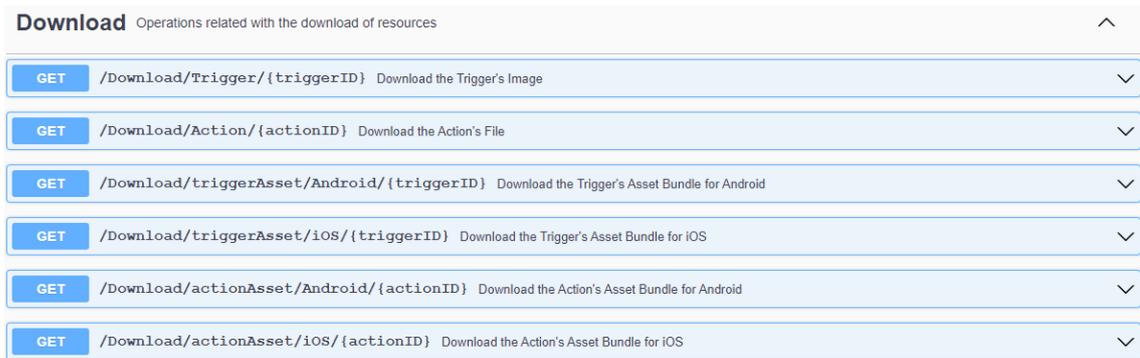
Figura 53 - Swagger Preview das operações definidas para as Ações.

- **Campaign:** Agrega operações relacionadas com as Campanhas.

Campaign		Operations related with Campaigns	^
GET	/Campaigns	Get a list with all the Campaigns created	∨
POST	/Campaigns	Create a new Campaign	∨
GET	/Campaigns/Active	Get a list with the currently active Campaigns	∨
GET	/Campaigns/{campaignID}	Get a specific Campaign by id	∨
DELETE	/Campaigns/{campaignID}	Delete a Campaign by id	∨
PATCH	/Campaigns/{campaignID}	Update a Campaign	∨
GET	/Campaigns/{campaignID}/Triggers	Get the list of Triggers being used by a specific Campaign	∨
POST	/Campaigns/{campaignID}/Triggers	Add a Trigger to a Campaign	∨
DELETE	/Campaigns/{campaignID}/Triggers	Remove Trigger from a Campaign	∨
GET	/Campaigns/{campaignID}/Triggers/{triggerID}	Get a Trigger from Campaign	∨
GET	/Campaigns/{campaignID}/Triggers/{triggerID}/Actions	Get the list of Actions of a Trigger	∨
POST	/Campaigns/{campaignID}/Triggers/{triggerID}/Actions	Add a Action to a Trigger of a specific Campaign	∨
DELETE	/Campaigns/{campaignID}/Triggers/{triggerID}/Actions	Remove Action from a Trigger of a Campaign	∨
GET	/Campaigns/{campaignID}/Triggers/{triggerID}/Actions/{actionID}	Get a Action added to a Trigger of a specific Campaign	∨
POST	/Campaigns/{campaignID}/Triggers/{triggerID}/Actions/{actionID}/Properties	Change/Update the properties of an Action	∨

Figura 54 - Swagger Preview das operações definidas para as Campanhas.

- **Download:** Agrega operações relacionadas com o download de recursos.

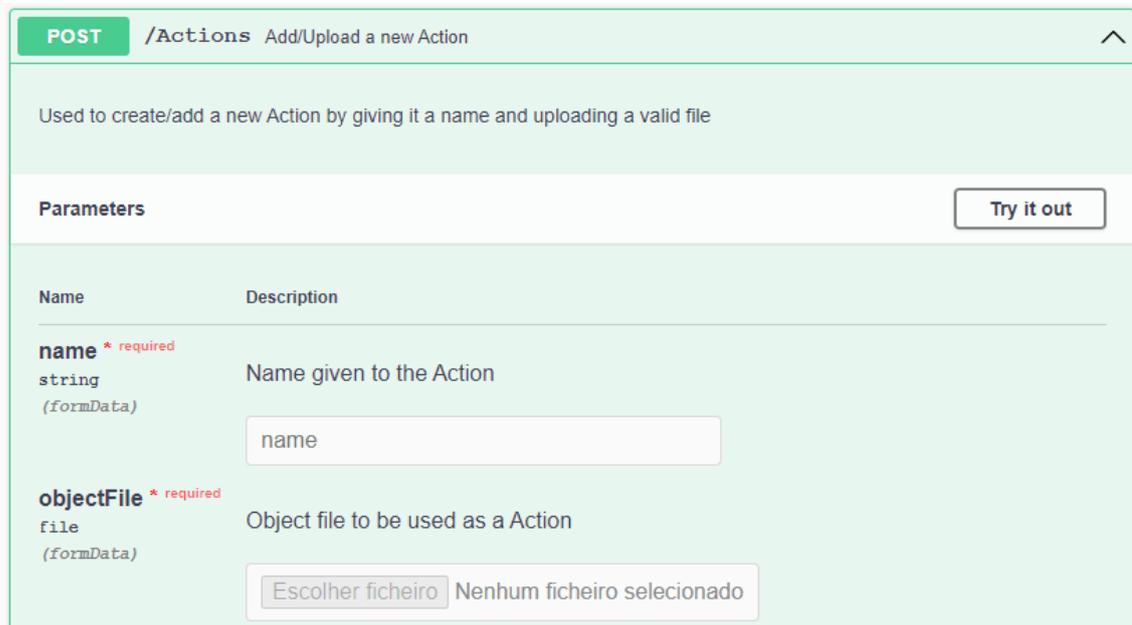


The image shows a Swagger UI section titled "Download" with the subtitle "Operations related with the download of resources". It contains six GET endpoints, each with a description and a dropdown arrow on the right:

- GET /Download/Trigger/{triggerID} Download the Trigger's Image
- GET /Download/Action/{actionID} Download the Action's File
- GET /Download/triggerAsset/Android/{triggerID} Download the Trigger's Asset Bundle for Android
- GET /Download/triggerAsset/iOS/{triggerID} Download the Trigger's Asset Bundle for iOS
- GET /Download/actionAsset/Android/{actionID} Download the Action's Asset Bundle for Android
- GET /Download/actionAsset/iOS/{actionID} Download the Action's Asset Bundle for iOS

Figura 55 - Swagger Preview das operações relacionadas com o download de recursos.

A figura seguinte é um exemplo de descrição de um pedido e das possíveis respostas a este, neste caso para a adição de uma nova Ação.



The image shows a Swagger UI section for a POST endpoint titled "POST /Actions" with the subtitle "Add/Upload a new Action". The description is "Used to create/add a new Action by giving it a name and uploading a valid file". There is a "Try it out" button. The parameters section is as follows:

Name	Description
name * required string (formData)	Name given to the Action <input type="text" value="name"/>
objectFile * required file (formData)	Object file to be used as a Action <input type="button" value="Escolher ficheiro"/> Nenhum ficheiro selecionado

Responses		Response content type
		application/json
Code	Description	
201	Resource created response. Action was added successfully. The response contains the new Action	
	Example Value Model	
	<pre> { "id": 0, "name": "string", "actionType": "string", "fileName": "string", "fileExtension": "string", "assetBundleStatus": 0, "properties": [{ "name": "string", "value": "string" }] } </pre>	
400	Bad Request. There is some problem with the request made by the client. Response should contain information about what's wrong with the request	
	Example Value Model	
	<pre> { "errors": [{ "name": "string", "message": "string" }], "type": "string", "title": "string", "status": "string", "traceId": "string" } </pre>	
500	Internal Server error. Something went wrong (server-side) with the request	

Figura 56 - Exemplo de especificação de um pedido à API presente no Backend

Capítulo 7

Arquitetura Implementada

Neste capítulo é explorada a solução arquitetural adotada que, não só define a estrutura do sistema, mas também especifica as tecnologias que foram utilizadas em cada uma das suas componentes.

O sistema é composto por 3 grandes componentes. Na figura seguinte encontra-se uma versão de alto nível da arquitetura do sistema, demonstrando as interações entre estas.

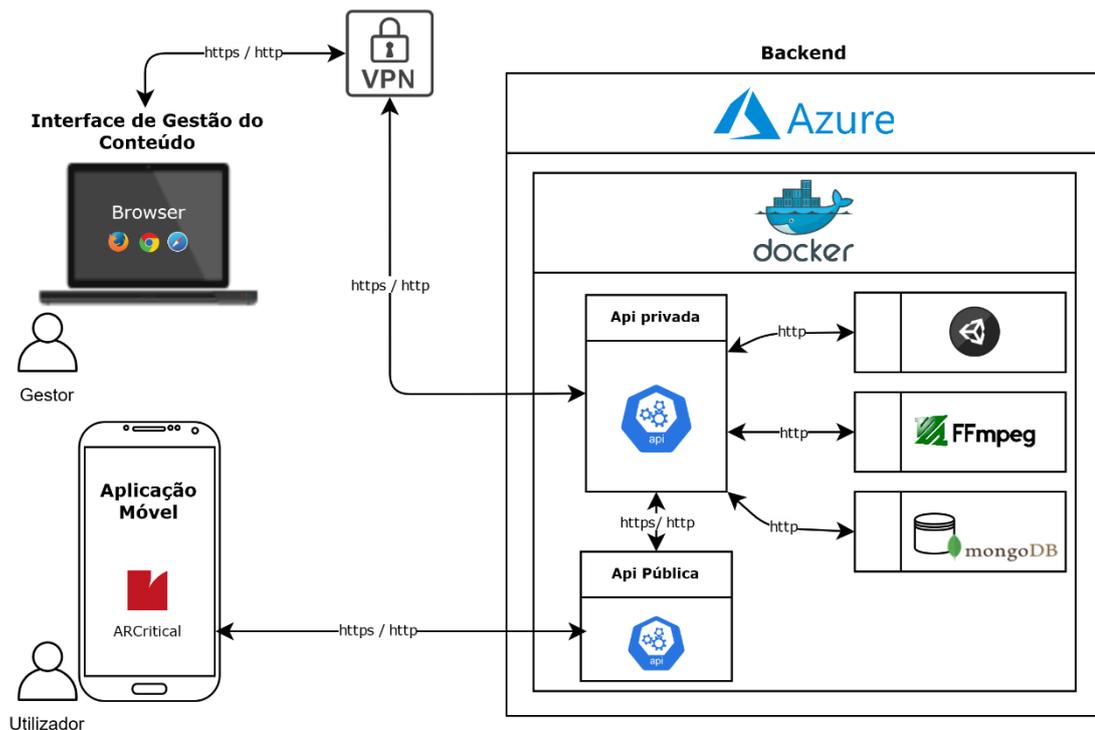


Figura 57 - Arquitetura de alto nível do sistema.

Backend: Esta componente é o coração de todo o sistema sendo a responsável por garantir a criação, customização e modificabilidade das diferentes Campanhas que os Gestores decidirem organizar. Através desta componente conseguimos fornecer a liberdade necessária para criar as experiências de realidade aumentada desejadas, sem a necessidade de atualizar a aplicação com uma nova *build*.

Interface de gestão do conteúdo: Sabendo que a equipa responsável (Gestores) poderão não ter os conhecimentos necessários para lidar com o Backend diretamente, foi criada uma interface de forma a facilitar esta interação, permitindo a Gestores fazer as alterações que desejarem. É através desta interface que qualquer pessoa, desde que tenha permissões de acesso de Gestor, pode criar experiências de realidade aumentada sem possuir qualquer conhecimento sobre a tecnologia ou do próprio sistema.

Aplicação móvel: A aplicação é responsável por fornecer as experiências de realidade aumentada definidas. Assim, esta deve ser capaz de fazer os pedidos para o Backend de forma

a atualizar as Campanhas ativas, e o respetivo conteúdo destas. Esta atualização tem de ser possível sem a necessidade de uma nova compilação (*build*) da aplicação. Para além disso é necessário que a aplicação esteja construída de forma a permitir customizar as Ações despoletadas. O reconhecimento de Triggers através da imagem recolhida pela câmara e a visualização das respetivas Ações é a funcionalidade principal desta componente.

As várias componentes comunicam entre si através de pedidos http ou https:

- **Aplicação móvel <-> Backend** de forma a obter novas campanhas ou atualizar existentes.
- **Interface de gestão <-> Backend** de forma a criar, editar ou remover Campanhas, adicionar novos Trigger e/ou Ações.

De seguida são exploradas cada uma das secções com maior detalhe.

7.1 Aplicação móvel

Começando pela Aplicação móvel, na seguinte secção explicamos a escolha da ferramenta de desenvolvimento de realidade aumentada para a nossa aplicação, e são detalhadas as fases de funcionamento pelas quais esta é composta.

Ferramenta de desenvolvimento

Uma das primeiras decisões a tomar foi a escolha de qual das ferramentas de desenvolvimento de realidade aumentada utilizar. Graças ao estudo do estado da arte desenvolvido, percebemos que todas elas possibilitam alcançar as funcionalidades propostas nos requisitos, tendo por isso a escolha descaído sobre os seguintes fatores:

- **Possibilidade de desenvolver para ambas as plataformas**, android e iOS, em **simultâneo**, facilitando bastante o desenvolvimento da aplicação.
- **Dispositivos suportados**: o ideal seria suportar a maior quantidade de dispositivos possíveis para que todos possam utilizar a aplicação, no entanto, é necessário ter em conta que, dispositivos mais antigos poderão não ter o desempenho desejado.
- **Preço**: o ideal será alcançar os requisitos pretendidos gastando o menos possível.
- **Funcionalidades de *Image Tracking***: sendo esta a funcionalidade principal da nossa aplicação será a que terá maior impacto na escolha da ferramenta. Apesar da implementação ser fazível através de qualquer uma das ferramentas de desenvolvimento investigadas, não significa que o façam da mesma forma ou que o resultado obtido seria o mesmo.

As ferramentas ARCore e ARKit apenas possibilitam o desenvolvimento para Android ou iOS respetivamente. Para além disso o AR Foundation permite o desenvolvimento através das duas em simultâneo, não existindo nenhuma razão para usar qualquer uma individualmente.

Tanto o Vuforia como o Wikitude abrangem uma maior quantidade de dispositivos Android do que o AR Foundation.

Tabela 17 - Versões Android e iOS suportadas pela AR Foundation, Vuforia e Wikitude.

	AR Foundation	Vuforia	Wikitude
Android	>=7.0	>=5.1.1	>= 5.0
iOS	>=11.0	>=12.0	>=11.0

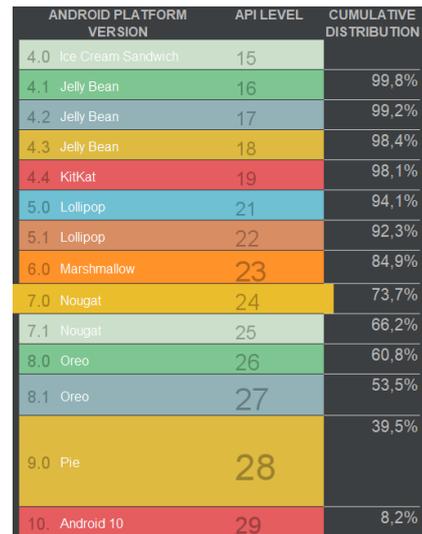


Figura 58 - Estatística de distribuição das versões de Android.
Fonte: Android Studio.

De forma a saber se a diferença na quantidade de dispositivos suportados seria significativa, e uma vez que não conseguimos obter as estatísticas das versões de android utilizadas pelos funcionários da Critical Software, recorreremos às da Google. De acordo com a informação recolhida através do Android Studio (figura 58), cerca de 74% dos dispositivos android no mundo têm a versão 7.0 ou superior. Sabendo que este número tem apenas tendência para aumentar, e que dispositivos com versões anteriores a esta poderão não ter o desempenho adequado, este fator acabou por não ter grande impacto na escolha do SDK.

O preço foi o fator que descartou o Wikitude das nossas escolhas, uma vez que, para além de ser caro e de se pagar durante o desenvolvimento, não oferece nada para o âmbito do projeto, que outras ferramentas como o Vuforia e o AR Foundation não tenham. O Vuforia apesar de ser pago, não se paga durante desenvolvimento, e é especializado em realidade aumentada com base em marcadores, sendo por isso uma alternativa viável até este ponto.

O Vuforia é especialmente reconhecido para a aplicação de realidade aumentada através de marcadores. Funcionalidades como **Image Targets**, **Object Targets**, ou **VuMarks** seriam importantes para o contexto do projeto deste estágio. No entanto os requisitos propostos requerem apenas o reconhecimento de imagens, algo que o AR Foundation também consegue para ambas as plataformas (através do AR Foundation apenas é possível o reconhecimento de objetos em iOS). Logo não faz sentido estar a pagar pelo Vuforia, uma vez que, não se irá tirar partido de nenhuma funcionalidade que o AR Foundation não tenha. Para além disso, muitas das funcionalidades disponibilizadas pelo Vuforia exigem o uso ferramentas externas, algo que não pretendemos uma vez que queremos manter o processo de criação das Campanhas simples, através da nossa própria interface. No entanto, tal como foi dito durante o estudo do estado da arte, o uso destas não é mutuamente exclusivo, podendo as 2 ferramentas de desenvolvimento ser utilizadas em simultâneo em caso de necessidade.

Optámos assim pelo AR Foundation, uma vez que permite cumprir com os requisitos funcionais propostos, permite o desenvolvimento para ambas as plataformas em simultâneo, é gratuito e

sendo desenvolvido e mantido pela Unity Technologies tem melhor integração e documentação com a plataforma Unity do que as restantes.

Fases de funcionamento

Através da definição dos casos de uso conseguimos identificar 2 fases de funcionamento da aplicação:

1ª - Fase de inicialização: Esta é a fase responsável por verificar se existem Campanhas ativas, fazer o download dos Triggers e das Ações necessárias, e eliminar as que já não o são.

2ª - Fase de funcionamento normal: Funcionamento normal da aplicação de reconhecimento de Triggers e visualização das respetivas Ações.

Começando pela **fase de inicialização**, um dos aspetos mais importantes desta fase é atualização das Campanhas. Para isso a aplicação faz um pedido ao Backend que devolve todas as Campanhas que se encontram ativas, incluindo informação sobre Triggers, respetivas Ações e propriedades de cada uma. Através desta informação obtida, a aplicação verifica se tem na cache todos os Triggers e Ações necessários para as Campanhas ativas, fazendo o download dos quais isso não se verifique. Triggers e Ações, de Campanhas anteriores, que deixam de ser necessários, são eliminados da cache.

Para que os Trigger e as Ações possam ser utilizadas pela aplicação sem a necessidade de uma nova compilação, têm de vir num formato específico denominado *AssetBundles*. Estes *Bundles* são criados no Backend (detalhado na secção seguinte 4.2), são específicos para cada plataforma (ou seja, o *AssetBundle*, de uma Ação, *para* android é diferente do *AssetBundle*, da mesma Ação, *para* iOS) e podem conter vídeos, áudio, texturas, modelos, entre outros, que podem ser carregados e utilizados durante o *runtime* da aplicação. Assim, quando a aplicação faz download de um Trigger ou Ação, o que realmente ela está a fazer é a ir buscar o *Bundle* destes, para que possam ser utilizados pela aplicação. A utilização dos *AssetBundles* não só permite-nos usar conteúdo que não foi compilado juntamente com a aplicação, mas também, **otimizar/processar o conteúdo** de acordo com a plataforma, **reduzir o consumo de memória** durante *runtime*, **reduzir o tamanho/tempo do download e o espaço ocupado pela aplicação** uma vez que pode ser aplicado um algoritmo de compressão.

Na **fase de funcionamento normal** é onde se coloca em prática as várias funcionalidades disponibilizadas pela ferramenta de desenvolvimento de realidade aumentada escolhida, AR Foundation:

- **Device Tracking:** de forma a saber a posição e orientação do dispositivo num dado espaço.
- **Plane & Feature Point Detection:** planos e pontos de referência (*feature points*) detetados servem, tal como o nome indica, como pontos de referências para elementos/objetos virtuais colocados num espaço.
- **Image Tracking:** Para reconhecimento e rastreamento dos Triggers.
- **Anchor:** Quando um clone de uma Ação é colocado em qualquer local de um espaço é lhe atribuído uma *Anchor*, para que a aplicação consiga rastrear a sua posição e orientação ao longo do tempo através dos pontos de referência detetados.

- **Raycast:** De forma a conseguir seleccionar objetos quando o utilizador toca no ecrã. Este toque é convertido num *ray* (definido por uma origem e direção) capaz de interseção com objetos colocados.

Outra funcionalidade que gostaríamos de ter implementado, ficando apenas como uma possível melhoria futura, seria a *Occlusion/Depth API* que traria melhorias ao realismo da experiência através da oclusão de objetos virtuais por reais. Esta não foi implementada uma vez que nenhum dos dispositivos que tínhamos para testes a suportava.

Por último ainda sobre a aplicação, um aspeto importante que foi tido em conta durante o desenvolvimento da mesma, foi a necessidade de desenvolvê-la de forma que facilmente seja possível adicionar novas propriedades de customização das Ações. Para isso adotamos um padrão de desenvolvimento da aplicação com base em Componentes. Desta forma cada tipo de Ação possui várias componentes que facilmente podem ser alteradas ou, substituídas por novas. Dando um exemplo cada tipo/formato de ação, possui uma Componente denominada *TransformComponent* que é responsável por controlar a escala, posição e orientação das Ações para que facilmente se consiga adotar comportamentos diferentes de acordo com as propriedades. Outro exemplo seria a *ParticleComponent* responsável por controlar o efeito de partículas que surgem quando um objeto é selecionado (figura 59), através desta componente facilmente conseguiríamos adicionar propriedades que, por exemplo, permitam definir o tamanho ou a cor destas partículas.



Figura 59 -Exemplo de sistema de partículas (Branças) em torno do objeto selecionado.

Assim, através deste padrão de desenvolvimento, facilmente conseguimos alterar o comportamento dos objetos, através de componentes existentes ou criando novas, caso surjam novas propriedades de customização no futuro.

Para esta componente optámos por usar Docker e dividir os serviços por containers devido à:

- **Portabilidade:** Graças ao desenvolvimento de um *Dockerfile* (documento com instruções para a criação dos containers de forma automática) para cada container, e de um *docker-compose* (ferramenta/documento que permite definir e configurar aplicações Docker com vários containers), facilmente se consegue colocar o sistema a correr em qualquer local onde o Docker esteja a funcionar, sabendo que aplicação irá funcionar sempre da mesma forma, uma vez que o ambiente é consistente e isolado.
- Facilidade de **deploy** e **manutenção** permitindo aplicar métodos automáticos de CI/CD (*Continuous integration / Continuous delivery*) perante alterações ou melhorias aplicadas aos serviços.
- **Escalabilidade:** Facilmente podem ser criados novos containers, em caso de necessidade.

De seguida são explorados com maior detalhe cada um dos containers desenvolvidos nomeadamente, as tecnologias utilizadas, o propósito e o funcionamento de cada um.

Api Privada

A Api Privada é o container principal, responsável por fornecer todas as funcionalidades descritas nos requisitos funcionais através de uma *Web API*. Esta foi desenvolvida a partir da *framework* .Net Core 3.1 usando a linguagem C#. Sendo a *framework* de escolha da Critical, através do uso desta não só consegui ter um maior nível de apoio ao longo do desenvolvimento, e consequentemente aprender mais, mas também facilita a integração com outros sistemas da Critical, caso venha a ser necessário.

Para este serviço foi adotada uma arquitetura baseada na *Onion Architecture*. Esta arquitetura está dividida em camadas e tem como objetivo aumentar a flexibilidade e diminuir o acoplamento entre estas. A figura seguinte demonstra as várias camadas adotadas, as componentes de cada uma e as dependências que existem:

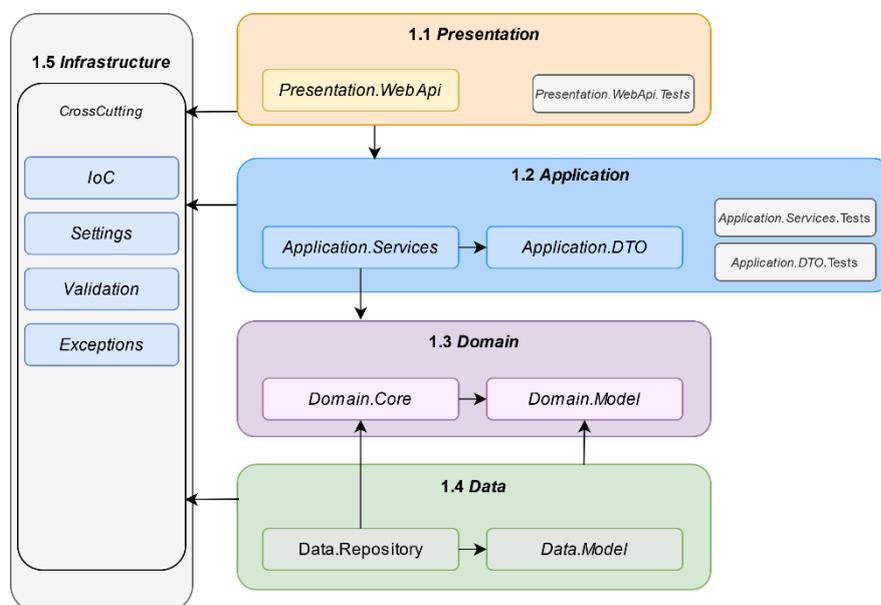


Figura 61 – Camadas utilizadas pela Web API seguindo a Onion Architecture.

Presentation: camada acessível pelos utilizadores através dos *endpoints* definidos pela *Web API* (*Presentation.WebApi.Tests* é a componente de testes unitários).

Application: camada composta pela lógica da aplicação e com a definição das entidades para uso externo (DTO – *Data Transfer Objects*). (*Application.Service.Tests* e *Application.Dto.Tests* são componentes de testes unitários).

Domain: camada central que possui a definição das entidades do negócio (*Domain.Model*, ex.: Triggers, Ações, Campanhas) e interfaces (*Domain.Core*, ex.: Interface do Repositório) (Interfaces definem contratos que devem ser cumpridos pela classe que as está a implementar).

Data: camada de acesso a dados (deve implementar a Interface do Repositório).

Infrastructure: camada responsável por fornecer serviços ou configurações utilizadas pelas várias camadas (*CrossCutting*).

As setas da figura representam as dependências, no entanto deve ser clarificado que estas setas podem começar ou terminar tanto numa componente como numa camada, retratando casos diferentes. Quando a seta inicia numa componente (ex.: *Application.Services*), apenas essa componente possui a dependência. Quando a seta inicia numa camada (ex.: 1.2 *Application*) todas as componentes dessa camada possuem a dependência. O mesmo se aplica para a ponta da seta. Dando dois exemplos, temos:

Application.Services -> 1.3 *Domain*: a componente *Application.Services* depende de todas as componentes dentro da Camada 1.3 *Domain*.

Data.Repository -> *Domain.Core*: a componente *Data.Repository* depende da componente *Domain.Core*.

Como mencionado anteriormente, o objetivo desta arquitetura é diminuir as dependências / acoplamento entre camadas e evitar dependências em cascata, em que uma camada depende sempre da seguinte (caso da *N-Layer Architecture*).

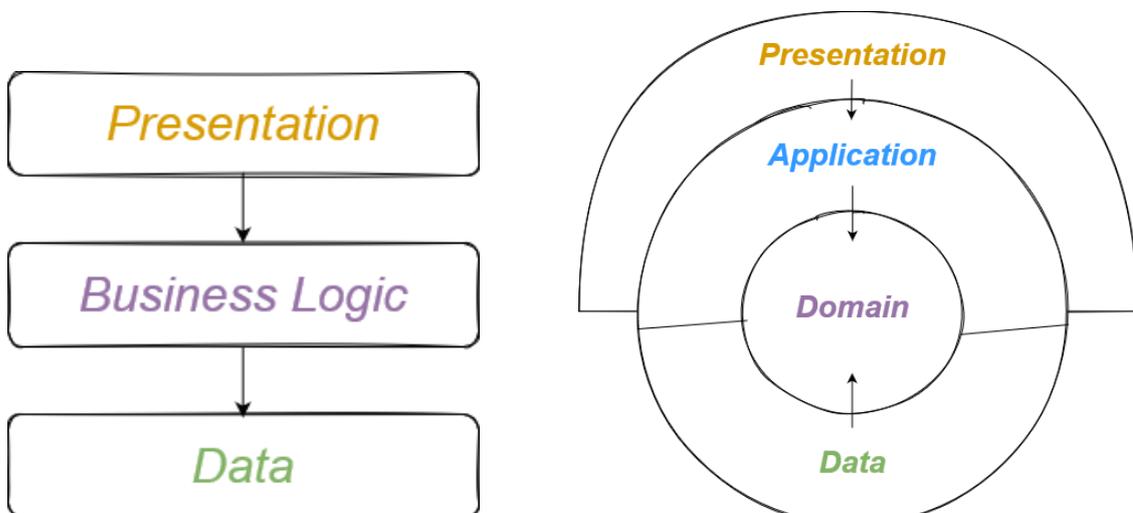


Figura 62 – Diferença de acoplamento das camadas entre a N-Layer Architecture (esquerda) e Onion Architecture (direita).

Um dos maiores problemas da *Layer Architecture* é a dependência ou acoplamento à camada de dados (*Data*). No caso da figura anterior, a camada de apresentação não funciona sem a

lógica de negócio que por sua vez necessita de acesso aos dados, sendo que, as dependências aumentam com a inserção de novas camadas. Este acoplamento torna difícil a manutenção e a capacidade de atualização e modificabilidade do sistema.

No caso da *Onion Architecture*, o objetivo é controlar o acoplamento. Camadas externas dependem apenas das camadas mais centralizadas, ou seja, o acoplamento é na direção do centro. No centro encontra-se a camada **Domain**, esta não depende de nenhuma outra camada, e possui a definição de entidades (ex.: Trigger, Ação, Campanha) e interfaces. Um exemplo de interface definida na camada *Domain* é a Interface do repositório, que define o comportamento de acesso aos dados que deve ser cumprido pela camada **Data**. De forma a clarificar temos o seguinte exemplo:

Na camada *Domain* foi definida uma Interface que descreve o método, *GetAllCampaigns*, que devolve uma lista com objetos da entidade **Campanha** (entidade definida na mesma camada). A camada *Data*, deve assim implementar um método que possua esse mesmo comportamento descrito. Independentemente da implementação e do tipo de base de dados utilizada, quer seja Sql ou NoSql, o método deve devolver a lista comprometida.

Esta partilha de interfaces verifica-se ao longo das restantes camadas e baseia-se no *Dependency Inversion Principle* (para mais informação ver referência [72]).

A adoção deste padrão arquitetural não só nos permite ter um projeto mais bem estruturado e definido, mas também facilita o desenvolvimento de testes, devido ao acoplamento controlado, e aumenta a modificabilidade da implementação de cada camada, incluindo a camada de dados, uma vez que estas apenas necessitam de respeitar as Interfaces definidas.

Em relação a testes no total foram definidos 157 testes unitários e de integração. Os testes unitários permitem testar funcionalidades individuais de cada camada, enquanto os testes de integração permitem validar funcionalidades entre camadas, ou sejam funcionalidades que envolvam várias camadas.

Ainda no Container Api Privada estão a ser utilizadas duas ferramentas:

- **Arcoreimg** – desenvolvida pela Google, faz a avaliação das imagens que estão a ser utilizadas como Trigger, devolvendo uma pontuação que indica a qualidade das mesmas para tal.
- **Fbx2Gltf** – desenvolvida pela Facebook, permite converter modelos 3D no formato fbx para o formato Gltf. Este último formato, para além de possuir menores dimensões, possui uma maior compatibilidade com a ferramenta usada para mostrar modelos 3D na Interface de Gestão do Conteúdo. A Aplicação móvel usa o ficheiro original fbx.

Por último, em relação a medidas de segurança, existem algumas a salientar. A primeira é a validação de todos os dados inseridos pelo utilizador, incluindo ficheiros. No caso dos ficheiros é feita a validação da extensão, do tamanho e do conteúdo do mesmo. Todos os formatos de ficheiros utilizados possuem uma assinatura específica que permite verificar se o conteúdo do ficheiro corresponde à extensão do mesmo (por exemplo, no caso de um ficheiro png os primeiros 8 bytes do ficheiro são sempre “0x89, 0x50, 0x4E, 0x47, 0x0D, 0x0A, 0x1A, 0x0A” em hexadecimal). Em relação aos restantes inputs foram usadas as medidas tradicionais de validação de *strings* como por exemplo expressões regulares.

Outra medida de segurança adotada é o facto de, para conseguir realizar pedidos a esta *Web API* é necessário que o pedido provenha da rede interna da Critical Software. Desta forma

garantimos que apenas funcionários da Critical conseguem fazer alterações nas Campanhas, uma vez que é necessária conta de funcionário para entrar nesta rede. Esta regra de segurança foi colocada na própria máquina do serviço Azure através do mapeamento de portos. Por outras palavras, apenas pedidos com endereços IP específicos à rede interna da Critical conseguem aceder ao porto através do qual a Api Privada está a receber os pedidos.

Api Pública

Tal como mencionado previamente, a Api Pública funciona como um proxy para Api Privada, disponibilizando apenas os métodos necessários para a Aplicação móvel. Apesar da Aplicação móvel ser apenas para uso interno na Critical, poderão existir ocasiões nas quais os utilizadores ainda não possuem acesso à sua conta para aceder à rede interna (por exemplo, iniciativa OnBoarding de boas-vindas a novos funcionários). Para além disso a necessidade de instalar uma VPN para que a Aplicação possa ser usada a partir de casa, levou à decisão de tornar públicas as funções necessárias para a Aplicação. No total a Api Pública disponibiliza cinco *endpoints* da Api Privada, um que devolve todas as Campanhas que se encontram ativas, e os restante quatro para fazer o download dos *AssetBundles* de Triggers/Actions para Android/iOS.

Container Unity

Este é o container responsável por fazer o processamento dos Triggers e das Ações, e construir os respetivos *AssetBundles* para que possam ser utilizados pela Aplicação móvel. Sempre que é adicionado um novo Trigger ou Ação através da Api Privada, esta faz um pedido http a este container para preparar o conteúdo de forma a poder ser usado.

Este Container possui acesso a um projeto Unity criado especificamente para o propósito de fazer o processamento do conteúdo de acordo com as plataformas nas quais será utilizado. Ao receber o pedido da Api Privada, são executadas duas instâncias do Unity Editor (subsequentemente), uma para Android e outra para iOS, que abrem este projeto e executam o método de criação dos *AssetBundles*.

De forma a clarificar, segue-se uma descrição, simplificada, por etapas, que descreve o funcionamento deste Container:

1º - Após ser adicionada uma nova Ação ou Trigger, através da Api Privada, esta faz uma copia do ficheiro para a pasta do projeto Unity. Posteriormente faz um pedido http para o Container Unity para criar os *AssetBundles* do novo conteúdo.

2º - Container Unity executa uma instância do Unity Editor, que abre o projeto e executa o método de criação de *AssetBundles*.

```
Unity.exe -c -quit -batchmode -nographics -buildTarget Android -projectPath /usr/UnityProject -executeMethod BuildActionAssetBundles actionName=myAction actionType=model actionPath=/usr/UnityProject/Assets/myAction.fbx
```

Figura 63 - Exemplo de comando de inicialização de uma instância do Unity Editor, com plataforma alvo Android.

3º - Sempre que é aberta uma nova instância, o Unity Editor faz *import* automático de novos *assets* presentes na pasta do projeto (este *import* tem em conta a plataforma escolhida, ou seja, otimiza o conteúdo de acordo com a plataforma para a qual se destina). Tirando partido desta funcionalidade foi desenvolvido um script que altera algumas propriedades do *import*, de acordo com as características do ficheiro, como por exemplo:

- Não alterar escala das imagens para potências de dois (comportamento por *default* é alterar a escala para a potência de dois mais próxima).
- Ficheiros de vídeo com transparência são convertidos de forma a adicionar uma representação interna do canal alfa, para que esta transparência funcione em dispositivos Android.
- São extraídas todas as animações dos modelos 3D e é criado/adicionado um controlador para as mesmas.
- São adicionados *colliders* aos modelos 3D de forma que seja possível detetar colisões através de *Raycasting*.

5ª- Após todas as etapas do *import*, é criado o *AssetBundle* da Ação ou Trigger para a plataforma selecionada. Processo repete-se para a outra plataforma.

6ª Após a criação dos *Bundles* para ambas as plataformas, é eliminada a cópia do ficheiro (Ação ou Trigger) da pasta do projeto Unity, e devolve-se uma resposta à Api Privada que indica que o processo terminou.

7ª Api privada verifica se os *AssetBundles* foram criados (se existem) e atualiza a base de dados em conformidade.

Container FFmpeg

Este container permite-nos fazer a conversão de vídeos, através da ferramenta open-source FFmpeg, para formatos suportados pelo Unity Editor instalado em subsistemas Linux.

Extension	Windows	macOS	Linux
.asf	✓		
.avi	✓		
.dv	✓	✓	
.m4v	✓	✓	
.mov	✓	✓	
.mp4	✓	✓	
.mpg	✓	✓	
.mpeg	✓	✓	
.ogv	✓	✓	✓
.vp8	✓	✓	✓
.webm	✓	✓	✓
.wmv	✓		

Figura 64 – Tabela com os formatos suportados pelo Unity Editor para cada plataforma. Fonte: [73]

Assim, quando é adicionada uma nova Ação, e sendo esta um vídeo, é feito um pedido para este Container, que, por sua vez, verifica se o formato do vídeo é válido para ser processado pelo Unity. Caso não o seja é feita a conversão para um formato válido. De acordo com a documentação o ideal para Linux será um container webm, com o codec de vídeo VP8 e de áudio Vorbis, sendo esta a especificação utilizada na conversão.

Este processamento de vídeo ocorre antes das operações realizadas pelo Container Unity.

Na conversão é ainda colocado um limite de qualidade de forma que os vídeos consumam menos recursos tanto da Aplicação móvel como do próprio Backend.

Base de dados

Em relação à base de dados, optámos por NoSQL uma vez que, não necessitamos de relacionamentos entre os objetos, as *queries* necessárias são simples sem necessidade de *joins* ou operações mais complexas, e o facto de não existir um esquema pré-definido facilita bastante perante a necessidade de alterações no sistema (por exemplo se quisermos adicionar mais propriedades a cada tipo de Ação).

Dentro das bases de dados NoSQL, optámos por usar MongoDB devido, à sua simplicidade, sendo os dados guardados sobre a forma de documentos Json/Bson, facilita a conversão de pedidos à Api Privada em operações CRUD (*Create, Read, Update, Delete*) à base de dados, e uma vez que os documentos não necessitam de ter qualquer tipo de esquema.

A base de dados é composta por 3 Coleções:

- **Ações** – onde é guardada informação sobre cada Ação, incluindo:
 - **Nome** dado à Ação
 - **Tipo** da Ação (Imagem, Vídeo, Áudio, Modelo)
 - **Nome do ficheiro**
 - **Extensão do ficheiro**,
 - **Status do *AssetBundle***:
 - *-1*: Ocorreu algum erro a construir o *AssetBundle* da Ação
 - *0*: *AssetBundle* está ainda a ser processado
 - *1*: *AssetBundle* foi construído com sucesso
 - Dicionário com **propriedades** de **customização** disponíveis
- **Triggers** – onde é guardada informação sobre cada Trigger, incluindo:
 - **Nome** dado ao Trigger
 - **Hash** da imagem (para evitar imagens duplicadas)
 - **Qualidade** da Imagem (calculada através da ferramenta *Arcoreimg*)
 - **Nome do ficheiro**
 - **Extensão do ficheiro**
 - **Status do *AssetBundle***:
 - *-1*: Ocorreu algum erro a construir o *AssetBundle* do Trigger
 - *0*: *AssetBundle* está ainda a ser processado
 - *1*: *AssetBundle* foi construído com sucesso
- **Campanhas**
 - **Nome** dado à Campanha
 - **Data de início**
 - **Data de fim**
 - Lista com **Triggers** e respetivas **Ações** e **propriedades** definidas

7.3 Interface de Gestão do Conteúdo

A interface de gestão do conteúdo possibilita o uso das funcionalidades disponibilizadas pelo Backend (mais concretamente a Api Privada) de forma que os Gestores consigam criar e editar as suas Campanhas. Esta componente foi desenvolvida em React, uma vez que se pretendia algo simples, de baixa complexidade, sendo a curva de aprendizagem de React menor quando comparada com outras ferramentas (ex.: Angular, destinada para projetos maiores, mais complexos).

No desenvolvimento desta interface, foi adotado o padrão típico de React de relacionamento entre componentes pai e filhos, no qual os pais comunicam com os filhos através de *props* (parâmetros) e os filhos comunicam com os pais através de eventos.

Nota: Esta componente do sistema funciona *client-side*, sendo as páginas fornecidas através de um container Docker com um servidor nginx, presente no Backend. Optámos, por uma questão de simplicidade e para mais facilmente distinguir as duas componentes, por não demonstrar isso na figura da arquitetura do sistema. Tal como os pedidos para a Api-Privada, para ter acesso a esta componente é necessário estar conectado à rede interna da Critical.

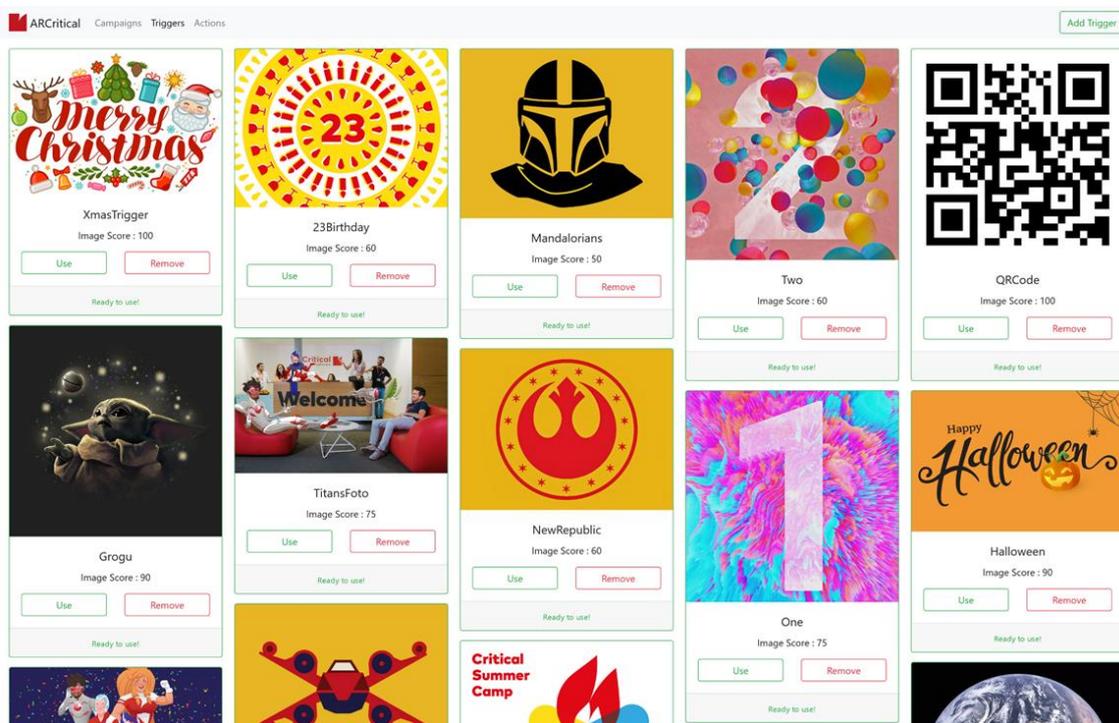


Figura 65 - Separador dos Triggers da Interface de Gestão do Conteúdo.

Capítulo 8

Funcionamento

Neste capítulo é feita uma descrição do funcionamento do sistema como um todo (*end-to-end*), com recurso a exemplos.

Estando dentro da intranet da Critical, quer seja através do uso de uma VPN ou estando fisicamente num dos escritórios, através da Interface de Gestão do Conteúdo conseguimos tirar partido das funcionalidades disponibilizadas pelo Backend para a criação de Campanhas de Realidade Aumentada.

Esta Interface é constituída por 3 separadores, um que permite gerir as Campanhas criadas, outro para os Triggers e, por último, um para as Ações.

Começando pelo separador dos Triggers, este, para além de permitir adicionar novas imagens para serem utilizadas como Trigger, permite também visualizar todas as que foram previamente adicionadas. Para além disso, para cada uma destas imagens indica, a respetiva qualidade para serem usadas como Trigger, e, se se encontram prontas para serem utilizadas (ou seja, se o Backend já fez o processamento das mesmas).

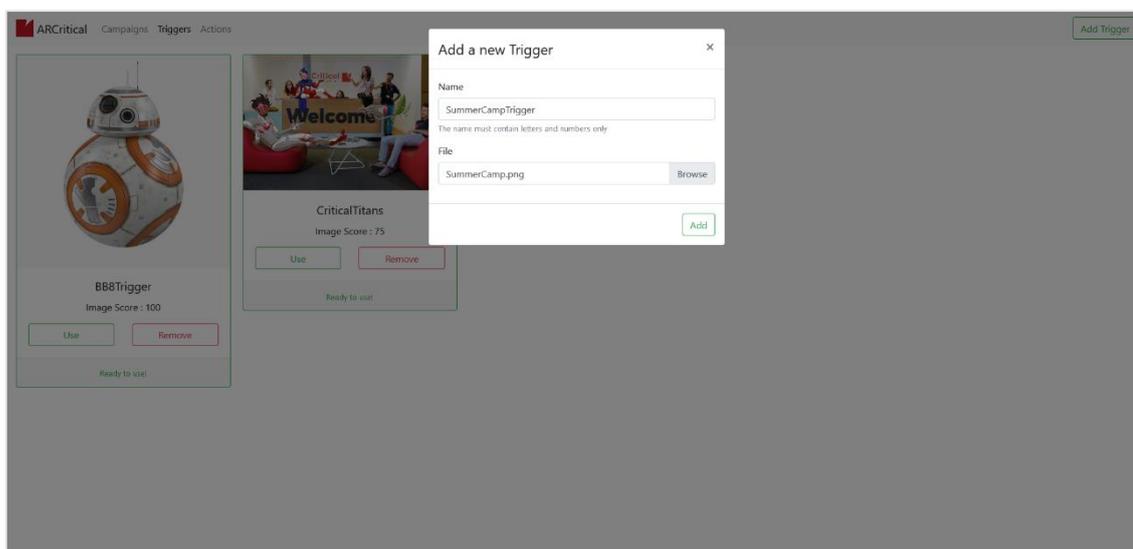


Figura 66 - Adicionar um novo Trigger (*SummerCampTrigger*) através da Interface de Gestão do Conteúdo.

Ao ser adicionado um novo Trigger, o Backend é responsável por fazer o processamento do mesmo para que possa ser utilizado pela Aplicação durante a execução.

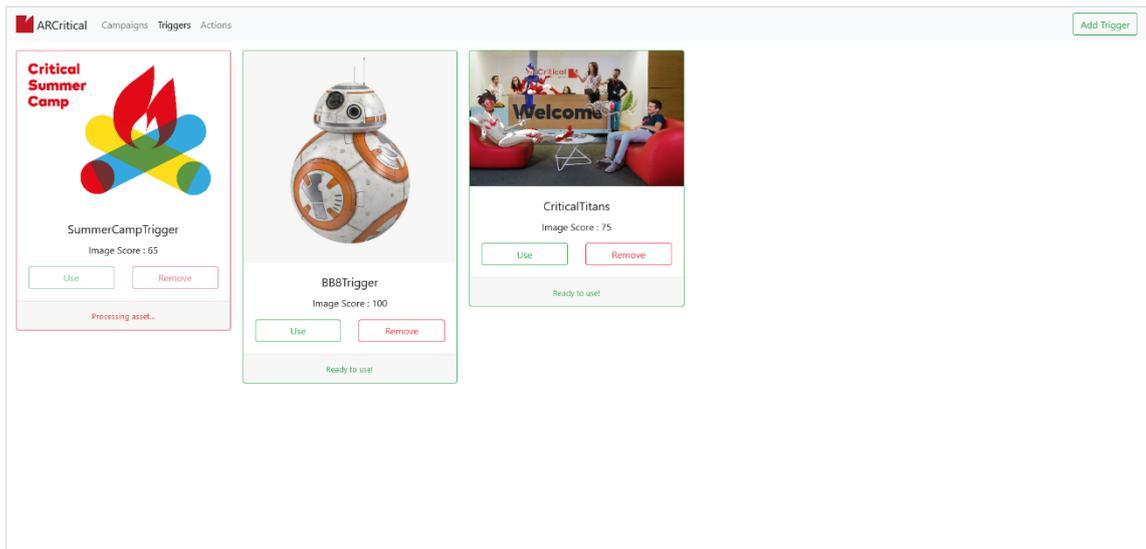


Figura 67 - Separador que contém todos os Triggers já adicionados. O Backend está ainda a processar o novo Trigger para que possa ser utilizado pela Aplicação.

O separador das Ações, permite adicionar imagens, vídeos, áudio ou modelos 3D para serem utilizadas como Ações. Além disso permite visualizar as Ações previamente adicionadas e, para cada uma destas, indica se já se encontram prontas para serem utilizadas.

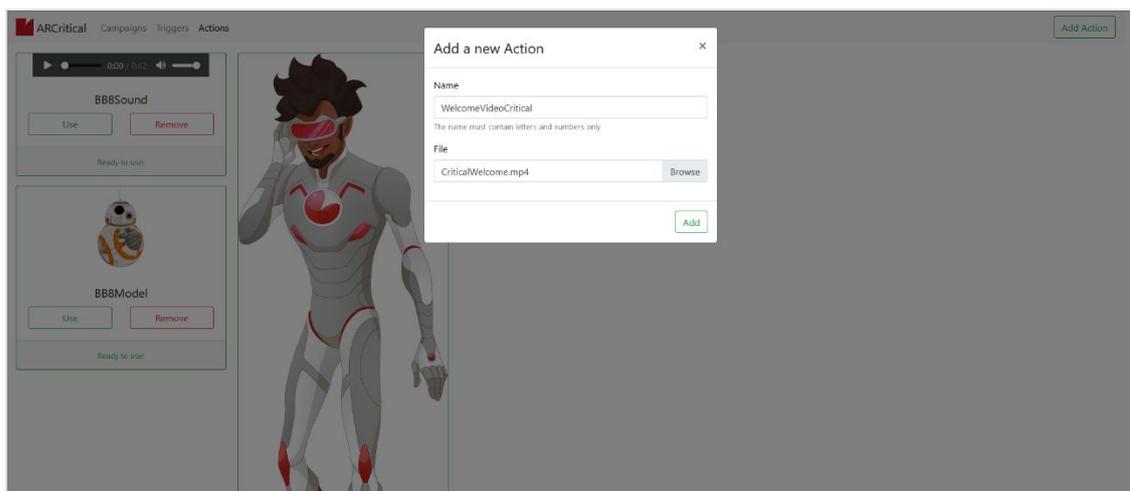


Figura 68 - Adicionar uma nova Ação (*WelcomeVideoCritical*) através da Interface de Gestão do Conteúdo.

Tal como acontece com os Triggers, as Ações, quando adicionadas, são processadas pelo Backend para que possam ser utilizadas pela Aplicação. Apenas após este processo estar concluído é que podem ser associadas a um Trigger de uma qualquer Campanha.

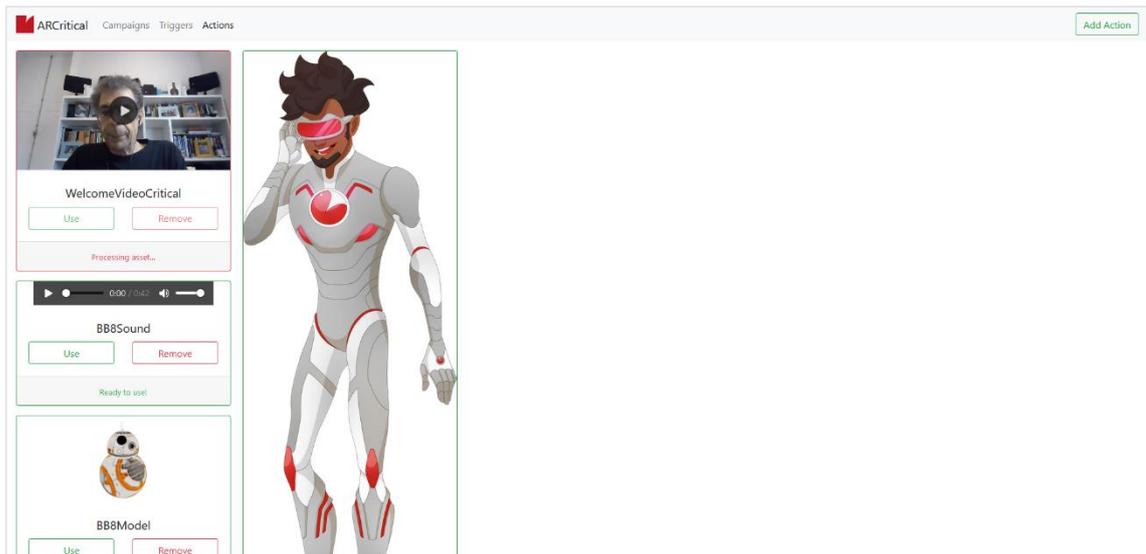


Figura 69 - Separador que contem todas as Ações já adicionadas. O Backend está ainda a processar a nova Ação para que possa ser utilizada pela Aplicação.

Após adicionar os Triggers e as Ações desejadas, é no separador das Campanhas onde conseguimos visualizar e gerir Campanhas existentes ou criar novas.

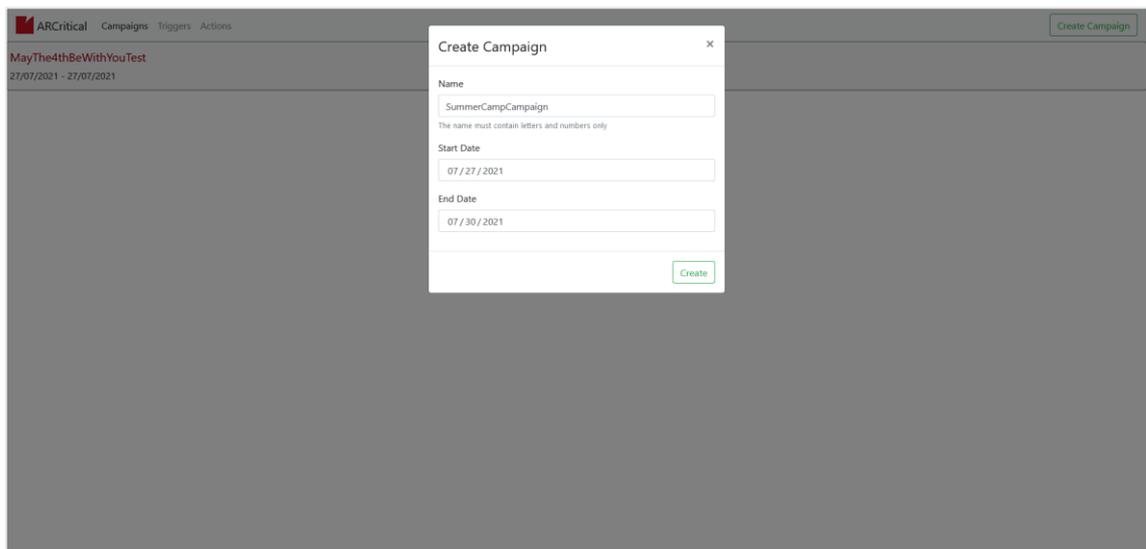


Figura 70 - Criação de uma nova Campanha, através da Interface de Gestão do Conteúdo. Nome dado à campanha: *SummerCampCampaign*, Data de início: 27/07/2021, Data de fim: 30/07/2021

Às Campanhas criadas, podemos associar qualquer Trigger que já esteja pronto para ser utilizado.

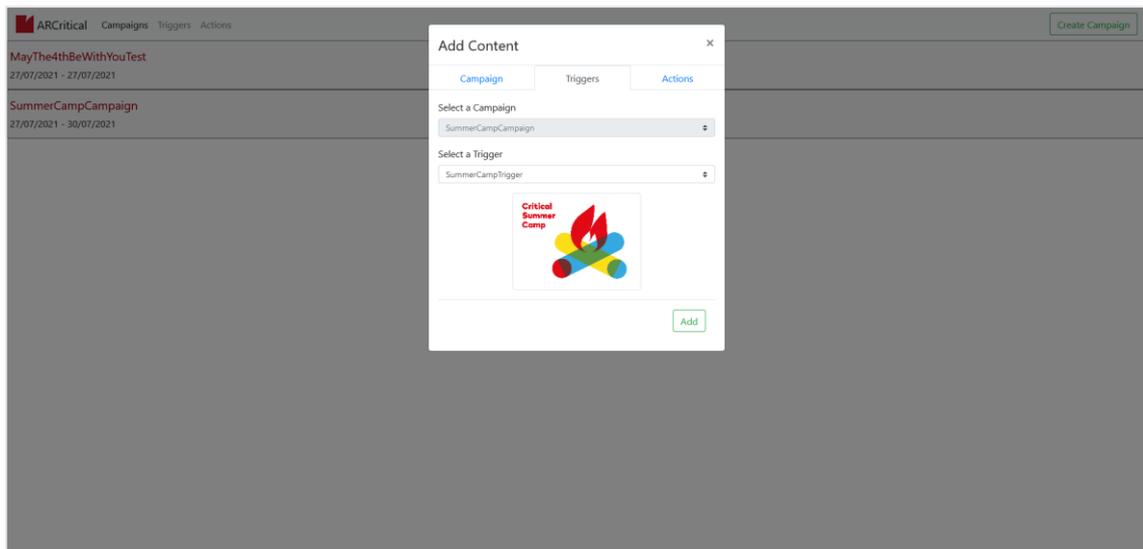


Figura 71 - Associar Trigger (*SummerCampTrigger*) à Campanha criada (*SummerCampCampaign*).

Aos Triggers associados a determinada Campanha podemos associar uma ou mais Ações, desde que estejam prontas para serem utilizadas. Desta forma, durante o período em que a Campanha está ativa, a Aplicação móvel, ao reconhecer os Triggers, irá mostrar as respetivas Ações em Realidade Aumentada.

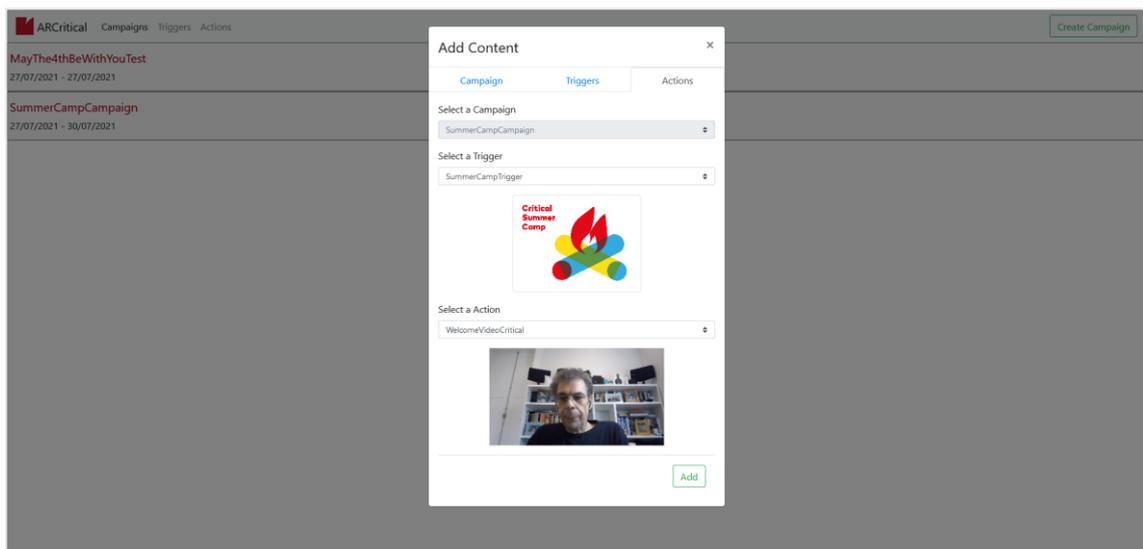


Figura 72 - Associar uma Ação (*WelcomeVideoCritical*) a um determinado Trigger (*SummerCampTrigger*) de uma Campanha (*SummerCampCampaign*).

Ainda no separador das Campanhas é possível visualizar o conteúdo de cada uma, incluindo os Triggers que já foram associados e respetivas Ações.

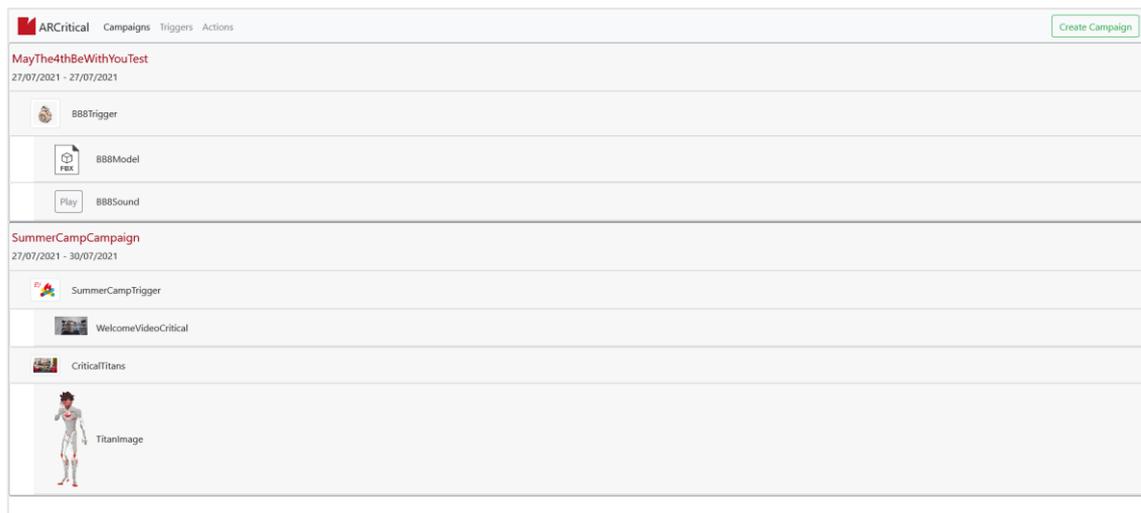


Figura 73 - Triggers e respetivas Ações de cada uma das Campanhas já criadas.

Para além disso, é aqui onde se consegue customizar cada Ação de acordo com as propriedades disponíveis.



Figura 74 – Customização de cada Ação de acordo com as propriedades disponíveis.

Neste exemplo pretendemos que, a Ação (*WelcomeVideoCritical*) ao ser despoletada através do Trigger (*SummerCampTrigger*) possua as seguintes propriedades:

- **Can be transformed:** queremos que a Aplicação permita ao utilizador rotacionar ou alterar a escala da Ação.
- **Can be cloned:** queremos que a Aplicação permita ao utilizador criar um clone da Ação de forma que possa colocá-la em qualquer lugar, deixando a sua posição de estar associada ao Trigger.
- **Loop:** queremos que o vídeo se mantenha em loop, enquanto estiver visível.
- **Has a control panel:** queremos que o vídeo possua um painel de controlo que permita pausar/continuar o mesmo e verificar o seu progresso.
- **Is transparent:** não queremos manter a transparência, o vídeo é opaco.

- **Keep video's aspect ration:** não queremos manter a proporção do vídeo. Queremos que o vídeo ocupe todo o tamanho do Trigger.

Após concluir a criação e gestão das Campanhas, é altura de abrir a Aplicação móvel. Ao abrir a aplicação, esta verifica se existem atualizações e se necessita de fazer o download de novos Triggers ou Ações. Caso seja necessário, a aplicação dá a escolha ao utilizador se pretende fazer o download agora ou mais tarde. Se o utilizador optar por fazer o download mais tarde, a aplicação usa apenas os Triggers e as Ações, de Campanhas ativas, que já se encontram disponíveis no dispositivo (se existirem).



Figura 75 - Utilizador pode optar por fazer o Download agora ou mais tarde.



Figura 76 - Progresso do Download de novos conteúdos.

Após concluído o Download, ao pressionar o botão “Continue”, inicia-se a fase de funcionamento normal da Aplicação. A Aplicação indica quantos Triggers são reconhecíveis:

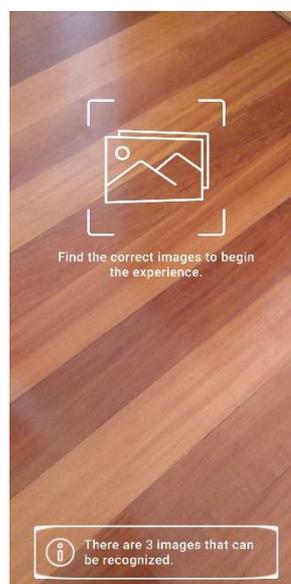


Figura 77 – Aplicação indica quantas imagens são reconhecíveis, ou seja, quantos Triggers existem.

Ao ser reconhecido qualquer um destes Triggers, são mostradas as respetivas Ações, em Realidade Aumentada, com as propriedades definidas.



Figura 78 - Ao ser reconhecido o Trigger a Aplicação mostra as respetivas Ações sobrepostas a este.

De acordo com as propriedades definidas, poderá ser possível seleccionar uma Ação através de um toque no ecrã, podendo esta ser transformada ou clonada e colocada em qualquer local. Segue-se um vídeo a demonstrar esse mesmo comportamento.

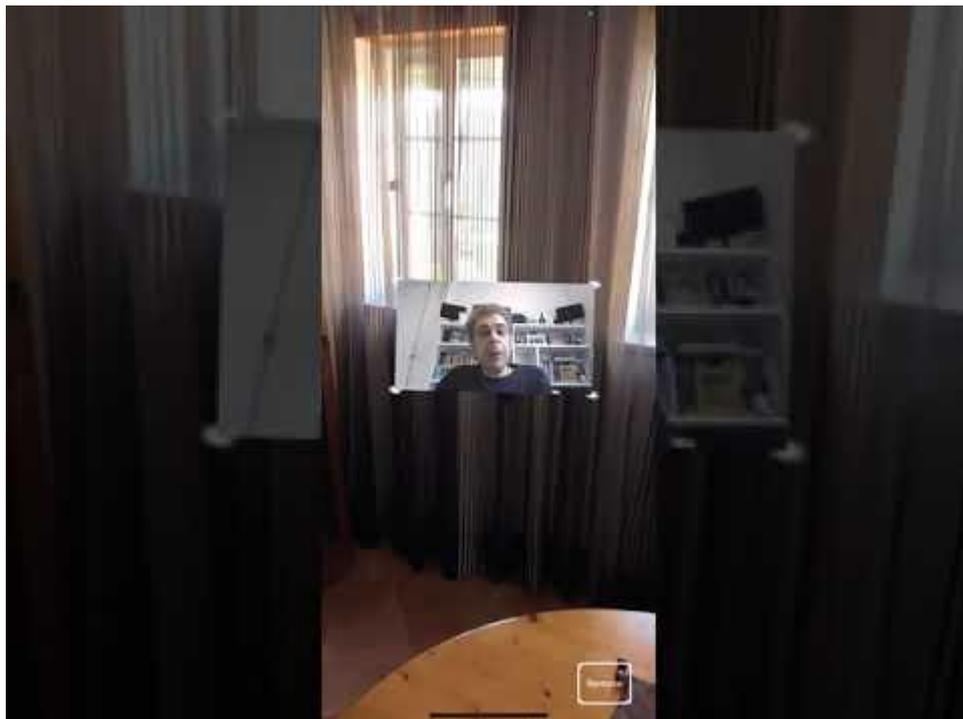


Figura 79 - Vídeo de demonstração do funcionamento da Aplicação

Capítulo 9

Testes

Neste Capítulo são especificados alguns dos testes (*end-to-end*) realizados com intuito de, não só validar o comportamento do sistema, mas também, verificar o impacto da qualidade do conteúdo utilizado sobre o desempenho tanto da Aplicação móvel como do Backend.

Estes testes foram realizados já com o sistema todo funcional, ou seja, o Backend já a correr na máquina disponibilizada pelo serviço Azure. Foi utilizado o dispositivo Oneplus 5T com: Snapdragon 835 Qualcomm, 6GB de RAM e Android 10, para testar a aplicação.

Antes de apresentar os resultados obtidos existem alguns pontos que devem ser tidos em conta perante a análise destes.

- Resultados associados à Aplicação móvel são dependentes do dispositivo utilizado. O facto de o dispositivo utilizado conseguir detetar determinado Trigger não significa que todos também o consigam fazer, especialmente quando a pontuação de qualidade da imagem é baixa.
- Os consumos de memória da Aplicação dependem da quantidade de informação recolhida pela câmara. Quanto mais informação do ambiente for recolhida (ex.: planos e *feature points* detetados) maior será o consumo de memória. Por esta mesma razão durante os testes tentei ao máximo que a câmara do dispositivo apontasse sempre para o mesmo sítio.
- De forma a recolher as estatísticas de uso de memória fui utilizada uma *Dev build* da Aplicação com recurso a um *Profiler*, logo os valores registados não correspondem aos valores finais após otimizações (*Profiler* consome \approx 33MB de memória).

9.1 Triggers

Começando pelos testes de Triggers, foram recolhidos os seguintes dados:

- **Resolução** da imagem.
- **Tamanho** da imagem em MegaBytes.
- **Formato** da imagem.
- **Pontuação** de qualidade da imagem para ser utilizada como Trigger.
- **Reconhecido**: indica se o Trigger foi reconhecido pela Aplicação.
- **Uso de memória** (App): memória que está a ser utilizada pela Aplicação, em MegaBytes.
- **Tempo de processamento** (Backend): tempo que o Backend demorou a processar o Trigger (hh:mm:ss).

Na aplicação, começamos por testar um Trigger de cada vez, sendo sempre associada a mesma Ação de forma a conseguir comparar os resultados obtidos.

Segue-se uma tabela com estes resultados:

Tabela 18- Resultados dos testes de Triggers.

Resolução	Tamanho (MB)	Formato	Pontuação	Reconhecido	Uso de memória da App (MB)	Tempo de Processamento (Backend)
300x300	0.016	jpg	5	✓	79-80	00:01:50
473x709	0.117	jpg	55	✓	80-81	00:01:55
512x512	0.609	png	75	✓	80-81	00:01:54
600x600	0.044	jpg	100	✓	80-81	00:01:55
652x1280	0.154	jpg	0	✓	83-84	00:01:53
900x900	0.245	jpg	100	✓	83-84	00:01:53
1024x681	0.521	jpg	15	✓	83-84	00:01:56
1160x1160	0.061	png	100	✓	85-86	00:01:53
1200x800	0.107	png	90	✓	85-86	00:01:58
1920x1080	0.225	jpg	0	✓	92-93	00:01:57
1920x1080	1.16	jpg	20	✓	92-93	00:01:58
1920x1080	1.92	png	15	✓	92-93	00:02:01
3840x2160	0.852	jpg	35	✓	93-94	00:02:04
7680x4320	6.55	jpg	5	✓	93-94	00:02:08
10315x7049	9.7	jpg	25	✓	96-97	00:02:12

Todos os Triggers utilizados foram reconhecidos pela Aplicação, mesmo os que tinham uma pontuação de qualidade igual ou perto de 0. Os resultados obtidos levam a acreditar que quanto maior a resolução do Trigger maior será o consumo de memória e o tempo de processamento no Backend.

Após verificar os consumos de memória usando um Trigger de cada vez, decidimos testar usar múltiplos em simultâneo. Para isso foram utilizados os 10 últimos Triggers da tabela anterior, começando de baixo para cima. Segue-se a tabela com os resultados obtidos:

Tabela 19 - Uso de memória com múltiplos Triggers em simultâneo.

Nº de Triggers	Uso de memória (MB)
2	109-110
4	135-136
6	159-160
8	172-173
10	182-183

Como era de esperar quanto mais Triggers forem utilizados em simultâneo, maior será o consumo de memória, sendo por isso algo que deve ser tido em conta durante a gestão das Campanhas. O número limite de Triggers usados em simultâneo está dependente da quantidade de memória disponível no dispositivo e na resolução/tamanho dos Triggers utilizados. De salientar que, ao contrário das Ações que necessitam de ser despoletadas para consumir memória, os Triggers estão sempre a consumir, já que a Aplicação necessita de os reconhecer.

9.2 Ações

Tal como nos Triggers, para testar as Ações, começámos por testar uma de cada vez, usando sempre o mesmo Trigger para despoletar cada uma.

Os dados recolhidos dependem do tipo de Ação. Para os vídeos temos:

- **Resolução** do vídeo.
- **Duração** do vídeo. (hh:mm:ss)
- **Tamanho** do vídeo em MegaBytes.
- **Formato** do vídeo.
- **Despoletada**: indica se a Ação foi despoletada / é visível corretamente.
- **Uso de memória (App)**: memória que está a ser utilizada pela Aplicação, em MegaBytes.
- **Tempo de Conversão (Backend-FFmpeg)**: tempo que o Backend-FFmpeg demorou a converter o vídeo para formatos / *codecs* suportados. (hh:mm:ss)
- **Tempo de processamento (Backend-Unity)**: tempo que o Backend-Unity demorou a processar a Ação. (não inclui a o tempo de conversão do ffmpeg) (hh:mm:ss)

Segue-se a tabela com os resultados obtidos para o teste de vídeos:

Tabela 20 - Resultados dos testes de Ações (Vídeos).

Resolução	Duração	Tamanho (MB)	Formato	Despoletada	Uso de Memória da App (MB)	Tempo de Conversão	Tempo de Processamento
1280x720	00:01:58	114	mp4	✓	90-91	00:05:22	00:01:59
1280x720	00:01:14	12.1	mp4	✓	90-91	00:02:06	00:01:55
1920x1080	00:03:10	40.4	mp4	✓	90-91	00:10:59	00:02:00
2160x3840	00:00:43	184	mp4	✓	90-91	00:16:18	00:01:58
360x360	00:01:20	5.03	webm (vp8)	✓	89-90	n.a.	00:05:12
490x476	00:00:04	0.229	webm (vp9)	✓	84-85	00:00:10	00:02:04
1920x1080	00:00:28	1.72	webm (vp9)	✓	89-90	00:03:35	00:08:30
1920x1080	00:00:30	75	webm (vp8)	✓	92-93	n.a.	00:13:33
1920x1080	00:00:30	70.5	webm (vp9)	✓	89-90	00:02:42	00:10:10
2560x1440	00:00:28	2.76	webm (vp9)	✓	89-90	00:04:30	00:13:23

Por norma o tempo de conversão e processamento é maior quanto maior a resolução, duração e tamanho do vídeo, no entanto existiram alguns casos em que isso não se verificou. Quanto ao uso de memória, este foi mais ou menos o mesmo independentemente destes fatores. Uma possível explicação para isto é a conversão do vídeo, que, apesar de não alterar a resolução original, limita outros fatores de qualidade (por ex.: *bitrate*) mantendo os consumos de memória idênticos. Todos os vídeos foram despoletados corretamente.

Para Ações do tipo áudio temos:

- **Duração** do áudio. (hh:mm:ss)
- **Tamanho** do áudio em MegaBytes.
- **Formato** do áudio.
- **Despoletada**: indica se a Ação foi despoletada corretamente.
- **Uso de memória (App)**: memória que está a ser utilizada pela Aplicação, em MegaBytes.
- **Tempo de processamento (Backend-Unity)**: tempo que o Backend demorou a processar a Ação. (hh:mm:ss)

Segue-se a tabela com os resultados obtidos para o teste de áudios:

Tabela 21 - Resultados dos testes de Ações (Áudio).

Duração	Tamanho (MB)	Formato	Despoletada	Uso de Memória da App (MB)	Tempo de Processamento
00:00:42	0.9	mp3	✓	91-92	00:02:20
00:01:00	0.98	mp3	✓	94-95	00:02:08
00:02:13	2.17	mp3	✓	106-107	00:02:19
00:04:13	4	mp3	✓	126 -127	00:02:32

Todos os áudios foram reproduzidos/despoletados corretamente. Ao contrário dos vídeos, quanto maior a duração/tamanho maior o consumo de memória. Estes valores de consumo parecem estar um pouco elevados e talvez seja necessário, futuramente, otimizar o processamento de áudio no Backend de forma a diminuir este consumo. Quanto aos tempos de processamento, estes foram mais ou menos semelhantes independentemente do tamanho ou duração.

Para Ações do tipo modelos 3D temos:

- Número de **Triângulos** pelos quais o modelo é composto.
- **Tamanho** do modelo em Megabytes.
- Se possui ou não **Animações**.
- **Despoletada**: indica se a Ação foi despoletada corretamente.
- **Uso de memória (App)**: memória que está a ser utilizada pela Aplicação, em MegaBytes.
- **Tempo de processamento** (Backend-Unity): tempo que o Backend demorou a processar a Ação. (hh:mm:ss)

Tabela 22 - Resultados dos testes de Ações (Modelos 3D).

Triângulos	Tamanho(MB)	Animação?	Despoletada	Uso de Memória da App (MB)	Tempo de Processamento
12 128	12.5	✓	✓	139-140	00:02:42
17 490	24.8	✗	✓	140-141	00:03:08
28 106	22.4	✓	✓	141-142	00:03:20
241 632	27	✗	✓	181-182	00:04:37

Todos os modelos e respetivas animações foram despoletadas corretamente. Pelos resultados obtidos acreditamos que quanto maior a complexidade do modelo maior consumo de memória e mais tempo demora o processamento no Backend. No entanto os consumos de memória estão um pouco elevados, sendo por isso necessário, futuramente, rever o processamento dos modelos de forma a diminuir estes valores.

Por último para Ações do tipo imagem, temos:

- **Resolução** da imagem
- **Tamanho** da imagem em Megabytes
- **Formato** da imagem
- **Despoletada**: indica se a Ação foi despoletada corretamente
- **Uso de memória (App)**: memória que está a ser utilizada pela Aplicação, em MegaBytes
- **Tempo de processamento** (Backend-Unity): tempo que o Backend demorou a processar a Ação. (hh:mm:ss)

Tabela 23 - Resultados dos testes de Ações (Imagens).

Resolução	Tamanho (MB)	Formato	Despoletada	Uso de Memória da App (MB)	Tempo de Processamento
1920x1228	0.290	jpg	✓	97-98	00:02:07
1920x5652	2.29	png	✓	94-95	00:02:12
3840x2160	1.1	jpg	✓	97-98	00:02:13
5000x3290	10	png	✓	104-105	00:02:16

Todas as imagens foram despoletadas corretamente. Tal como nos Triggers quanto maior a resolução da imagem maior será o consumo de memória e o tempo de processamento.

Capítulo 10

Conclusão e Oportunidades Futuras

É já tradição a organização de certas iniciativas, por parte da Critical Software, de forma a promover a interação com os seus funcionários. Seja para receber novos funcionários, ou para celebrar determinadas épocas especiais ao longo do ano, são organizadas diversas atividades, ou oferecidas algumas surpresas que simbolizam a ocasião. De forma a melhorar ou expandir as possibilidades destas iniciativas surge a realidade aumentada. Ao apontarem câmara dos seus dispositivos móveis para determinados conteúdos ou estímulos, os funcionários serão capazes de visualizar a experiência de realidade aumentada preparada pela equipa responsável pela iniciativa. Para isso, é necessário criar um sistema que possibilite: visualizar o conteúdo de realidade aumentada através de uma aplicação móvel, definir quais os elementos servirão como estímulo para despoletar esse conteúdo, e, por último, que permita definir e customizar ao máximo o conteúdo que será mostrado. Um aspeto considerado essencial é a importância de não ser preciso atualizar a aplicação com uma nova *build* quando se pretende fazer alterações a este conteúdo, e ao mesmo tempo ser capaz de garantir a liberdade necessária para que, a equipa responsável, consiga criar as experiências que desejar, sem requerer qualquer tipo de conhecimento sobre realidade aumentada ou sobre o funcionamento do sistema.

O projeto iniciou-se pelo estudo do estado da arte de realidade aumentada. Começámos por analisar a sua história e possíveis aplicações, de forma a perceber o estado atual da tecnologia, como tem evoluído e onde está a ser utilizada. Foi analisada com maior detalhe a utilização de realidade aumentada para marketing, percebendo de que forma estão a aproveitar a tecnologia para atrair novos clientes ou reforçar a relação com clientes existentes. De seguida, foi feita uma análise das limitações da tecnologia, de forma a perceber os cuidados que se devem ter, e possíveis desafios que poderão ser encontrados. Destes desafios destaca-se o facto de ser difícil de alcançar o realismo de elementos virtuais, principalmente em dispositivos móveis, devido à grande variedade de hardware existente. Outra limitação encontrada foi o facto de não existir uma maneira bem definida de interagir com os elementos virtuais. Através do estudo do futuro da realidade aumentada percebemos que, esta interação é uma das áreas que se encontra ainda a ser investigada, assim como, a constante evolução de técnicas e funcionalidades de forma a alcançar um maior nível de realismo. Posteriormente foram estudadas, as várias abordagens para a aplicação de realidade aumentada, os vários conceitos associados à tecnologia e as ferramentas de desenvolvimento existentes.

Subsequentemente a este estudo foi definida uma especificação inicial dos requisitos funcionais. Através desta especificação e juntamente com o estudo realizado, percebemos que a ferramenta AR Foundation seria a ideal para cumprir com os objetivos propostos. A escolha foi feita com base, na possibilidade de desenvolver para ambas as plataformas (android e iOS) em simultâneo, na quantidade de dispositivos suportados, no preço, e por último, nas funcionalidades de *Image Tracking* fornecidas. De forma a confirmar esta escolha e de investigar qual seria a melhor abordagem para garantir uma implementação flexível, com a capacidade de utilizar novos os conteúdos sem a necessidade de uma nova compilação da aplicação, decidimos avançar com o desenvolvimento de um protótipo. Para além de verificar o referido, o protótipo permitiu também a familiarização, não só com as tecnologias escolhidas, mas também com as

ferramentas (BitBucket e Jira) e os processos utilizados pela Critical Software, bem como a metodologia de desenvolvimento escolhida para o projeto, Agile Kanban.

Após o desenvolvimento do protótipo foi definida e implementada a solução arquitetural para o sistema, sendo esta composta por 3 componentes: **Aplicação móvel** que permite visualizar as experiências de realidade aumentada definidas, tirando partido de várias funcionalidades disponibilizadas pela ferramenta de desenvolvimento escolhida. **Backend**, sendo o coração de todo o sistema, garante a criação customização e modificabilidade das experiências de realidade aumentada. É também a componente responsável por fazer o processamento do conteúdo para que este possa ser utilizado pela aplicação sem haver a necessidade uma nova compilação da mesma. **Interface de Gestão do Conteúdo**, que torna simples e intuitiva a comunicação com o Backend para criar e gerir as experiências de realidade aumentada desejadas, não exigindo da parte dos utilizadores qualquer nível de conhecimento sobre as tecnologias que estão a ser utilizadas pelo sistema.

Por último, para além dos vários testes unitários e de integração desenvolvidos ao longo do todo o projeto, foram realizados imensos testes (*end-to-end*) de forma a verificar e validar o comportamento do sistema como um todo.

Desta forma, foi construído um sistema que possibilita a criação de experiências de realidade aumentada, por qualquer pessoa, independentemente do seu nível de conhecimento, utilizando qualquer conteúdo desejado sem que haja a necessidade de fazer alterações na Aplicação móvel, cumprindo assim com os objetivos propostos para este estágio. Para além disso, o sistema está implementado de forma a ser possível tirar partido de inúmeras oportunidades futuras que tornam a solução ainda mais completa. Na Aplicação móvel, com a evolução dos dispositivos móveis, cada vez mais serão suportadas certas funcionalidades capazes de aumentar o realismo e a imersão da experiência, que poderão ser adotadas para o nosso sistema. Podemos também expandir as propriedades de customização das experiências e aumentar os tipos e formatos de ficheiros aceites pelo Backend. Em vez de usar apenas imagens como estímulos, usar também objetos para despoletar as experiências. Ser possível definir uma sequência de animações para os modelos 3D. Tornar as experiências personalizadas/individualizadas de acordo com o utilizador que as está a visualizar, ou permitir customizar de acordo com a hora do dia, estado do tempo, entre outros.

Referências

- [1] B. Poetker, “A Brief History of Augmented Reality (+Future Trends & Impact),” *g2*, 2019. <https://learn.g2.com/history-of-augmented-reality> (accessed Oct. 23, 2020).
- [2] G. Jha, P. Singh, and L. Sharma, “Recent advancements of augmented reality in real time applications,” *Int. J. Recent Technol. Eng.*, vol. 8, no. 2 Special Issue 7, pp. 537–542, Jul. 2019, doi: 10.35940/ijrte.B10100.0782S719.
- [3] M. A. Sánchez-Acevedo, B. A. Sabino-Moxo, and J. A. Márquez-Domínguez, “Mobile augmented reality: Evolving human-computer interaction,” in *Virtual and Augmented Reality: Concepts, Methodologies, Tools, and Applications*, vol. 1, IGI Global, 2018, pp. 200–221.
- [4] J. Peddie, *Augmented Reality Where We Will All Live*, 1st ed. Springer, 2017.
- [5] Myron Krueger, “Myron Krueger,” *the digital age*, 2010. [http://thedigitalage.pbworks.com/w/page/22039083/Myron Krueger](http://thedigitalage.pbworks.com/w/page/22039083/Myron%20Krueger) (accessed Oct. 24, 2020).
- [6] M. W. Krueger, “‘VIDEOPLACE’: A Report from the ARTIFICIAL REALITY Laboratory,” *Leonardo*, vol. 18, no. 3, p. 145, 1985, doi: 10.2307/1578043.
- [7] L. B. Rosenberg, “Virtual fixtures: perceptual tools for telerobotic manipulation,” in *1993 IEEE Annual Virtual Reality International Symposium*, 1993, pp. 76–82, doi: 10.1109/vrais.1993.380795.
- [8] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster, “A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment,” *Pers. Ubiquitous Comput.*, vol. 1, no. 4, pp. 208–217, 1997, doi: 10.1007/BF01682023.
- [9] Wikipedia, “1st & Ten (graphics system) - Wikipedia,” 2020. [https://en.wikipedia.org/wiki/1st_%26_Ten_\(graphics_system\)](https://en.wikipedia.org/wiki/1st_%26_Ten_(graphics_system)) (accessed Oct. 31, 2020).
- [10] “ARQuake - Wearable Computer Lab.” <http://www.tinmith.net/arquake/> (accessed Oct. 26, 2020).
- [11] M. N. Billinghurst and A. Henrysson, “Research Directions in Handheld AR,” *Int. J. Virtual Real.*, vol. 5, no. 2, pp. 51–58, Jan. 2006, doi: 10.20870/ijvr.2006.5.2.2690.
- [12] P. Strauss, “Mini Augmented Reality Ads Hit Newstands,” *17 december*, 2008. <https://technabob.com/blog/2008/12/17/mini-augmented-reality-ads-hit-newstands/> (accessed Oct. 26, 2020).
- [13] Wikipedia, “Google Glass Wikipedia,” 2013. https://en.wikipedia.org/wiki/Google_Glass (accessed Oct. 26, 2020).
- [14] Microsoft Corporation, “Hardware do HoloLens (1ª gen) | Microsoft Docs,” *Microsoft Docs*, 2019. <https://docs.microsoft.com/pt-br/hololens/hololens1-hardware> (accessed Oct. 26, 2020).
- [15] Microsoft Corporation, “HoloLens 2—Overview, Features, and Specs | Microsoft HoloLens,” 2019. <https://www.microsoft.com/en-us/hololens/hardware> (accessed Oct. 26, 2020).

- [16] Wikitude, “2017 Augmented Reality Year in Review | Wikitude Blog,” *Wikitude*, 2017. <https://www.wikitude.com/blog-2017-augmented-reality-year-in-review/> (accessed Jan. 12, 2021).
- [17] Microsoft Portugal News Center, “Uma nova realidade: Microsoft anuncia chegada de HoloLens a Portugal - News Center,” 2020. <https://news.microsoft.com/pt-pt/2020/10/20/uma-nova-realidade-microsoft-anuncia-chegada-de-hololens-a-portugal/> (accessed Oct. 26, 2020).
- [18] Microsoft reporter, “Surgeons are using HoloLens to ‘see inside’ patients before they operate on them,” *Microsoft*, 2018. <https://news.microsoft.com/en-gb/2018/02/08/surgeons-use-microsoft-hololens-to-see-inside-patients-before-they-operate-on-them/> (accessed Oct. 26, 2020).
- [19] W. S. Khor, B. Baker, K. Amin, A. Chan, K. Patel, and J. Wong, “Augmented and virtual reality in surgery-the digital surgical environment: Applications, limitations and legal pitfalls,” *Ann. Transl. Med.*, vol. 4, no. 23, 2016, doi: 10.21037/atm.2016.12.23.
- [20] AccuVein, “AccuVein Vein Visualization-Improves IV 1-Stick Success 3.5x,” 2020. <https://www.accuvein.com/> (accessed Oct. 26, 2020).
- [21] M. Morozov, “The Applications of AR in Military — Jasoren,” *Jasoren*. <https://jasoren.com/augmented-reality-military/> (accessed Oct. 26, 2020).
- [22] T. Khan, K. Johnston, and J. Ophoff, “The Impact of an Augmented Reality Application on Learning Motivation of Students,” *Adv. Human-Computer Interact.*, vol. 2019, 2019, doi: 10.1155/2019/7208494.
- [23] Wikitude GmbH, “Wikitude Augmented Reality: the World’s Leading Cross-Platform AR SDK,” 2020. <https://www.wikitude.com> (accessed Oct. 26, 2020).
- [24] Wikitude, “Multiple Object Tracking - Wikitude Augmented Reality - YouTube,” 2020. <https://www.youtube.com/watch?v=iak9zBhhoR4> (accessed Oct. 26, 2020).
- [25] M. Y. C. Yim, S. C. Chu, and P. L. Sauer, “Is Augmented Reality Technology an Effective Tool for E-commerce? An Interactivity and Vividness Perspective,” *J. Interact. Mark.*, vol. 39, pp. 89–103, Aug. 2017, doi: 10.1016/j.intmar.2017.04.001.
- [26] IKEA, “Aplicações móveis - IKEA,” 2020. <https://www.ikea.com/pt/pt/customer-service/mobile-apps/> (accessed Oct. 26, 2020).
- [27] Nike, “Nike Fit Digital Foot Measurement Tool,” 2019. <https://news.nike.com/news/nike-fit-digital-foot-measurement-tool> (accessed Oct. 26, 2020).
- [28] Warby Parker, “Mobile App | Warby Parker.” <https://www.warbyparker.com/app> (accessed Oct. 26, 2020).
- [29] Niantic, “RA+ já disponível em Pokémon GO! - Pokémon GO,” 2017. https://pokemongolive.com/pt_br/post/arplus/ (accessed Oct. 26, 2020).
- [30] T. Stone, “Hands-on with Minecraft Earth at E3 | Minecraft,” 2019. <https://www.minecraft.net/en-us/article/hands-on-minecraft-earth-e3> (accessed Oct. 26, 2020).
- [31] Thyssenkrupp, “thyssenkrupp Elevator – bringing new vision to elevator maintenance with Microsoft HoloLens - YouTube,” 2019. <https://www.youtube.com/watch?v=biNebig1gUI> (accessed Oct. 26, 2020).

- [32] Upskill, "Boeing Augmented Reality Secret: Skylight For Assembly," 2019. <https://upskill.io/landing/upskill-and-boeing/> (accessed Oct. 26, 2020).
- [33] Porsche AG, "Porsche sets new quality benchmarks with its Inno-Space," 2018, Accessed: Oct. 26, 2020. [Online]. Available: <https://newsroom.porsche.com/en/company/porsche-inno-space-quality-innovation-plant-leipzig-augmented-reality-production-it-albrecht-reimold-14815.html>.
- [34] C. Moldrich, "Welcome to the Inno-Space: inside Porsche's tech powerhouse | CAR Magazine," 2018. <https://www.carmagazine.co.uk/car-news/tech/porsche-inno-space-inside-the-leipzigs-tech-innovation-project/> (accessed Oct. 26, 2020).
- [35] Munesh Kumar Sharma, "Augmented Reality Navigation," *Int. J. Eng. Res.*, vol. V9, no. 06, Jun. 2020, doi: 10.17577/ijertv9is060441.
- [36] Envisics, "Envisics | LinkedIn," *LinkedIn*. <https://www.linkedin.com/company/envisics-limited> (accessed Jan. 12, 2021).
- [37] Andrew Makarov, "How Augmented Reality Indoor Navigation Works - MobiDev," *MobiDev Blog*. <https://mobidev.biz/blog/augmented-reality-indoor-navigation-app-developement-arkit> (accessed Oct. 26, 2020).
- [38] ViewAR, "Augmented Reality Indoor Navigation - GuideBOT." <https://guidebot.viewar.com/> (accessed Oct. 26, 2020).
- [39] Wikitude, "7 ways to use augmented reality in marketing today - Wikitude." <https://www.wikitude.com/blog-7-ways-to-use-augmented-reality-in-marketing-today/> (accessed Jan. 01, 2021).
- [40] J. Beam, "Augmented Reality | Jim Beam® Since 1795." <https://www.jimbeam.com/en/behind-the-bourbon/app> (accessed Jan. 01, 2021).
- [41] Philadelphia Insectarium & Butterfly Pavilion, "Get Our Insect AR App | Philadelphia Insectarium & Butterfly Pavilion." <https://www.phillybutterflypavilion.com/insect-ar-app-download/> (accessed Jan. 01, 2021).
- [42] F. J. Delgado, M. F. Abernathy, J. White, and W. H. Lowrey, "Real-time 3D flight guidance with terrain for the X-38," in *Enhanced and Synthetic Vision 1999*, Jul. 1999, vol. 3691, pp. 149–156, doi: 10.1117/12.354416.
- [43] Google AR & VR, "Introduction to Augmented Reality and ARCore," *Coursera*. <https://www.coursera.org/learn/ar> (accessed Nov. 10, 2020).
- [44] Qualcomm, "Wireless Technology Innovation | Mobile Technology | Qualcomm," *Qualcomm*, 2020. <https://www.qualcomm.com/> (accessed Oct. 26, 2020).
- [45] Qualcomm Technologies Inc., "The Mobile Future of eXtended Reality (XR)," *Qualcomm Media*, 2017, [Online]. Available: <https://www.qualcomm.com/media/documents/files/the-mobile-future-of-extended-reality-xr.pdf>.
- [46] O. Guler and I. Yucedag, "Developing an CNC lathe augmented reality application for industrial maintenance training," Dec. 2018, doi: 10.1109/ISMSIT.2018.8567255.
- [47] J. C. P. Cheng, K. Chen, and W. Chen, "Comparison of Marker-Based and Markerless AR: A Case Study of An Indoor Decoration System," Jul. 2017, pp. 483–490, doi: 10.24928/jc3-2017/0231.

- [48] Google Developers, "ARCore Overview | ARCore | Google Developers," *Developers.Google.Com*, 2018. <https://developers.google.com/ar/discover/> (accessed Jan. 01, 2021).
- [49] Google Developers, "ARCore supported devices | Google Developers," *ARCore supported devices*, 2020. <https://developers.google.com/ar/discover/supported-devices> (accessed Jan. 01, 2021).
- [50] Google Developers, "Depth API overview for Unity | ARCore | Google Developers," *Google Developers*, 2020. <https://developers.google.com/ar/develop/unity/depth/overview> (accessed Jan. 01, 2021).
- [51] E. Lai, "Google Developers Blog: Improving shared AR experiences with Cloud Anchors in ARCore 1.20," *Google Developers*, 2020. <https://developers.googleblog.com/2020/10/improving-shared-ar-experiences-with.html> (accessed Jan. 01, 2021).
- [52] Apple, "ARKit: Apple Developer Documentation," *Apple Developer*, 2020. <https://developer.apple.com/documentation/arkit> (accessed Jan. 01, 2021).
- [53] Apple, "Tracking and Altering Images | Apple Developer Documentation," *Apple Developer*. https://developer.apple.com/documentation/arkit/tracking_and_altering_images (accessed Jan. 01, 2021).
- [54] Apple, "Scanning and Detecting 3D Objects," *Apple Developer*, 2020. https://developer.apple.com/documentation/arkit/scanning_and_detecting_3d_objects (accessed Jan. 01, 2021).
- [55] Apple, "Occluding Virtual Content with People | Apple Developer Documentation," *Apple Developer*, 2020. https://developer.apple.com/documentation/arkit/occluding_virtual_content_with_people (accessed Jan. 01, 2021).
- [56] Apple, "AR / VR - Videos - Apple Developer," *Apple Developer*. <https://developer.apple.com/videos/graphics-and-games/ar-vr/> (accessed Jan. 01, 2021).
- [57] Unity, "About AR Foundation," 2020. <https://docs.unity3d.com/Packages/com.unity.xr.arfoundation@4.0/manual/index.html> (accessed Jan. 01, 2021).
- [58] Vuforia, "Getting Started | VuforiaLibrary," *Vuforia Library*, 2020. <https://library.vuforia.com/> (accessed Jan. 01, 2021).
- [59] Vuforia, "Multi Targets | VuforiaLibrary," *Vuforia Library*, 2020. <https://library.vuforia.com/features/images/multi-target.html> (accessed Jan. 01, 2021).
- [60] Vuforia, "Cylinder Targets," *Vuforia Library*, 2020. <https://library.vuforia.com/features/images/cylinder-targets.html> (accessed Jan. 01, 2021).
- [61] Vuforia, "VuMark | VuforiaLibrary," *Vuforia Library*, 2020. <https://library.vuforia.com/features/objects/vumark.html> (accessed Jan. 01, 2021).
- [62] Vuforia, "Vuforia Fusion | VuforiaLibrary," *Vuforia Library*, 2020.

- <https://library.vuforia.com/articles/Training/vuforia-fusion-article.html> (accessed Jan. 01, 2021).
- [63] Vuforia, “Vuforia Engine pricing | PTC,” *Wikitude*, 2020. <https://www.ptc.com/en/products/vuforia/vuforia-engine/pricing> (accessed Jan. 01, 2021).
- [64] Wikitude, “Wikitude SDK Documentation,” *Wikitude*, 2020. <https://www.wikitude.com/documentation/> (accessed Jan. 01, 2021).
- [65] Wikitude, “Getting started Wikitude SDK Unity,” *Wikitude*, 2020. <https://www.wikitude.com/external/doc/documentation/latest/unity/gettingstartedunity.html#recommended-usage-of-this-documentation> (accessed Jan. 01, 2021).
- [66] Wikitude, “Supported Devices Wikitude SDK Unity 9.5.0 Documentation,” *Wikitude*, 2020. https://www.wikitude.com/external/doc/documentation/latest/unity/supporteddevice_sunity.html (accessed Jan. 01, 2021).
- [67] Wikitude, “Wikitude Store: Find Best Pricing for your Augmented Reality Experiences.,” *Wikitude*, 2020. <https://www.wikitude.com/store/> (accessed Jan. 01, 2021).
- [68] F. Herpich, R. L. M. Guarese, and L. M. R. Tarouco, “A Comparative Analysis of Augmented Reality Frameworks Aimed at the Development of Educational Applications,” *Creat. Educ.*, vol. 08, no. 09, pp. 1433–1451, 2017, doi: 10.4236/ce.2017.89101.
- [69] D. Amin and S. Govilkar, “Comparative Study of Augmented Reality Sdk’s,” *Int. J. Comput. Sci. Appl.*, vol. 5, no. 1, pp. 11–26, Feb. 2015, doi: 10.5121/ijcsa.2015.5102.
- [70] A. Hanafi, L. Elaachak, and M. Bouhorma, “A comparative study of augmented reality SDKs to develop an educational application in chemical field,” in *ACM International Conference Proceeding Series*, 2019, vol. Part F1481, doi: 10.1145/3320326.3320386.
- [71] Atlassian Agile Coach, “What is Agile? | Atlassian,” *Atlassian*, 2020. <https://www.atlassian.com/agile> (accessed Jan. 12, 2021).
- [72] Wikipedia, “Dependency inversion principle - Wikipedia,” 2021. https://en.wikipedia.org/wiki/Dependency_inversion_principle (accessed Jul. 21, 2021).
- [73] Unity Documentation, “Unity - Manual: Video file compatibility.” <https://docs.unity3d.com/Manual/VideoSources-FileCompatibility.html> (accessed Jul. 30, 2021).

Apêndices

Apêndice A – Requisitos Funcionais do Backend e da Interface de Gestão do Conteúdo, e respectivas prioridades.

Requisitos Funcionais				
Componente	ID	Nome	Descrição	Prioridade
Backend + Interface de Gestão do Conteúdo	RF-1	Adicionar novos Triggers	Através destas Componentes, qualquer Gestor deve ser capaz de adicionar novos Triggers.	Must
	RF-2	Remover Triggers	Através destas Componentes, qualquer Gestor deve ser capaz de remover Triggers previamente adicionados.	Must
	RF-3	Avaliar uma imagem	Através destas Componentes, qualquer Gestor deve ser capaz de avaliar se uma imagem pode ser utilizada como Trigger e se possui boa qualidade para	Should
	RF-4	Adicionar novas Ações	Através destas Componentes, qualquer Gestor deve ser capaz de adicionar novas Ações.	Must
	RF-5	Remover Ações	Através destas Componentes, qualquer Gestor deve ser capaz de remover Ações previamente adicionadas.	Must
	RF-6	Criar Campanhas	Através destas Componentes, qualquer Gestor deve ser capaz de criar Campanhas com uma data de início e de fim.	Must
	RF-7	Editar Campanhas	Através destas Componentes, qualquer Gestor deve ser capaz de editar as Campanhas.	Should
	RF-8	Remover Campanhas	Através destas Componentes, qualquer Gestor deve ser capaz de remover Campanhas criadas (mesmo estando ativas).	Must
	RF-9	Associar Triggers a Campanhas	Através destas Componentes, qualquer Gestor deve ser capaz de associar um ou mais Triggers a Campanhas existentes.	Must
	RF-10	Associar Ações a Triggers	Através destas Componentes, qualquer Gestor deve ser capaz de associar uma ou mais Ações a Triggers existentes numa determinada Campanha.	Must
	RF-11	Remover um Trigger de uma Campanha	Através destas Componentes, qualquer Gestor deve ser capaz de remover um Trigger de uma Campanha.	Must
	RF-12	Remover uma Ação associada a um Trigger	Através destas Componentes, qualquer Gestor deve ser capaz de remover uma Ação de um Trigger existente numa determinada Campanha.	Must
	RF-13	Ações customizáveis	As Ações devem possuir vários parâmetros de customização.	Must
	RF-14	Alterar a customização de cada Ação	Através destas Componentes, qualquer Gestor deve ser capaz de alterar os vários parâmetros de customização de cada Ação.	Must
	RF-15	Customizações diferentes para a mesma Ação	A mesma Ação quando associada a Triggers diferentes pode ter parâmetros de customização diferentes.	Should
	RF-16	Obter todas as Campanhas ativas	Através do Backend deve ser possível obter todas as Campanhas ativas, e fazer o download dos respetivos Triggers e Ações.	Must
	RF-17	Testar Campanha	Qualquer Gestor deve ser capaz de testar as Campanhas criadas antes de as tornar públicas.	Could
	RF-18	Processar conteúdo	O Backend deve ser capaz de preparar / processar o conteúdo (Triggers e Ações) para que este possa ser utilizado pela aplicação.	Must

Apêndice B – Requisitos Funcionais da Aplicação móvel, e respetivas prioridades.

Requisitos Funcionais				
Componente	ID	Nome	Descrição	Prioridade
Aplicação Móvel	RF-19	Atualizar Campanhas	Aplicação deve ser capaz de obter as novas Campanhas ou atualizar existentes sem necessitar de uma nova <i>build</i> .	Must
	RF-20	Reconhecimento de Triggers	Aplicação deve ser capaz de reconhecer Triggers através da imagem recolhida pela câmara do dispositivo.	Must
	RF-21	Despoletar Ações	Ao reconhecer um Trigger a aplicação deve despoletar as respetivas Ações com os parâmetros de customização definidos.	Must
	RF-22	Tirar fotografia	Através da aplicação deve ser possível tirar fotografias que incluam o conteúdo de realidade aumentada gerado.	Could
	RF-23	Gravar vídeo	Através da aplicação deve ser possível gravar vídeos que incluam o conteúdo de realidade aumentado gerado.	Could

Apêndice C – Análise de Riscos

Análise de Riscos								
Categoria	ID	Risco	Probabilidade	Severidade	Consequências	Para diminuir a probabilidade	Para diminuir a severidade	Para monitorizar
Riscos Técnicos	1	Alterações de requisitos numa fase avançada do projeto	Baixa	Média	Poderá exigir alterações sobre funcionalidades que já foram implementadas, podendo atrasar o projeto, ou não existir tempo para aplicar estas alterações.	Realizar as reuniões planeadas, mantendo assim uma comunicação constante do que está a ser desenvolvido. Adotar uma metodologia de desenvolvimento adequada face à probabilidade de ocorrência de alterações de requisitos.	Fazer a análise das implicações impostas por estas alterações e possíveis conflitos que possam criar. Tendo em conta esta análise, fazer as alterações necessárias, caso factíveis, ou abandonar os requisitos com menor prioridade se necessário.	Manter uma constante comunicação com os orientadores sobre as tarefas que estão a ser desempenhadas.
	2	Solução proposta não permite cumprir com todos os requisitos ou possíveis alterações dos mesmos	Baixa	Muito-Alta	Não cumprimento de todos os objetivos / requisitos propostos para o sistema.	Fazer um bom estudo e planeamento do sistema tendo em conta possíveis oportunidades futuras. Desenvolver um sistema que tenha em conta estas oportunidades.	Exige reavaliar a arquitetura proposta para o sistema face aos requisitos existentes, e fazer as alterações necessárias caso sejam possíveis, tendo em conta o que já foi feito e o orçamento de tempo disponível.	Manter uma constante comunicação com os orientadores, não só sobre as tarefas que estão a ser desempenhadas, mas também possíveis oportunidades futuras que possam ser aplicadas e devem ser tidas em conta no desenho da arquitetura.
	3	Conflitos entre requisitos	Média	Média	Podem gerar quebras no desenvolvimento, perdendo tempo a resolver estes conflitos.	Descrever os requisitos de uma forma concisa e clara para que não existam dúvidas ao longo do desenvolvimento. Manter uma comunicação constante com os orientadores.	Fazer uma análise e reestruturação destes requisitos tendo em conta os conflitos encontrados.	Manter uma constante comunicação com os orientadores sobre os requisitos e as tarefas / <i>user stories</i> criadas a partir destes.
	4	Fraca ou incompleta documentação	Baixa	Média	Dificuldade de perceber o trabalho que foi feito, assim como as decisões tomadas ao longo do desenvolvimento.	Juntamente com os <i>code reviews</i> existir também uma análise da documentação produzida para cada tarefa/ <i>user story</i> .	Adicionar uma nova tarefa / <i>user story</i> com o objetivo de complementar a documentação necessária.	Juntamente com os <i>code reviews</i> , existir também uma análise da documentação produzida.
	5	Recursos/Servidores disponibilizados para o Backend não são suficientes	Baixa	Média	Incumprimento dos atributos de qualidade definidos.	Planear a arquitetura do sistema tendo em conta o âmbito no qual o sistema será utilizado (apenas por funcionários da Critical +/- 1000)	Reestruturar o sistema para que existam várias instâncias das componentes do Backend. Aumentar os recursos da máquina utilizada.	Utilizar as ferramentas de monitorização providenciadas pelo serviço Microsoft Azure.
	6	Dependência de sistemas externos	Baixa	Muito-Alta	Na ocorrência de problemas ou alterações nestes sistemas externos, a nossa solução poderá ser afetada exigindo alterações de acordo com o sucedido		Diminuir ao máximo as dependências aos sistemas externos que estão a ser utilizados	Monitorizar possíveis alterações nestes sistemas externos.
Riscos de Recursos	7	Falta de acesso a outros dispositivos / smartphones para testes	Média	Alta	Impossibilidade de testar a aplicação em várias plataformas e hardware. Tratando-se de uma aplicação de realidade aumentada, requer o uso da câmara, não sendo possível testar com emuladores.	Constante comunicação com os orientadores alertando para a necessidade destes dispositivos.	Tirar partido dos dispositivos de familiares e amigos para realizar alguns testes.	Comunicação com os orientadores
	8	Falta de acesso a um Mac	Média	Muito-Alta	Impossibilidade de compilar a aplicação para plataformas iOS, uma vez que é necessário acesso ao programa Xcode	Constante comunicação com os orientadores alertando para a necessidade de um dispositivo que suporte o Xcode.		Comunicação com os orientadores
Riscos da Gestão do Projeto	9	Impossibilidade de realizar reuniões de gestão do Kanban Board (terças e quartas)	Baixa	Baixa	Não ocorre a revisão do Kanban Board juntamente com os orientadores da Critical, podendo não existir tarefas a desempenhar bem definidas e provocando atrasos. Pode ainda provocar o risco 1.	Constante comunicação com os orientadores	Caso não existam tarefas planeadas, terá de ser realizada uma reunião o mais cedo possível. O estagiário poderá também definir sozinho as tarefas/ <i>user stories</i> que irá desempenhar, a partir dos requisitos.	Comunicação com os orientadores
	10	Erros de estimação e possíveis atrasos no desenvolvimento	Média	Alta	Poderá levar à propagação de atrasos e a um possível incumprimento da entrega final (Risco - 12) ou abandono de requisitos com uma menor prioridade.	Fazer estimações e planeamento juntamente com os orientadores.	Será necessária a reformulação das estimativas de acordo com estes atrasos e reestruturar o planeamento adequadamente.	Acompanhar o planeamento/estimativas de forma a verificar possíveis atrasos tendo em conta o <i>backlog</i> de tarefas por realizar.
	11	Dependência entre tarefas / <i>user stories</i>	Média	Baixa	Interrompe o fluxo de trabalho podendo criar atrasos. Exige colocar temporariamente uma tarefa em pausa enquanto se terminam outras necessárias.	Definir quais as próximas tarefas a desempenhar tendo em conta possíveis dependências entre as componentes do sistema.	Adaptar Kanban Board de acordo com as necessidades.	Tirar partido das funcionalidades disponibilizadas pela ferramenta Jira e o seu Kanban Board.
	12	Não cumprir com a entrega final do estágio estipulada	Média	Média	Pressão, sobre o estagiário. Incumprimento de determinados requisitos ou objetivos do sistema.	Cumprir com o planeamento definido, assim como, com os processos associados à metodologia de desenvolvimento escolhida.	Recorrer à época especial.	Acompanhar as estimativas de forma a verificar possíveis atrasos tendo em conta o <i>backlog</i> de tarefas por realizar.
	13	Surgimento de novas tarefas não planeadas	Média	Média	Poderá criar atrasos no projeto.	Constante comunicação com os orientadores	Exige reestruturar o planeamento adequadamente.	Comunicação com os orientadores

