

Instituto Superior de Engenharia do Porto

Aplicações Web Multimédia com Realidade Aumentada

“Magic Catalog”

Lino Manuel Cunha Pereira

Dissertação para obtenção do grau de Mestre em
Engenharia Informática

Área de especialização em
Sistema Gráficos e Multimédia

Orientador: Prof. Doutor Carlos Vaz de Carvalho do Departamento de Engenharia Informática do Instituto Superior de Engenharia do Porto

Co-orientador: Luís Castro, CEO da empresa Front-end Multimédia, Lda.

Júri:

Presidente:

João Paulo Jorge Pereira, Professor Adjunto, ISEP

Vogais:

António Fernando Vasconcelos Cunha Castro Coelho, Professor Auxiliar, FEUP

Carlos Miguel Miranda Vaz de Carvalho, Professor Adjunto, ISEP

Luís Miguel Marçal Vilaça Castro, CEO da empresa Front-end Multimédia

Porto, Outubro 2011

I. Resumo

A Realidade Aumentada veio alterar a percepção que o ser humano tem do mundo real. A expansão da nossa realidade à Realidade Virtual possibilita a criação de novas experiências, cuja aplicabilidade é já tida como natural em diversas situações. No entanto, potenciar este tipo de interacção pode ser um processo complexo, quer por limitações tecnológicas, quer pela gestão dos recursos envolvidos. O desenvolvimento de projectos com realidade aumentada para fins comerciais passa assim muitas vezes pela optimização dos recursos utilizados tendo em consideração as limitações das tecnologias envolventes (sistemas de detecção de movimento e voz, detecção de padrões, GPS, análise de imagens, sensores biométricos, etc.).

Com a vulgarização e aceitação das técnicas de Realidade Aumentada em muitas áreas (medicina, educação, lazer, etc.), torna-se também necessário que estas técnicas sejam transversais aos dispositivos que utilizamos diariamente (computadores, tablets, telemóveis etc.). Um dominador comum entre estes dispositivos é a internet uma vez que as aplicações *online* conseguem abarcar um maior número de pessoas.

O objectivo deste projecto era o de criar uma aplicação *web* com técnicas de Realidade Aumentada e cujos conteúdos fossem geridos pelos utilizadores. O processo de investigação e desenvolvimento deste trabalho passou assim por uma fase fundamental de prototipagem para seleccionar as tecnologias que melhor se enquadravam no tipo de arquitectura pretendida para a aplicação e nas ferramentas de desenvolvimento utilizadas pela empresa onde o projecto foi desenvolvido.

A aplicação final é composta por um *FrontOffice*, responsável por mostrar e interpretar as aplicações criadas e possibilitar a integração com outras aplicações, e um *BackOffice* que possibilita aos utilizadores, sem conhecimentos de programação, criar novas aplicações de realidade aumentada e gerir os conteúdos multimédia utilizados.

A aplicação desenvolvida pode servir de base para outras aplicações e ser reutilizável noutros âmbitos, sempre com o objectivo de reduzir custos de desenvolvimento e de gestão de conteúdos, proporcionando assim a implementação de uma Framework que permite a gestão de conteúdos em diferentes áreas (medicina, educação, lazer, etc.), onde os utilizadores podem criar as suas próprias aplicações, jogos e ferramentas de trabalho. No decorrer do projecto, a aplicação foi validada por especialistas garantindo o cumprimento dos objectivos propostos.

Palavras-chave: Realidade Aumentada e Gestão de Conteúdos Multimédia.

II. Abstract

Augmented Reality has changed the way we view and interact with the real world. By expanding our reality with virtual realities allows the creation of new experiences. The applicability of these experiences, often result in a natural way to interact with virtual content, despite the fact that, creating such interaction can be complex, either by management or by the technological limitations of the involved resources.

The development of augmented reality projects, for commercial purposes, is often driven to optimize resources by taking into account the limitations of the technologies surrounding the systems (voice and motion detection, pattern detection, GPS, image analysis, biometric sensors, etc.).

There are many areas where augmented reality is applied (medicine, education, leisure etc.), with the popularization and reception of these techniques, it is necessary that this applications can be applied in every day devices (computers, tablet computers, mobile phones etc.). A common dominator between these devices is the Internet, therefore, applications that run online can reach a greater number of people.

In this perspective, the objective of this thesis project is to create a web application that uses augmented reality techniques whose contents are manageable by users. The application will consist on a FrontOffice and a BackOffice, which can be used by users, without programming knowledge, to create new applications with augmented reality and manage the media used. The FrontOffice will be responsible for display and interpret the created applications and enable integration with other applications in the browser.

The process of research and development will go through a prototyping phase to find technologies that fit the type of architecture required for the application and development tools used by the company where the project will be developed.

It is the intention to build an application that serves as the basis for other applications and can be reusable in other areas, always with the aim of reducing development costs and subsequent management of content. During the project, the application was verified by experts, thus providing the implementation of a framework that enables content management in different areas (medicine, education, leisure, etc...), where users can create their own applications, games and work tools.

Keywords: Augmented Reality and Multimedia Content Management.

III. Agradecimentos

A todos aqueles que me ajudaram durante esta etapa da vida, agradeço o vosso apoio, carinho e motivação, com especial atenção para...

O meu orientador, Prof. Doutor Carlos Vaz de Carvalho, pelas palavras de motivação, orientação e desafio.

A Front-end pela oportunidade de realizar este projecto em âmbito profissional, em particular ao meu co-orientador, Luís Castro, pelo voto de confiança.

Aos meus colegas e amigos de trabalho que directa ou indirectamente me ajudaram prestando e partilhando os seus conhecimentos.

Ao Duarte Gomes pela cedência dos seus recursos musicais.

Ao Eric Socolofsky pela disponibilidade e prontidão de resposta as questões técnicas colocadas.

Aos meus amigos, eles sabem quem são, que me ouviram a falar horas a fio e conseguiram aturar o meu super entusiasmo.

À minha mãe, pelo apoio, carinho e amor que sempre me inspiram.

À Fatinha, amor da minha vida, pela paciência, pela ajuda incansável e pelas palavras de incentivo que contribuíram para eu alcançar mais um marco na minha vida.

IV. Índice

1	Introdução.....	1
1.1	Enquadramento.....	2
1.2	Objectivo do trabalho	3
1.3	Motivação.....	3
1.3.1	Pessoal	3
1.3.2	Institucional	4
1.4	Organização da Dissertação	4
2	Conceitos e Estado da Arte	5
2.1	Tecnologias.....	8
2.1.1	Computer Vision.....	9
2.1.2	PlayStation Move	12
2.1.3	WiiMote	14
2.1.4	Kinect.....	15
2.1.5	Mobile AR	16
2.1.6	Realidade aumentada no Quotidiano	17
3	Metodologia	25
3.1	Identificação do Problema	25
3.2	Pesquisa Bibliográfica	26
3.3	Implementação dos Protótipos	26
3.3.1	Protótipo Flash	27
3.3.2	Protótipo Unity 3D	28
3.3.3	Protótipo Flash + Unity 3D	29
3.4	Validação dos Protótipos.....	30
3.5	Análise das Ferramentas de Desenvolvimento.....	30
3.6	Análise de Resultados e Produção de Conclusões	32
4	Implementação	35
4.1	Identificação dos requisitos	35
4.2	Descrição Técnica.....	37

4.2.1	Magic Catalog	38
4.2.2	Magic Catalog Viewer	39
4.2.3	API Javascript Magic Catalog.....	43
4.2.4	ARCMS	44
4.2.5	MC Media Viewer	53
4.3	Análise e apresentação de resultados.....	57
4.3.1	Testes de hardware	58
4.3.2	Testes de software	59
4.3.3	Demonstrações de utilização	63
4.3.4	Limitações das tecnologias utilizadas.....	65
5	Conclusões	67
5.1	Desenvolvimento Futuro	68
6	Bibliografia	69

V. Lista de Figuras

Ilustração 1 - Diagrama simplificado do Contínuo Realidade-Virtualidade de Milgram.....	5
Ilustração 2 – Sistema AR	7
Ilustração 3 – Sistema de Processamento de Informação AR	8
Ilustração 4 – Computer Vision Algorithm.....	9
Ilustração 5 – Comparação do padrão detectado com o padrão de referência	10
Ilustração 6 – Sistema de Coordenadas AR Toolkit.....	11
Ilustração 7 – Sistema de Coordenadas PS3 Move.....	13
Ilustração 8 – Sistema de Coordenadas WiiMote	14
Ilustração 9 – Projecto WiiMote	15
Ilustração 10 – Kinect grelha de infravermelhos para calcular a profundidade	16
Ilustração 11 – Kinect Mapa de profundidade usando gradientes de cor (branco = perto e azul = longe).....	16
Ilustração 12 – AR Weather - TV Patrol.....	17
Ilustração 13 – AR Soccer - MIT Media Lab	18
Ilustração 14 – AR Military - ARMAR.....	18
Ilustração 15 – AR Medical - TU München as surgeons from LMU.....	19
Ilustração 16 – Protótipo de aplicação laser na manutenção de uma impressora, mostrando como se remove a bandeja do papel. (Cortesia de Steve Feiner, Blair MacIntyre, Dorée Seligmann e Columbia University.).....	19
Ilustração 17 – Argon – Virtual Tour Guide.....	20
Ilustração 18 – Magic Book - Total Immersion.....	20
Ilustração 19 - AR Modelação de Objectos.....	21
Ilustração 20 – AR TISSOT Reality	22
Ilustração 21 – AR Sábado.....	22
Ilustração 22 – AR LEGO	22
Ilustração 23 – AR Coca-Cola	23
Ilustração 24 – AR Ray Ban Virtual Mirrow.....	23
Ilustração 25 – Protótipo Flash.....	27
Ilustração 26 - Protótipo Unity 3D.....	28
Ilustração 27 - Protótipo Flash + Unity 3D	29
Ilustração 28 - Tecnologias/Linguagens de programação utilizadas	38
Ilustração 29 – Magic Catalog Viewer	39
Ilustração 30 - Subsistemas Magic Catalog Viewer	41
Ilustração 31 – Algoritmo de carregamento de Objectos Multimédia	42
Ilustração 32 - ARCMS Autenticação	44

Ilustração 33 – ARCMS Gestão de Utilizadores	45
Ilustração 34 – ARCMS Gestão de Marcadores	46
Ilustração 35 – ARCMS Gestão de Catálogos	47
Ilustração 36 – ARCMS Configuração avançada do catálogo.....	48
Ilustração 37 – ARCMS Gestão de Objectos Multimédia.....	49
Ilustração 38 – ARCMS Base de Dados.....	50
Ilustração 39 – MC Media Viewer na edição de um objecto	53
Ilustração 40 – MC Media Viewer sem objecto	53
Ilustração 41 – MC Media Viewer com objecto 3D	53
Ilustração 42 – Demo Magic Catalog.....	63
Ilustração 43 – My Band.....	64
Ilustração 44 – Dazbook Catalog.....	65

VI. Lista de Tabelas

Tabela 1 – Resumo dos resultados da avaliação dos protótipos	32
Tabela 2 – Resumo dos resultados dos testes ao Magic Catalog Viewer	59
Tabela 3 – Resumo do Inquérito de Qualidade do ARCMS	61

I. Lista de Abreviações

- API** - *Application Programming Interface* (ou Interface de Programação de Aplicações)
- AR** - *Augmented Reality* (ou Realidade Aumentada).
- ARCMS** - *Augmented Reality Content Management System*. É o nome do sistema de gestão de conteúdos do Magic Catalog
- ARM** - *Augmented Reality Marker*, é um objecto da realidade cujo objectivo é ser substituído, ocultado ou servir de referência para localização da visualização do objecto virtual. No contexto desta tese, o conceito “Marcador” é utilizado quando estamos a referir ao conjunto de ficheiros de imagem (ARM) e respectivo padrão numérico, geridos pelo ARMCS.
- ARMs** - É utilizado para descrever o plural de ARM
- AS3** - ActionScript 3
- CMS** - *Content Management System* (Sistema de Gestão de Conteúdos)
- COLLADA** - COLLADA is a COLLABorative Design Activity for establishing an interchange file format for interactive 3D applications.
- CPU** - Central Processing Unit
- DLL** - Dynamic-link library (also written without the hyphen), or DLL, is Microsoft's implementation of the shared library concept in the Microsoft Windows and OS/2 operating systems.
- FLV** - Flash Video is a container file format used to deliver video over the Internet using Adobe Flash Player
- GIF** - The Graphics Interchange Format. Is a bitmap image format that was introduced by CompuServe in 1987 and has since come into widespread usage on the World Wide Web due to its wide support and portability.
- GLUT** - GLUT (pronounced like the glut in gluttony) is the OpenGL Utility Toolkit, a window system independent toolkit for writing OpenGL programs.
- GPS** - Global Positioning System (Sistema de Posicionamento Global)
- GPU** - Graphics Processing Unit
- JPG** - Is a commonly used method of lossy compression for digital photography (image).
- PDF** - *Portable Document Format*.
- PNG** - Portable Network Graphics, também conhecido como PNG's Not GIF) é um formato de dados utilizado para imagens, que surgiu em 1996 como

substituto para o formato GIF, devido ao facto de este último incluir algoritmos patenteados.

SWF - SWF (Shockwave Flash) é um formato de ficheiro de aplicações *web*, criado pela Macromedia, actualmente adquirida pela Adobe.

ZIP - O formato de ficheiro ZIP é um formato de compressão de dados e armazenamento.

1 Introdução

Na sociedade actual, cada vez mais referimos o termo Realidade Aumentada (AR - Augmented reality) para descrever características e funcionalidades em jogos e aplicações para dispositivos móveis e computadores pessoais. Mas o que é afinal Realidade Aumentada?

“Realidade aumentada é um termo que define a interacção directa ou indirecta da nossa visão e percepção do mundo real, na qual estendemos a nossa realidade com elementos virtuais gerados por computadores ou sensores em tempo real”.[Wikipedia, 2011]

Existem várias formas de implementar realidade aumentada mas normalmente as soluções são fechadas para as empresas que adquirem este tipo de aplicações, uma vez que não existe liberdade ou forma de modificar os conteúdos existentes. Dada a sua complexidade, os problemas de performance subjacentes à computação em tempo real ou a limitação dos sensores utilizados no processo a maioria das aplicações têm que ser desenvolvidas por programadores.

No entanto, a procura e investimento neste tipo de *software* por parte das empresas e instituições está a aumentar, dada a crescente preocupação destas em proporcionar novas formas de interacção entre os seus produtos e o mercado, assim como estabelecer novos canais de comunicação com os seus clientes.

Existe a necessidade destas aplicações serem mais flexíveis, isto é, que possibilitem a gestão dos seus conteúdos, a modificação do seu âmbito e propósito inicial sem a intervenção obrigatória de um programador ou o desenvolvimento de um novo projecto.

Com a globalização dos mercados e da *internet* é quase indispensável que as empresas e instituições disponibilizem na *internet* um “repositório” com informação sobre a sua missão, propósito, objectivos e fundamentalmente sobre o valor acrescentado que proporcionam aos seus clientes e que as distingue dos seus concorrentes, seja pelos seus produtos ou serviços.

Face às condições tecnológicas favoráveis, assiste-se à criação e expansão de novas formas de perceber e experimentar os produtos, isto é, quando uma pessoa utiliza uma aplicação, pensa no que pode fazer, interagir, sentir e experienciar durante esse período, isto é, na forma como irá fazer parte da aplicação onde o mundo real se funde com o mundo virtual.

1.1 Enquadramento

No mundo empresarial actual, existe a necessidade das empresas comunicarem com os seus clientes de forma distinta e eficaz, dada a crescente importância do relacionamento sustentado entre as instituições e o seu mercado. Face a este contexto económico-social, as empresas procuram oferecer serviços únicos e especializados para fidelizar clientes actuais e até mesmo atrair clientes potenciais acrescentando assim valor aos produtos que comercializam.

A comunicação através do *site* da instituição, tem vindo a crescer e, actualmente toda a informação está *online*, pois a mera informação sobre os produtos e o usual formulário de contactos já não é suficiente. Assim, é necessário manter os clientes atentos ao que empresa produz de novo e à mensagem que esta pretende transmitir.

O recurso à realidade aumentada proporciona outra forma de visualizar produtos e os conteúdos multimédia mas a gestão destes conteúdos nem sempre é realizada de forma directa e transparente. Existem empresas onde a gestão é realizada por um responsável pelo *site*, que nem sempre tem conhecimentos de programação ou design, pelo que algumas empresas vêm-se obrigadas a recorrer a *copywriters* para escrever e produzir conteúdos, enquanto outras subcontratam empresas especializadas em conteúdos *Web*.

A criação de conteúdos (e.g. áudio, vídeo e modelação 3D) acarreta sempre elevados custos para as empresas, especialmente quando se trata de produção específica e focalizada para um determinado objectivo, este custo é maior quando se acrescenta o desenvolvimento da aplicação com realidade aumentada. Dado que a maioria das aplicações com AR são fechadas e não contemplam gestão de conteúdos, sempre que é necessário modificar um vídeo ou modelo 3D da aplicação, o detentor do *software* terá de incorrer custos adicionais.

Como existe sempre um custo associado à produção de audiovisuais e modelação 3D (mesmo que interno) a inserção de uma funcionalidade de gestão de conteúdos nos sistemas de AR poderá diminuir consideravelmente os custos para as empresas produtoras e para os detentores deste tipo de *software*. Por um lado, as empresas que desenvolvem o *software* poderão vender um produto *standard* a outras empresas, sem encargos adicionais e necessidades de desenvolvimento extra, por outro lado, as compradoras sempre que necessitem de trocar o conteúdo da aplicação poderão fazê-lo autonomamente.

“Lembrai-vos de que tempo é dinheiro” [Benjamin Franklin]

1.2 Objectivo do trabalho

Os objectivos deste trabalho são:

1. Desenvolver uma aplicação com AR e que possibilite a apresentação de múltiplos objectos multimédia (e.g. imagens, vídeos, animações ou modelos 3D) em simultâneo.
A aplicação deve ser reutilizável em projectos com diferentes âmbitos sem necessidade de intervenção técnica.
A aplicação deve funcionar em ambiente Cliente-Servidor ou em *Standalone*.
2. Desenvolver um *BackOffice* capaz de:
 - a. Gerir os objectos multimédia da aplicação a cima referida;
 - b. Permitir a pré-visualização dos objectos utilizados;
 - c. Permitir modificar propriedades como a escala, tamanho e rotação dos objectos;
 - d. Associar um objecto multimédia a um ARM (*Augmented Reality Marker*) único de identificação;
3. Desenvolver uma API para integração com outras aplicações.

1.3 Motivação

A motivação para desenvolver este projecto adveio dos sucessivos pedidos à empresa onde trabalho, “FronD-End Multimédia, Lda.”, para desenvolver aplicações para exposições e demonstrações de diversos produtos nas quais se solicita o uso interactivo com a audiência. Esta necessidade manifesta por parte do mercado, identifica a oportunidade de investigar, inovar e construir uma base sólida de recursos que permita desenvolver rapidamente este tipo de soluções.

1.3.1 Pessoal

Desenvolver aplicações com Realidade Aumentada é fruto do interesse e gosto por jogos que usam a mesma tecnologia no seu *gameplay*. Aliado a isto, possibilita a minha especialização sobre o tema na área da multimédia e *Web*, com a aquisição destes novos conhecimentos e competências, resulta o desenvolvimento de uma diversidade de novos projectos e jogos no seio da empresa onde exerço actividade. Pode-se dizer que é um caso de “juntar o útil ao agradável”.

1.3.2 Institucional

Na perspectiva de alargar a oferta na área de negócio onde actua, a Front-end, reconheceu neste projecto a oportunidade de desenvolver uma plataforma que respondesse rapidamente às necessidades manifestadas pelos seus clientes. Com a existência de várias oportunidades, tornou-se necessário investir no conhecimento das tecnologias de realidade aumentada para expandir o leque de soluções apresentadas aos clientes da empresa.

1.4 Organização da Dissertação

O documento está organizado em seis capítulos, sendo que o primeiro capítulo corresponde à introdução do tema abordado.

O segundo capítulo descreve e contextualiza o conceito de realidade aumentada recorrendo a exemplos de utilização.

No terceiro capítulo apresenta-se o método de investigação, como foi abordado a resolução do problema encontrado, explica-se a implementação e avaliação realizada aos protótipos com o fim de validar as tecnologias utilizadas na implementação do projecto final.

No quarto capítulo identifica-se os requisitos de desenvolvimento deste projecto, o método de implementação e as tecnologias utilizadas. Efectua-se a avaliação do trabalho, onde são descritos os testes efectuados e os resultados obtidos. No final do capítulo são apresentados exemplos de implementação e as limitações das tecnologias utilizadas na aplicação.

No quinto capítulo sucedem as conclusões do projecto realizados e apresenta-se as perspectivas de desenvolvimento futuro.

O sexto e último capítulo deste documento contém as informações das referências bibliográficas utilizadas.

2 Conceitos e Estado da Arte

Segundo Milgram, a Realidade Aumentada (AR) é um dos pólos que constitui a Realidade Mista (MR). A MR é a relação entre a AR e a Realidade Virtual (VR). Na VR o observador é imerso por um mundo sintético, que pode ou não imitar as propriedades do mundo real ou de um mundo fictício. Na AR os objectos sintéticos são visualizações do mundo real através de um ecrã. Estes dois conceitos devem ser considerados lados opostos do mesmo *continuum* referido como Continuum Realidade-Virtualidade (RV) tal como apresentado na Ilustração 1. O ambiente MR é gerado quando os objectos do mundo real e virtual são representados em conjunto num único ecrã, ou seja, dentro do intervalo definido pelo *continuum* RV [Milgram, Takemura, Utsumi, & Kishino, 2011].

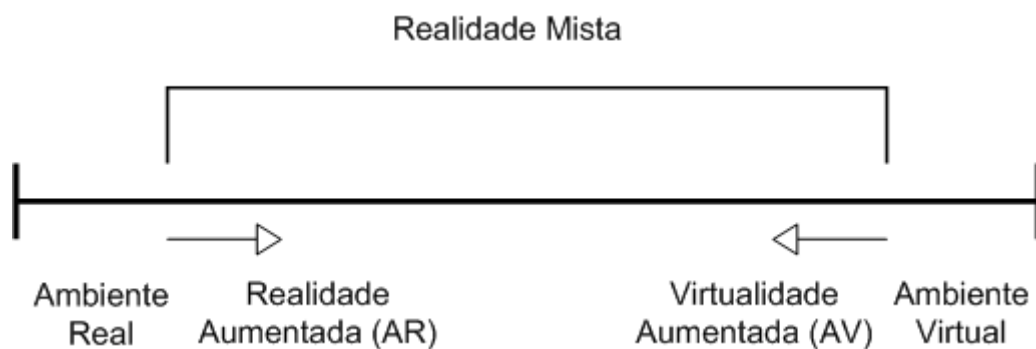


Ilustração 1 - Diagrama simplificado do Continuum Realidade-Virtualidade de Milgram

Existem algumas vantagens na utilização da AR em comparação com o uso exclusivo de VR. A VR é ideal para substituir o mundo real em simulações, treinos e jogos, enquanto a AR “melhora” e possibilita uma interacção mais sofisticada com mundo real, combina imagens reais e virtuais, é interactiva em tempo real e é realizada em ambiente 3D. A renderização dos objectos é mais fácil, uma vez que aparecem em menos quantidade do que na VR. A detecção e reconhecimento são mais complexos por diversos motivos, requerendo uma maior largura de banda (processamento de vídeo e dados) e sendo necessário tipificar o meio envolvente. Em termos de portabilidade, no sistema VR o utilizador está apenas num só sítio e na AR o utilizador “desloca-se” para a tarefa no mundo real [Cooper, 2011].

A visualização de realidade aumentada pode ser dividida em duas categorias, a primeira é composta pelos óculos e ecrãs transparentes, onde são apresentados os objectos virtuais e pelos quais conseguimos visualizar o mundo real, a segunda é composta pela visualização da realidade a partir de um monitor, onde as imagens são capturadas por uma câmara de vídeo [Milgram, Takemura, Utsumi, & Kishino, 2011].

Em ambas as categorias de AR referidas anteriormente, existem problemas que estão continuamente a ser optimizados:

- Os gráficos gerados por computador devem ser actualizados com mais frequência, para que os objectos apareçam no sítio correcto do mundo real
- A detecção deve ser mais precisa e mais rápida, para que os objectos fiquem alinhados com o mundo real

Durante o processo de detecção de AR podem aparecer erros provenientes da fonte de obtenção dos dados:

- Erros estáticos
 - Distorções ópticas
 - Desalinhamentos mecânicos
 - Erros de Detecção
 - Parâmetros de visualização incorrectos
- Erros dinâmicos
 - Atrasos na resposta do sistema (maior fonte de erros)

Os Marcadores (ARM) são um dos elementos fundamentais num sistema AR. Um marcador pode ser qualquer objecto ou pessoa, desde que seja, distinto, orientado e identificável por processos automáticos. Os marcadores podem ser associados e substituídos no ecrã por qualquer conteúdo virtual. [Cooper, 2011]

É comum nas aplicações e jogos com AR serem usados padrões para localizar, analisar e representar os objectos virtuais. Estes padrões podem ser rostos, formas do corpo, imagens impressas, comandos de jogos que combinam infra-vermelhos, giroscópios, acelerómetros, sensores de campos magnéticos terrestres, sistemas de posicionamento global e câmaras para analisar a posição da pessoa ou do padrão a localizar. Permitindo assim, por exemplo, controlar veículos num jogo, jogar golfe, interagir com personagens animadas ou “fazer” parte activa do jogo.

Geralmente para desenvolver uma aplicação ou jogo com AR é necessário o seguinte:

1. Sensores para captar informação (Câmaras de Vídeo, Sensores de pressão, GPS, Microfones, etc.)
2. Software ou Hardware para detectar e interpretar os padrões que se pretende identificar.

3. Dispositivos para reproduzir o resultado dos processamentos efectuados no ponto anterior e integrar a realidade na aplicação (realidade virtual) ou vice-versa.

O método mais comum para criar este tipo de aplicações é usar uma *webcam* para capturar a “nossa realidade”, uma biblioteca de reconhecimento de padrões para interpretar as imagens capturadas, um motor gráfico 3D e um ecrã para representar os objectos virtuais “embebidos” no nosso mundo. Existem outros métodos e técnicas para desenvolver aplicações AR, no entanto, este documento irá focar o método anteriormente descrito, sendo este utilizado frequentemente em jogos e aplicações *web*.

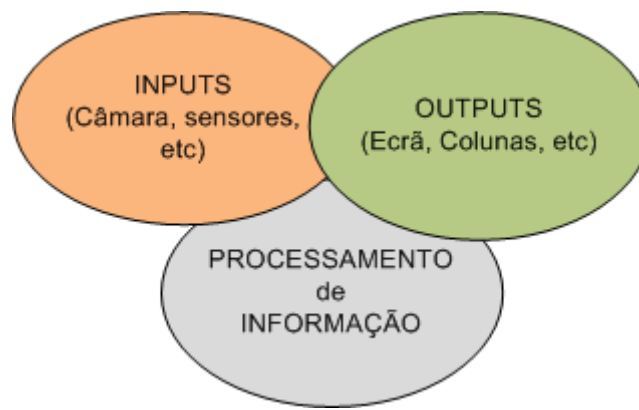


Ilustração 2 – Sistema AR

Ao abordar o sistema AR (Ilustração 2) de uma forma simplista e considerando que os programadores não possuem um controlo directo sobre a concepção dos sistemas de entrada e saída (i.e. câmara de filmar e ecrã), podemos inferir que o sistema “Processamento de Informação” é um factor chave na optimização da performance da nossa aplicação e na forma como esta é desenvolvida. Um bom exemplo são os jogos desenvolvidos para a PlayStation (PlayStation Move e PlayStation Eye) e para a Wii (Wiimote e Sensor Bar) onde os periféricos de entrada e de saída já estão especificados e só podem ser optimizados pelo fabricante. A mesma analogia também pode ser feita em aplicações para dispositivos móveis que usam o GPS e a câmara como entrada de informação. Nos exemplos dados anteriormente, os programadores estão limitados aos dispositivos e tecnologias disponíveis, o que implica, que na maioria dos casos, o sistema de processamento de informação tem um grande impacto na aplicação final.

Na maior parte dos casos, o sistema de processamento de informação pode ser decomposto em 4 componentes base:

- Detecção de Padrões
- Conversor de Coordenadas
- Motor Gráfico 3D
- Interface da Aplicação

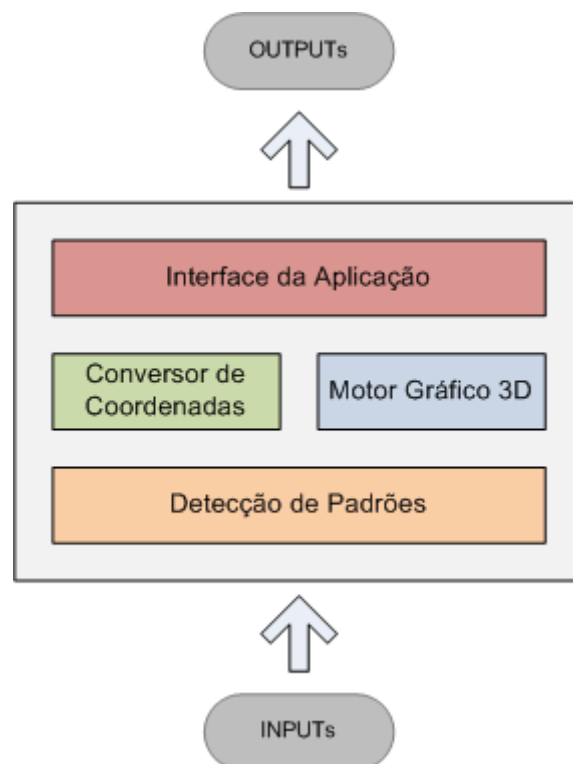


Ilustração 3 – Sistema de Processamento de Informação AR

2.1 Tecnologias

Existem múltiplas tecnologias para a criação de AR e algoritmos de detecção de padrões, neste sentido, irão ser apresentados a tecnologias mais conhecidas e com resultados comprovados de sucesso. Antes de mais, cumpre referir que os exemplos apresentados são centrados nos objectivos deste trabalho e apenas representam uma amostra significativa das tecnologias que existem actualmente.

2.1.1 Computer Vision

O algoritmo *Computer Vision* implementado na API ARToolKit¹, tem como base a detecção dos cantos de um quadrado de uma imagem capturada por uma câmara, o algoritmo prevê a posição do padrão através da análise das bordas conexas dos quatro cantos detectados [Kato & Billinghurst, 2011]. Esta tecnologia é *Open Source*, multi-plataforma e resolve dois problemas dos sistemas de AR: a detecção e a interacção. [Cooper, 2011]

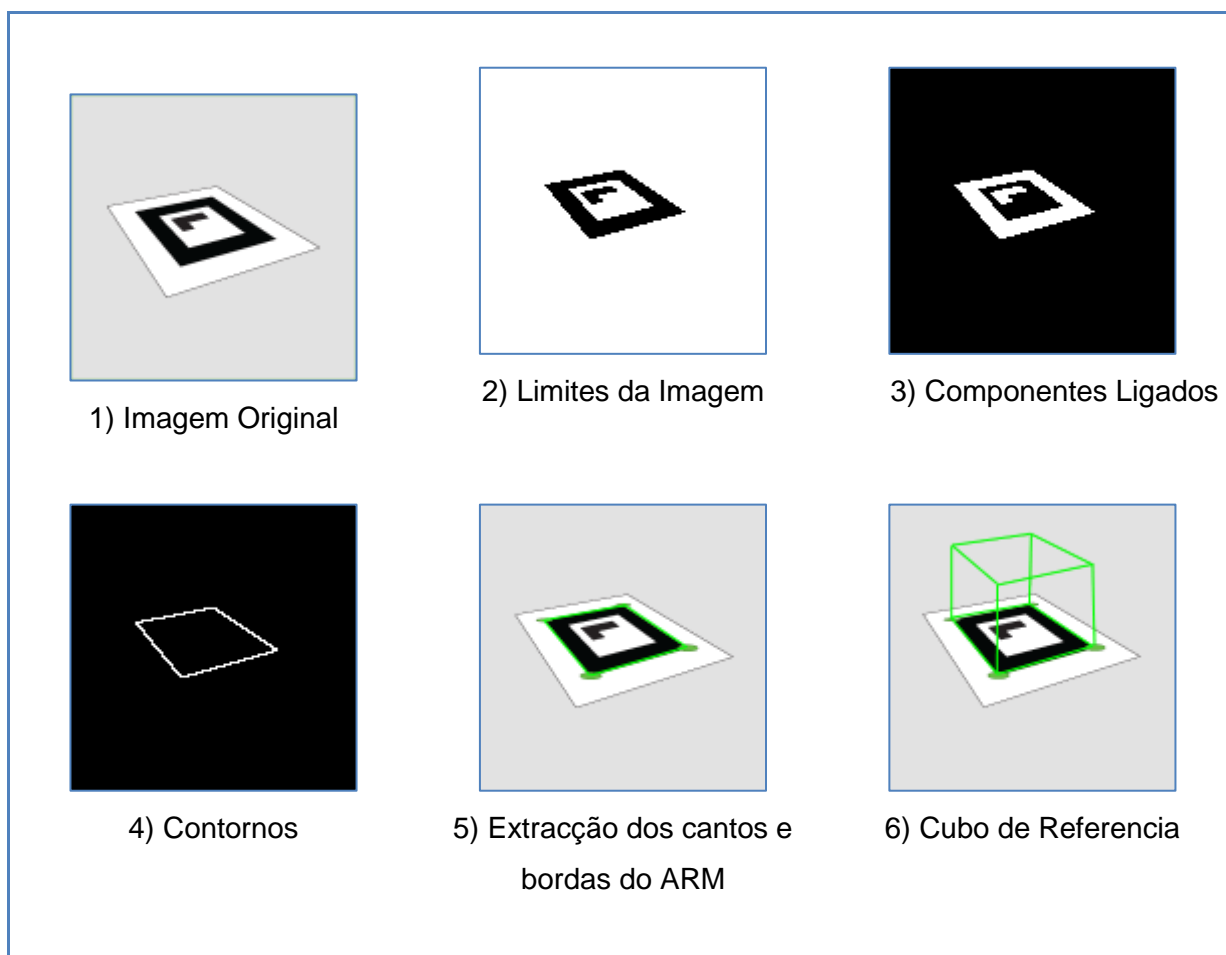


Ilustração 4 – Computer Vision Algorithm

A Ilustração 4 representa, de forma sumária, os diferentes estados do algoritmo. Nesta fase da detecção apenas interessa identificar os quadrados que possam existir na imagem. Para isso, a partir da imagem original (imagem 1) é aplicado um filtro para detectar os limites da imagem (*threshold*), este filtro transforma uma imagem a cores numa imagem a preto e branco (imagem 2). O próximo passo é detectar e etiquetar todos os componentes da imagem que estão ligados (imagem 3). Após a identificação destes componentes, é necessário detectar e estimar os contornos dos quadrados que possam existir. Para todos os contornos encontrados (imagem 4), temos que procurar e extrair os cantos do quadrado

¹ ARToolKit - é uma biblioteca de software para construir aplicação com realidade aumentada.
Site: <http://www.hitl.washington.edu/artoolkit/>

(imagem 5), assim é possível estimar a localização de um cubo de referência (imagem 6) que irá conter as coordenadas e o padrão nele encontrado.

Depois de identificar todos os quadrados na imagem é necessário cruzar os padrões detectados com os padrões de referência que existem na aplicação (ver Ilustração 5).

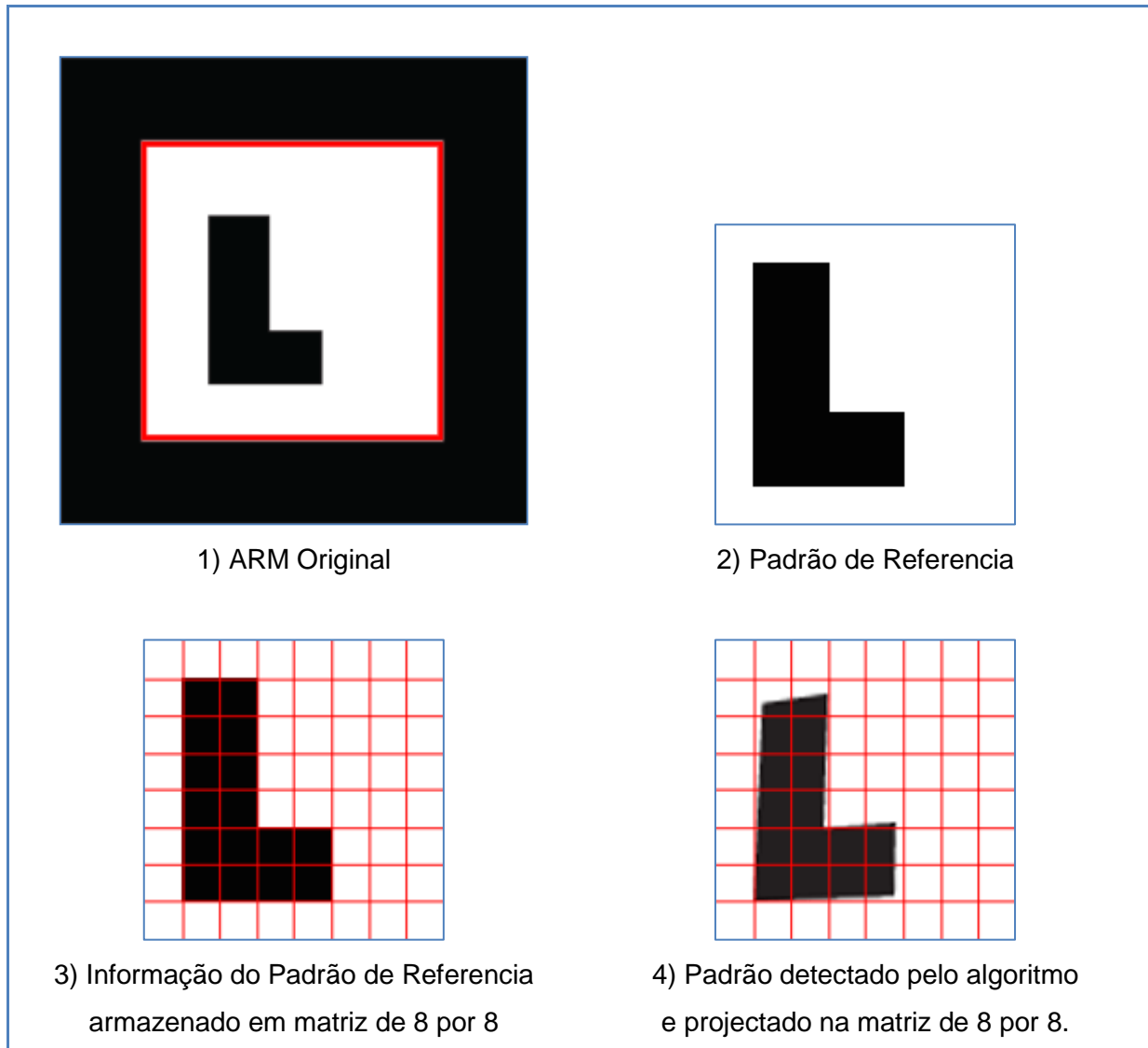


Ilustração 5 – Comparação do padrão detectado com o padrão de referência

O processo de comparação devolve a probabilidade de o padrão encontrado ser um padrão de referência, para isso a matriz de pontos detectada é comparada com a matriz de referência e respectivas transformações. Isto implica que o ARM tem que ser orientado, ou seja, rotações de 90° no ARM tem que resultar numa imagem distinta entre todas as rotações.

Com a informação da probabilidade podemos ajustar o grau de precisão com que os ARM são detectados. Por outro lado, também podemos aumentar o número de elementos da

matriz para obter resultados mais precisos e um maior número de ARM distintos. No entanto, quanto maior for a matriz de comparação, mais pesado irá ser o processo de detecção.

Como resultado final destes dois sub-processos identificamos o ARM e as suas coordenadas, com a orientação do padrão do ARM obtemos as coordenadas X e Y, a profundidade do ARM (coordenada Z) é obtida através do tamanho do quadrado detectado em relação ao número de pixels que ele ocupa no ecrã (o tamanho de referência do ARM é 8 por 8 cm, que inclui as bordas pretas).

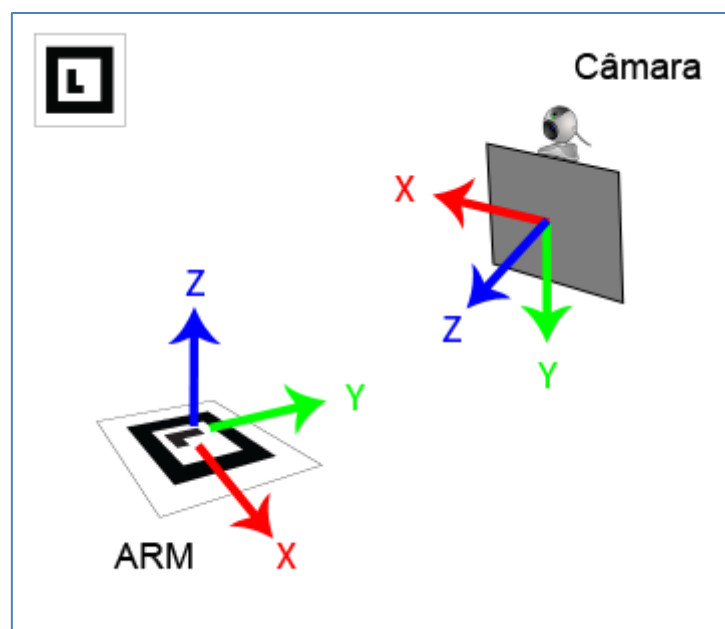


Ilustração 6 – Sistema de Coordenadas AR Toolkit

O conversor de coordenadas permite abstrair o interface da aplicação do motor 3D utilizado, uma que vez este componente transforma as coordenadas calculadas pelo algoritmo de detecção de padrões para o sistema de coordenadas do motor 3D.

Tendo em conta que existe diferentes implementações deste algoritmo de detecção, em várias linguagens de programação, a escolha do motor gráfico depende do tipo de aplicação e das ferramentas de desenvolvimento utilizadas.

Se a aplicação é um *Standalone*, compilada para um sistema específico, existe uma variedade de motores 3D, muitos dos quais *opensource*, como por exemplo o OpenGL, OGRE e o Blender. No caso de uma aplicação para *browser*, a escolha é mais limitada, no entanto podemos recorrer a motores como Papervision 3D, Away3D, Alternativa3D, Sandy3D e Unity 3D.

As principais limitações da tecnologia de detecção são:

- A representação dos objectos é feita apenas quando os ARMs estão visíveis
- A iluminação e a sombra podem afectar a detecção
- O alcance da detecção difere com o tamanho do ARM
- A precisão da detecção varia com o ângulo do ARM
- A velocidade da detecção diminui com o número de ARM visíveis

2.1.2 PlayStation Move

O PlayStation Move [PS Move Community, 2011] é um comando de jogo que em conjunto com a câmara PlayStation Eye consegue detectar os movimentos e a posição do jogador. O comando está equipado com acelerómetro de 3 eixos, um giroscópio de 3 eixos e um magnetómetro que ajuda a calibrar a orientação do comando com o campo magnético da terra. A posição do jogador é determinada pela câmara através da detecção da esfera luminosa no topo do comando. Assumindo que o sistema de coordenadas é o representado na Ilustração 7, as coordenadas X e Y são obtidas pela posição da esfera no plano da imagem e a profundidade (coordenada Z) é determinada pelo seu diâmetro na imagem capturada pela câmara.

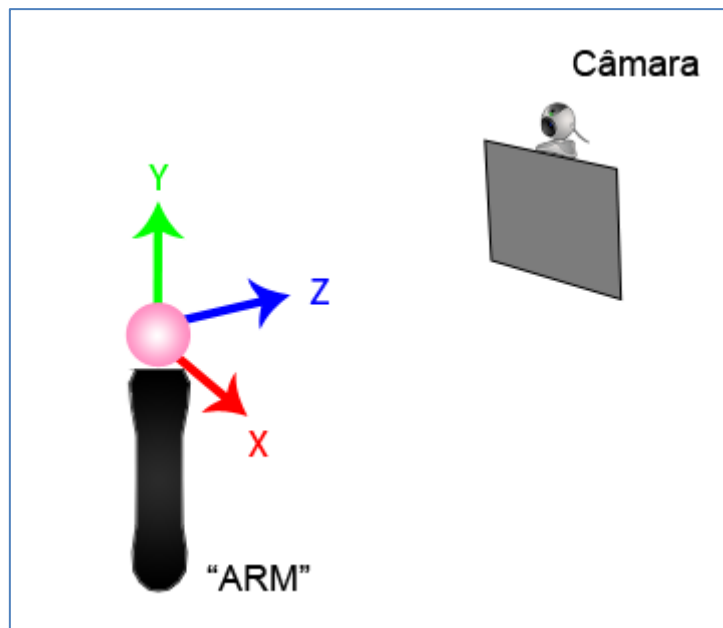


Ilustração 7 – Sistema de Coordenadas PS3 Move

Segundo o fabricante, a utilização da forma esférica (luminosa) permite que o controlador funcione com baixa latência, uma vez que esfera é um objecto geométrico fácil de detectar. Fazendo a analogia com um sistema de colisões de objectos tridimensionais, uma das técnicas utilizadas na detecção de colisões entre dois objectos complexos, passa por envolver os objectos em esferas e verificar a sua intersecção, a equação geométrica que define a esfera permite facilmente verificar estas colisões. Para facilitar a detecção da esfera, o comando possui três LEDs de cor que emitem luz na gama RGB, a cor da esfera é atribuída automaticamente após avaliação do ambiente em que o jogador se encontra.

O PS3 Move utiliza quatro tecnologias que se complementam:

- Localização espacial do jogador através da esfera luminosa do comando
- Detecção de movimento pelos sensores do comando
- Detecção de movimento através da análise dos pixéis nas imagens capturadas pela câmara
- Detecção de padrões recorrendo a algoritmos semelhantes ao *computer vision*

Este conjunto de tecnologias aumenta consideravelmente o número de aplicações que se podem desenvolver e ao mesmo tempo que possibilita novas formas interagir com os jogos.

2.1.3 WiiMote

O Wiimote [Lee, 2011] é um comando de jogos para a Wii, à semelhança do PlayStation Move, na versão mais recente, também possui incorporado um acelerómetro e um giroscópio. Uma das diferenças está presente na câmara incorporada numa das extremidades do comando e na Sensor Bar (que funciona como “ARM”) que está fixa por baixo ou por cima do ecrã. O Wiimote utiliza uma câmara de infravermelhos para detectar os “ARMs”, que são dois LEDs infravermelhos localizados na Sensor Bar.

Embora o sistema base da Wii seja mais limitado para desenvolver aplicações com realidade aumentada, é sempre possível utilizar uma câmara extra para detectar padrões e juntamente com o comando criar um sistema semelhante ao do PlayStation Move. Dando o exemplo de um jogo em *First Person Shooter*, e considerando que o sistema de coordenadas é o representado na Ilustração 8, é possível obter as coordenadas X e Y através do ponto médio calculado pela captura da imagem dos dois pontos infravermelhos na câmara do comando. A profundidade (coordenada Z) é obtida pela distância entre os dois pontos.

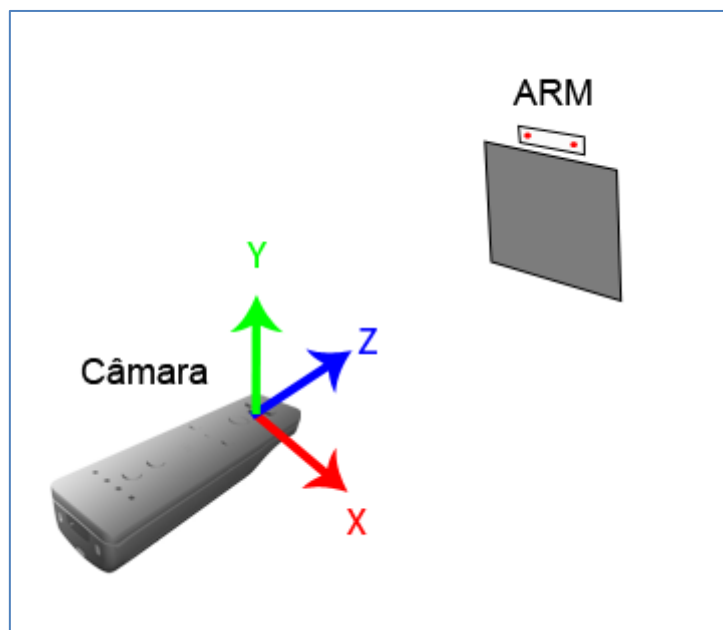


Ilustração 8 – Sistema de Coordenadas WiiMote

Existem biblioteca, como por exemplo o Wii Flash², que permitem usar o Wiimote em outros sistemas e linguagens de programação. No projecto desenvolvido em âmbito curricular, apliquei esta tecnologia para criar um sistema idêntico ao utilizado no filme “Minoraty Report” que permitia visualizar uma galeria de imagens utilizando as duas mãos. Ao contrário do sistema utilizado na Wii, o Wiimote estava fixo e os LEDs é que se movimentavam (ver Ilustração 9). Neste projecto a imagem era projectada numa parede ou tela, o Wiimote era posicionado de forma a capturar a imagem projectada e o utilizador tinha um par de luvas com um LED infravermelho no dedo indicador que era accionado quando este encostava no dedo médio. Se fosse detectado um LED em cima de uma imagem era possível “pegar” e movê-la, se dois LEDs fossem detectados por cima de uma imagem, era possível aumentar e diminuir o tamanho da imagem com base na distância entre dois os LEDs, em simultâneo era possível rodar a imagem com base no declive da recta que passa nos dois LEDs.

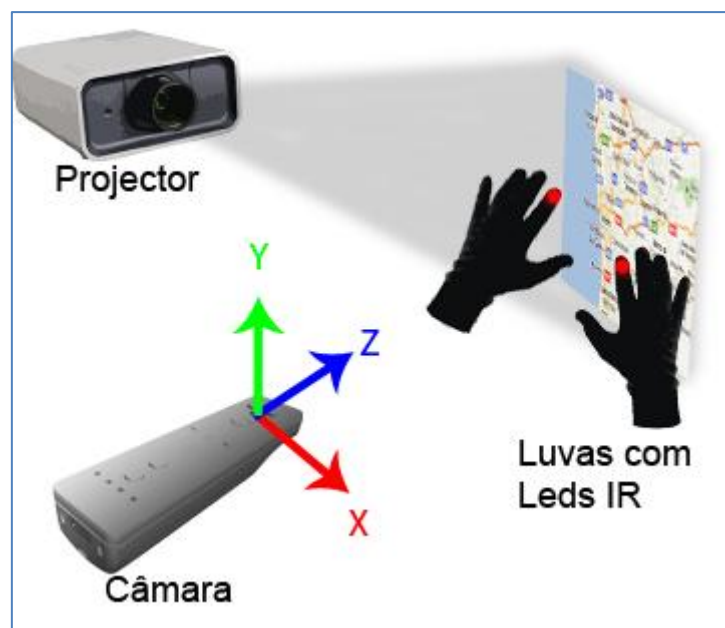


Ilustração 9 – Projecto WiiMote

2.1.4 Kinect

O Kinect é um multi-sensor desenvolvido pela Microsoft para a plataforma de jogos Xbox. É composto por uma câmara RGB, um sensor de profundidade, que combina um projector laser infravermelho e um sensor CMOS monocromático, e um *array* de 4 microfones. O

² WiiFlash is a project dedicated to the Wiimote and Flash® applications. Site: <http://wiiflash.bytearray.org/>

software proprietário do Kinect permite a captura 3D do corpo de uma pessoa, assim como o reconhecimento facial e de vocal.

Em Junho de 2011 a Microsoft lançou um SDK para o desenvolvimento de aplicações utilizando o Kinect. Investigadores do MIT Media Lab estão a trabalhar numa extensão de JavaScript para o Google Chrome chamado depthJS que permite os utilizadores controlar o *browser* com gestos. [Microsoft, 2011]



Ilustração 10 – Kinect grelha de infravermelhos para calcular a profundidade

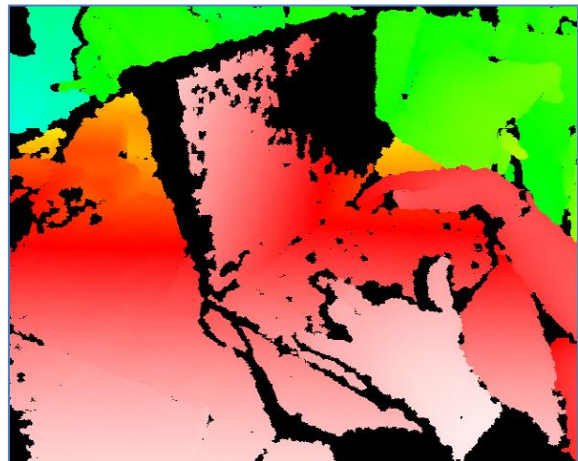


Ilustração 11 – Kinect Mapa de profundidade usando gradientes de cor (branco = perto e azul = longe)

2.1.5 Mobile AR

Com os avanços tecnológicos dos dispositivos móveis, na capacidade de processamento de informação e nos componentes incorporados, estes podem ser considerados os equipamentos mais propícios para a utilização de aplicações com realidade aumentada, uma vez que podem ser usados em qualquer lugar e fazem parte da vida diária de milhões de pessoas.

Os dispositivos actuais vêm equipados com câmara RGB, sensores de movimento como acelerómetros e giroscópios, GPS, WiFi e ecrãs táctil multi-toque. A integração do GPS nos dispositivos móveis permitiu alargar o âmbito das aplicações desenvolvidas, para além serem capaz de reconhecer padrões (e.g. versão do ARToolkit para dispositivos moveis) e posicionar os objectos virtuais na nossa realidade, com o GPS, é possível usar a “pessoa” como ARM para gerar a realidade aumentada.

Existem aplicações de realidade aumentada que só fazem sentido quando utilizadas num dispositivo móvel, como por exemplo:

- Saber o menu de um restaurante apontando o telemóvel para a porta do mesmo.
- Consultar informação sobre as lojas à medida que se passa estas na rua.
- Ver fotografias de um lugar, tiradas em diferentes alturas do dia e por outras pessoas.

Esta combinação de tecnologias permite transformar um simples telemóvel num dispositivo multifuncional, que através de aplicações com realidade aumentada vão facilitar e enriquecer as experiências no dia-a-dia das pessoas.

2.1.6 Realidade aumentada no Quotidiano

A realidade aumentada está presente no nosso quotidiano, quer por utilização directa ou indirecta destas tecnologias. Muitas das vezes estamos presentes perante estas tecnologias e nem nos apercebemos da sua aplicação.

De seguida são apresentados alguns exemplos de utilização de realidade aumentada utilizada em diferentes cenários.

Iniciando com o exemplo de uma apresentação da previsão da meteorologia onde o apresentador interage com o mapa virtual para saber o tempo nos diferentes locais.[ABS-CBN Regional Network Group, 2011]



Ilustração 12 – AR Weather - TV Patrol

Na análise televisiva de um jogo de futebol os comentadores desenham um linha virtual no campo para verificar se um jogador esta fora de jogo e círculos para estimar se o jogador esta a distancia mínima permitida da bola. Estas ferramentas possibilitam observar aspectos técnicos que são difíceis de perceber observando só para as imagens do jogo. [MIT, 2011]

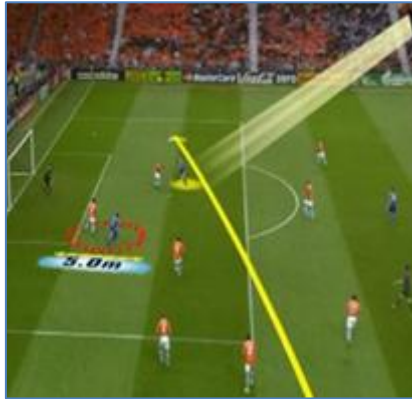


Ilustração 13 – AR Soccer - MIT Media Lab

O ensino da condução de um tanque ou de um avião é levada a cabo por simuladores que usam os comandos físicos para o utilizador pilotar o veículo e utilizam uns óculos especiais que permitem a visão periférica do mundo virtual.[Henderson & Feiner, 2011]



Ilustração 14 – AR Military - ARMAR

Na medicina é possível um cirurgião conseguir ver os sinais vitais do paciente, a localização correcta dos órgãos e músculos através de óculos especiais, eliminando a necessidade de ter que olhar para o lado para verificar os sinais do paciente na máquina, permitindo ao médico concentra-se na operação. O treino para as cirurgias também pode ser realizado pelo mesmo método, simulando todo o processo da intervenção. [Technische Universität München, 2011]



Ilustração 15 – AR Medical - TU München as surgeons from LMU

Na manutenção e reparação de máquinas complexas é mais fácil receber instruções em desenhos 3D sobrepostos sobre o equipamento real, em vez de consultar os manuais em texto e imagens. Com este método é mostrado passo-a-passo que tarefas devem ser realizadas e como devem ser. Os desenhos 3D podem ser animados, tornando as indicações mais perceptíveis. [Azuma, 2011]



Ilustração 16 – Protótipo de aplicação laser na manutenção de uma impressora, mostrando como se remove a bandeja do papel. (Cortesia de Steve Feiner, Blair MacIntyre, Dorée Seligmann e Columbia University.)

Sistemas virtuais de visitas guiadas possibilitam que as pessoas que utilizam um dispositivo próprio ou um telemóvel visualizem informações sobre a localização actual e qual caminho a seguir para continuar com a visita. [Georgia Tech, 2011]

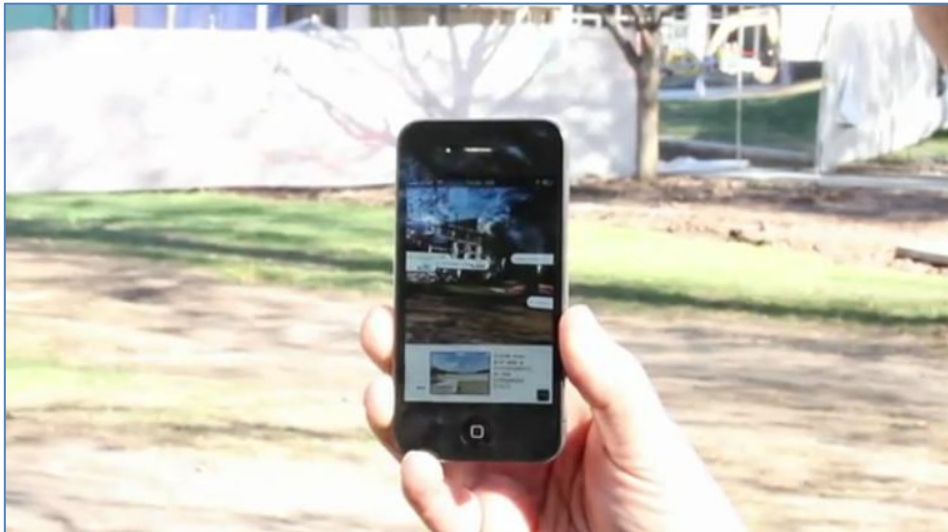


Ilustração 17 – Argon – Virtual Tour Guide

Os Livros “Mágicos” através do *software* proprietário e com a utilização da *webcam* permitem a visualização das folhas em papel seja animada e com transmutações de objectos 3D, em alguns casos é possível interagir com este objectos.[Total Immersion, 2011]



Ilustração 18 – Magic Book - Total Immersion

Os investigadores da University of Canterbury's Human Interface Technology Laboratory, conceberam uma aplicação que transforma rascunhos feitos em papel nas respectivas representações tridimensionais e que possibilitam interagir com o modelo gerado. [Bergig, Hagbi, El-Sana, & Billinghamurst, 2009]



Ilustração 19 - AR Modelação de Objectos

A marca de relógios TISSOT disponibilizou as suas colecções na aplicação “TISSOT REALITY” que simula os relógios no pulso do cliente. Para isso, é necessário imprimir um relógio de papel que tem a imagem de um ARM, instalar no computador a aplicação com a colecção pretendida e para ver os relógios simulados no pulso mostra-se o relógio de papel à webcam.[TISSOT, 2011]

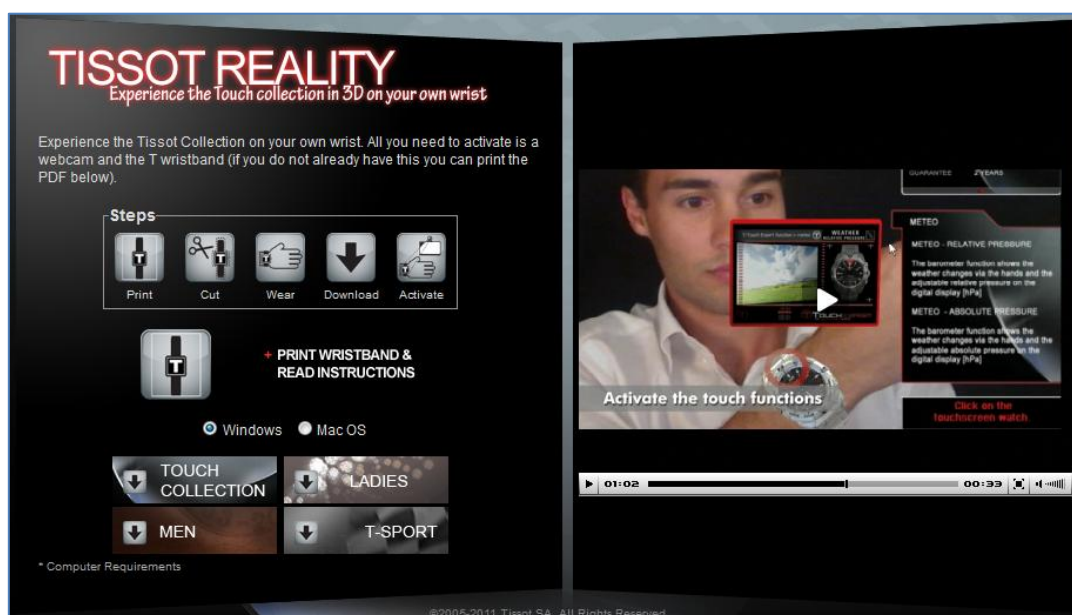


Ilustração 20 – AR TISSOT Reality

A revista Sábado foi a primeira newsmagazine a utilizar realidade aumentada (site: <http://ra.sabado.pt/>). Na revista foram impressos ARMs que visualizados pela *webcam* se convertiam em imagens, sons e vídeos. Por cada tópico existia uma explicação adicional ou conteúdo fotográfico extra. [Revista Sábado, 2011]



Ilustração 21 – AR Sábado

A LEGO na Disney Land de Orlando (EUA) disponibiliza uma aplicação para pais e crianças verem o modelo 3D dos Legos que estão dentro da caixa que vão comprar. Ao mostrar a caixa para a *webcam* (acima do ecrã) os Legos aparecem montados em cima da caixa. [Metaio, 2011]



Ilustração 22 – AR LEGO

A Coca-Cola numa edição especial da Coca-Cola ZERO lançou umas garrafas com um ARM no rótulo, quando visualizadas na aplicação distribuída pela empresa era possível controlar a posição de um helicóptero do filme Avatar. [Coca-Cola, 2011]



Ilustração 23 – AR Coca-Cola

A Ray Ban disponibiliza a aplicação Virtual Mirrow que a partir do reconhecimento facial simula vários modelos de óculos colocados na pessoa. [Ray Ban, 2011]

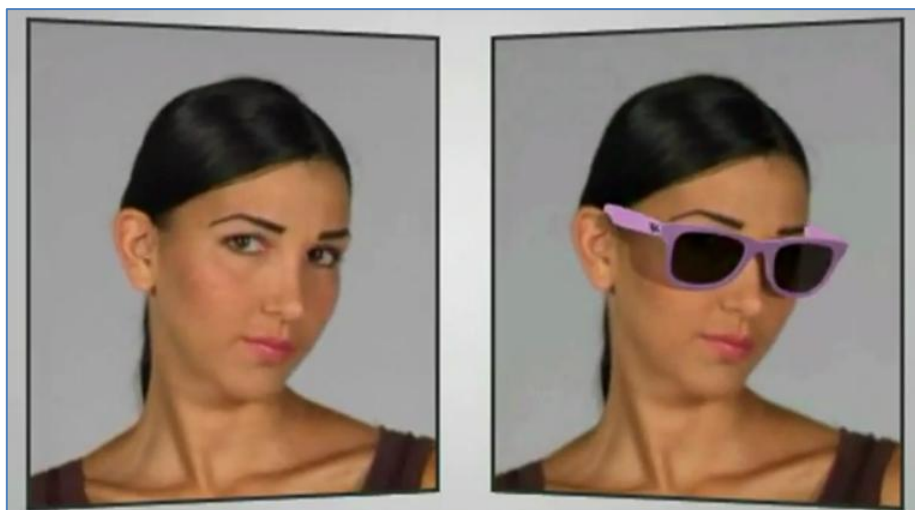


Ilustração 24 – AR Ray Ban Virtual Mirrow

A ADIDAS, num lançamento de uma modelo de sapatilhas, disponibilizou uma aplicação que detectava o novo modelo de sapatilhas e mostrava uma animação personalizada com vários elementos alusivos a ocasião. [Metaio, 2011]



Os exemplos apresentados são apenas alguns dos trabalhos desenvolvidos com realidade aumentada, que se apresenta como uma tecnologia em expansão e com aceitação no público em geral. Neste sentido, as empresas que desenvolvem este tipo de aplicações, começam a disponibilizar ferramentas para o utilizador criar as suas próprias aplicações. Este acontecimento está a fomentar o aparecimento de novas ideias e novas aplicações no mercado.

3 Metodologia

Neste capítulo descrevem-se os principais conceitos alusivos à metodologia de investigação usada neste projecto, pretendendo delinear a forma como foram validadas ou rejeitadas as posições tomadas. O método de investigação utilizado foi um misto de investigação descritiva (que descreve situações, eventos e condições que existem na actualidade), investigação avaliativa (feita para ajuizar sobre a qualidade dos programas, procedimentos e materiais) e investigação experimental (para avaliar a diferenças dos resultados obtidos).

3.1 Identificação do Problema

Na gestão de desenvolvimento de aplicações é necessário otimizar os recursos humanos e o tempo que cada colaborador fica alocado a um projecto, sendo mais crítico quando se trata de micro e pequenas empresas, onde é esperada a polivalência e a intervenção multi-projecto de cada colaborador.

Existem projectos-piloto ou de sazonalidade desconhecida, fora da área de negócio habitual da empresa, que acarretam um custo mais elevado, não só devido ao tempo dispendido com a investigação dos recursos que vão ser utilizados, mas também, pela rapidez de execução devido à falta de familiarização com as tecnologias.

No caso específico da empresa onde exerço actividade, surgem anualmente em média entre dois a três projectos de realidade aumentada e normalmente com um curto prazo de execução. Nestes casos, nem sempre é possível alocar um colaborador que já tenha participado neste tipo de projectos ou dar formação necessária sobre as tecnologias envolvidas. Por norma, em cada projecto de AR, os respectivos conteúdos multimédia são produzidos especificamente para aquele cliente. Frequentemente, é necessário actualizar estes conteúdos após a finalização da implementação do projecto, o que significa que o colaborador que foi alocado a este projecto vai ter que proceder com as modificações necessárias, o que implica diferir o desenvolvimento do seu trabalho actual.

Neste âmbito, para minimizar impacto nos custos internos da empresa com o tempo dispendido, surge a necessidade de criar uma aplicação que abstrai a programação e facilite a gestão de conteúdos multimédia nas aplicações AR.

3.2 Pesquisa Bibliográfica

A *internet* foi a maior fonte de recolha de conhecimento, através de pesquisa no Google e Wikipédia, onde facilmente se identifica as referências para os *sites* e fóruns de discussão especializados em Realidade Aumentada com a respectiva documentação técnica.

Procura-se entender os diferentes mecanismos necessários ao funcionamento da Realidade Aumentada, como podem contribuir para a alcançar os objectivos propostos nesta tese, como se pode melhorar e interligar as múltiplas tecnologias que existem na actualidade.

Neste sentido, a pesquisa foi realizada para obter os conhecimentos técnicos necessários ao desenvolvimento deste tipo de aplicações, com o objectivo de acrescentar valor tecnológico aos processos inerentes nas aplicações com Realidade Aumentada.

3.3 Implementação dos Protótipos

Como forma de cumprir os objectivos propostos, surgiram três soluções que podiam ser abordadas no desenvolvimento da aplicação, implementação em Flash³, Unity 3D⁴ ou Flash interligado com Unity 3D. Ambas as tecnologias funcionam em *browser*, tem acesso a *webcam* dos computadores e podem utilizar um motor gráfico 3D para visualizar imagens, vídeos e modelos 3D. Recorreu-se a método prototipagem para escolher a solução mais viável à implementação da aplicação final.

³ Adobe® Flash® Player is a cross-platform, browser-based application runtime that provides uncompromised viewing of expressive applications, content, and videos across browsers and operating systems.

⁴ Unity is a multiplatform game development tool, designed from the start to ease creation. A fully integrated professional application, Unity just happens to contain the most powerful engine this side of a million dollars.
Website: <http://unity3d.com/>

3.3.1 Protótipo Flash

O desenvolvimento da aplicação em Flash (AS3), utiliza a biblioteca de reconhecimento de padrões “FlarToolKit”⁵ que possibilita aceder ao *feed* de vídeo da *webcam*, utilizar *players* de vídeo, mostrar imagens, incorporar SWFs e objectos 3D, numa aplicação que executa no *browser*. O protótipo implementado detecta múltiplos ARMs em simultâneo e exhibe cubos coloridos dependendo do ARM encontrado (Ilustração 25). Foi criada uma classe AS3 que gere os ARMs encontrados e os respectivos eventos.

Tecnologias utilizadas no protótipo:

- Flash CS4 (AS3)
- FlarToolKit
- Papervision3D_2.1.920

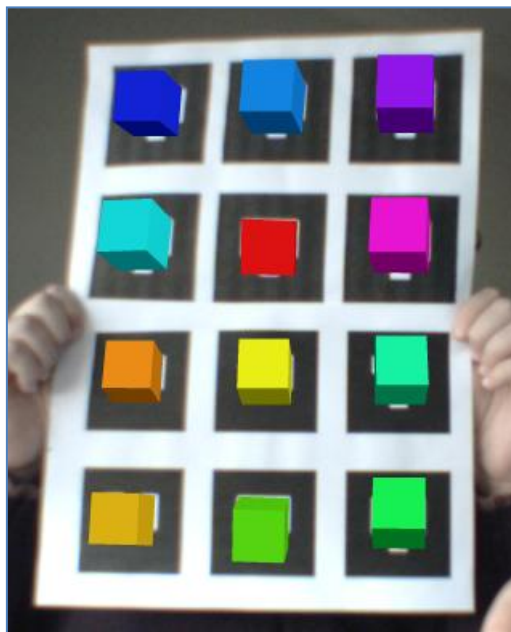


Ilustração 25 – Protótipo Flash

⁵ FlarToolKit is AS3 ported version of ARToolKit. Actually, FLARToolKit is based on NyARToolkit, Java ported version of ARToolKit. FLARToolKit recognize the marker from input image. and calculate its orientation and position in 3D world. te adquirida pela Adobe.

Website: <http://www.libspark.org/wiki/saqoosha/FLARToolKit/en>

3.3.2 Protótipo Unity 3D

O desenvolvimento em Unity tira partido da performance do motor 3D existente na plataforma e do reconhecimento de padrões que é realizado pela *plugin* “ARToolKit”⁶. Neste caso o protótipo teve que ser compilado num executável *Standalone* do Windows, uma vez que o *plugin* “ARToolKit” existente no Unity 3D necessita do GLUT instalado na máquina para executar. À semelhança do protótipo em Flash, foi criada uma classe em JavaScript que gere os ARMs encontrados e os respectivos eventos.

Tecnologias utilizadas no protótipo:

- Unity 3D V2.7 (windows)
- ArToolKit 2.72.1 (windows)
- Glut-3.7.6 (windows)

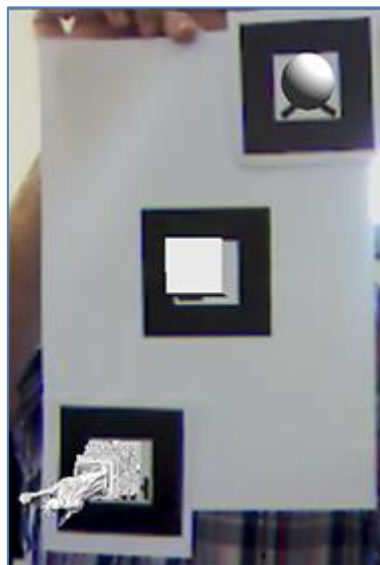


Ilustração 26 - Protótipo Unity 3D

⁶ ARToolKit is a software library for building Augmented Reality (AR) applications. These are applications that involve the overlay of virtual imagery on the real world.
Website: <http://www.hitl.washington.edu/artoolkit/>

3.3.3 Protótipo Flash + Unity 3D

Foi desenvolvido um protótipo que usa Flash e Unity 3D, este último tem um componente *Web* semelhante ao Flash Player que permite a integração via *browser*. Neste protótipo o Flash (à direita na Ilustração 27) processa o reconhecimento dos padrões através do FLAR Manager e envia para o Unity 3D (à esquerda na Ilustração 27) as coordenadas do padrão via JavaScript do *browser*, o Unity 3D processa as coordenadas e mostra os objectos 3D no ecrã.

O FlarToolKit foi substituído pelo FLAR Manager neste protótipo devido às vantagens que o último apresenta face ao primeiro. O FLAR Manager é um componente que gere os eventos e ARM detectados pelo FlarToolKit, esta camada de abstracção permite uma gestão dos ARMs mais eficiente e acrescenta novas funcionalidades (como *smoothing* entre duas detecções do mesmo ARM em posições diferente).

Tecnologias utilizadas no protótipo:

- Unity 3D V2.7 (windows)
- Flash CS4 (AS3)
- Papervision3D_2.1.920
- FLAR Manager 1.0.3

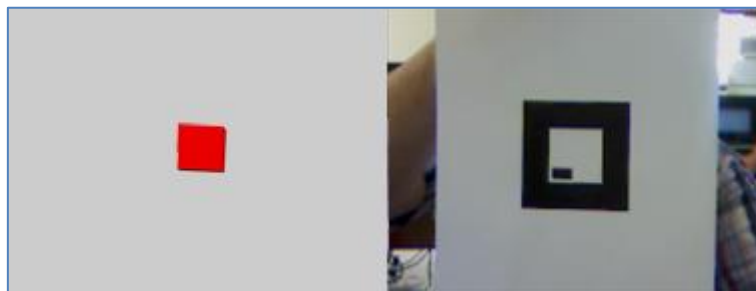


Ilustração 27 - Protótipo Flash + Unity 3D

3.4 Validação dos Protótipos

A validação dos protótipos foi realizada em conjunto com os dois potenciais clientes da aplicação final, pelos colaboradores dos departamentos de programação (quatro pessoas) e design (três pessoas) da Front-End, perfazendo uma amostra de onze pessoas inquiridas. Posteriormente aos testes realizados aos protótipos por parte dos utilizadores, foi entregue um questionário para avaliação dos mesmos, onde se classificou com os valores: “Muito Mau”, “Mau”, “Razoável”, “Bom” e “Muito Bom”, os seguintes tópicos:

- Facilidade de instalação – Considera-se “Razoável” se a utilizador conseguiu instalar a aplicação e necessitou de ajuda para finalizar o processo de instalação.
- Fluidez na apresentação dos objectos – Considera-se “Razoável” se os objectos apresentados piscam algumas vezes durante movimentação no ecrã.
- Aspecto gráfico dos objectos – Considera-se “Razoável” se os objectos desenhados apresentação poucas deformações e pontos negros resultantes do *render* do motor gráfico.
- Rapidez de detecção dos padrões – Considera-se “Razoável” se os padrões são detectados em menos de 2 segundos.
- Aspecto geral dos protótipos – Pergunta de confirmação das respostas dadas anteriormente
- Desempenho geral dos protótipos – Pergunta de confirmação das respostas dadas anteriormente

No momento da entrega do questionário cada utilizador era entrevistado com o intuito de verificar os valores indicados no questionário e perceber a razão do valor atribuído a cada tópico. As entrevistas pessoais permitiram completar as informações obtidas pelos resultados.

Os resultados dos questionários são apresentados no capítulo 3.6 Análise de Resultados e Produção de Conclusões.

3.5 Análise das Ferramentas de Desenvolvimento

A principal ferramenta de desenvolvimento utilizada na programação das linguagens PHP, HTML, CSS e Javascript foi o IDE PhpEdit 4 e o design da aplicação foi criado em PhotoShop. A utilização destas ferramentas a nível profissional possibilitou que o desenvolvimento decorresse sem contratemplos.

As primeiras versões dos componentes em Flash (AS3) foram compiladas com o Adobe CS4. Com o aumento da complexidade e recursos utilizados nos componentes, maior era o tempo de compilação (em média 5 minutos), o que se tornou num inconveniente à celeridade do desenvolvimento do projecto.

Na tentativa de resolver esta situação, o *player* de vídeo que era instanciado na *timeline* do editor do Flash CS4, foi exportado para uma biblioteca SWC e passou a ser incluído por código. O ficheiro FLA ficou apenas como projecto para iniciar a compilação do código AS3 e esta alteração reduziu o tempo de compilação para 4 minutos. Esta situação levantou algumas questões que conduziram a uma pesquisa para resolver este problema.

Com o lançamento do Adobe CS5, o projecto passou a ser compilado com esta versão, mas o tempo de compilação mantinha-se extenso. Após alguma investigação e com ajuda dos documentos técnicos da Adobe, chegou-se à conclusão que existia um problema nas Creative Suites quando era incluído um número elevado de *imports* e de bibliotecas em simultâneo por uma classe externa ao ficheiro FLA do projecto.

Perante esta situação, optou-se por testar a compilação dos componentes Flash no IDE FlashDevelop configurado para utilizar o Flex SDK v4 da Adobe na compilação. O tempo de cada compilação diminuiu para uns surpreendentes 10 segundos e, com o sucesso nos resultados obtidos os componentes passaram a ser compilados neste IDE.

Os componentes Unity 3D dos protótipos foram compilados com o IDE Unity 3D V2.7. A principal linguagem de programação foi o Javascript para animação e posicionamento dos objectos, no entanto o interface de comunicação com detecção de padrões (ARToolKit) foi programado em C# que utiliza um DLL (programado C/C++). No Unity 3D existe a possibilidade programar em Javascript, C# e Boo Script, à semelhança do dotNET, na compilação, o código é convertido para uma linguagem intermédia. De referir que Javascript utilizado no Unity 3D não é igual ao Javascript utilizado nos *browsers*, uma vez que as classes obedecem às regras da programação orientada a objecto clássica, em vez do método de prototipagem usado nos *browsers*.

À semelhança do Flash, as aplicações em Unity 3D, compiladas para o componente *Web* também usam o Javascript do *browser* e podem efectuar chamadas a funções internas ao componente. Esta conformidade de funcionalidades torna estas duas tecnologias nas mais viáveis para cumprir os objectivos propostos.

3.6 Análise de Resultados e Produção de Conclusões

A análise de resultados dos testes aos protótipos, em conjunto com os questionários e entrevistas realizadas aos utilizadores, permitiram identificar qual a tecnologia de desenvolvimento da aplicação final.

O questionário produziu o seguinte quadro resumo da avaliação dos protótipos:

Tabela 1 – Resumo dos resultados da avaliação dos protótipos

	Flash	Unity 3D	Unity 3D + Flash
Facilidade de instalação	Muito Bom	Razoável	Bom
Fluidez na apresentação dos objectos	Bom	Muito Bom	Muito Bom
Aspecto gráfico dos objectos	Bom	Muito Bom	Muito Bom
Rapidez de detecção dos padrões	Bom	Bom	Razoável
Aspecto geral do protótipo	Bom	Bom	Bom
Desempenho geral protótipo	Bom	Bom	Razoável

Numa primeira análise, os resultados são bastantes equilibrados entre os 3 protótipos, sendo necessário esclarecer o que os utilizadores consideram como “Razoável” e “Muito Bom”, pois são os valores que iram ter mais peso na decisão final.

Tendo em consideração o *feedback* obtido pelas entrevistas aos utilizadores, foi realizada uma análise global a cada um dos protótipos que resultou nas seguintes conclusões:

Protótipo Flash

Os utilizadores consideraram a instalação do Flash Player fácil ou não sendo necessária na maior parte dos casos. Segundo as estatísticas⁷ o Flash Player está instalado em 99% dos computadores com internet, o que facilita o acesso à aplicação por parte dos utilizadores.

Existe um conjunto de pequenos problemas relacionados com a performance quando se usam objectos 3D, uma das principais razões deve-se ao Flash usar o CPU em vez do GPU da placa gráfica para o processamento 3D dos objectos.

⁷ Fonte: http://www.adobe.com/products/player_census/flashplayer/ [2011/09/08]

Protótipo Unity 3D

Apesar da boa performance na visualização dos objectos 3D, os utilizadores (com menos conhecimentos técnicos) encontraram algumas dificuldades na instalação do GLUT, uma vez que é necessário colocar o DLL do GLUT na pasta “System32” do Windows para a aplicação funcionar. Este protótipo não funciona correctamente quando compilado para o componente *Web* do Unity 3D porque o ARToolKit necessita de ter acesso às bibliotecas do DLL do GLUT a partir do *browser*.

Protótipo Unity 3D + Flash

Os utilizadores consideraram o aspecto gráfico do componente *Web* do Unity 3D tão bom como o da aplicação Windows. No entanto, acharam estranho a detecção de padrões (Flash) estar separada da apresentação dos objectos 3D (Unity 3D). Mesmo tendo mais qualidade, os objectos 3D congelavam muitas vezes no ecrã, o que à primeira vista parecia ser um problema do Flash ou do Unity 3D. Concluiu-se assim, ser um problema no número de chamadas por segundo às funções de Javascript feitas pelo Flash, o número era tão elevado que o *browser* não tinha tempo de processar todas as chamadas e estas acabavam por ser descartadas, isto demonstra que os objectos não eram actualizados no componente *Web* do Unity 3D.

Conclusões do desenvolvimento dos protótipos

Um dos objectivos principais é que a aplicação funcione no modelo Cliente-Servidor, ficando assim excluída a possibilidade do desenvolvimento exclusivo em Unity 3D devido às limitações anteriormente expostas.

Outro objectivo passa pela integração com outras aplicações via *browser*, a solução Unity 3D + Flash apresenta problemas de performance, dada a visualização dos objectos 3D não ser realizada no mesmo plano que os ARM (são componentes diferentes) e no pior dos casos os utilizadores terem que instalar dois *players* diferentes, sendo também remetidos para *sites* distintos, o que pode causar alguma confusão aos utilizadores.

Por exclusão de hipóteses, a aplicação será desenvolvida só em Flash e com o FLAR Manager como *Framework* de detecção dos ARMs. O BackOffice será implementado em PHP com a Zend Framework e o servidor de base de dados em MySQL.

A selecção destas tecnologias teve em consideração as ferramentas de desenvolvimento utilizadas na Front-end, uma vez que potenciam o cumprimento dos objectivos propostos integrando-se no âmbito da empresa. Na perspectiva dos utilizadores finais, estas tecnologias minimizam as dificuldades de “instalação” das aplicações, uma vez que o Flash é *cross-platform* e é uma tecnologia que os utilizadores estão habituados a utilizar, o que possibilita uma harmonia entre funcionalidade, escalabilidade e performance.

4 Implementação

Este capítulo descreve a implementação da solução desenvolvida, as linguagens de programação utilizadas, as ferramentas de desenvolvimento para produzir a aplicação e as partes do código mais relevantes da solução.

4.1 Identificação dos requisitos

O levantamento dos requisitos foi conduzido tendo em conta os dois grupos de utilizadores que vão trabalhar com a aplicação, os clientes finais e os programadores das empresas.

Após a validação da tecnologia a ser utilizada na aplicação intitulada Magic Catalog, foi necessário perceber o que os utilizadores esperavam da aplicação que ia gerir os conteúdos multimédia, designado por ARCMS – *Augmented Reality Content Management System*, do Magic Catalog. Neste sentido foram realizadas três reuniões, uma com programadores, outra com clientes e uma final com ambos os grupos.

A reunião de programadores teve como objectivo traçar os detalhes técnicos e funcionalidades que se esperavam da aplicação. Nesta reunião participaram programadores com e sem conhecimentos técnicos das tecnologias envolvidas, este misto de valências foi extremamente importante, uma vez que um dos objectivos da aplicação é encapsular o processo da implementação do sistema AR.

Na reunião com os clientes foi focada a apresentação gráfica e funcionalidades presentes na aplicação. Desta reunião surgiu um dos objectivos proposto, que se centra em conseguir pré-visualizar os conteúdos sem a necessidade de abrir o Magic Catalog Viewer.

A reunião em conjunto serviu para confirmar os requisitos propostos por cada grupo, confrontar ideias, resolver conflitos de requisitos e identificar funcionalidades inatingíveis para o período de implementação.

Os requisitos identificados foram:

- A aplicação para um determinado cliente, deverá ter um acesso centralizado e *online*.
- A interface do ARCMS deve ser simples intuitiva.
- A Gestão de Utilizadores do ARCMS deve:
 - Permitir inserir, editar e eliminar utilizadores.
 - Atribuir diferentes perfis de acesso à aplicação.
 - Registar a actividade de cada utilizador.
- A Gestão de Marcadores deve:
 - Inserir, editar, eliminar e ordenar os marcadores.
 - Permitir editar a imagem que identifica o marcador.
 - Permitir editar o ficheiro que contém a informação do padrão da imagem utilizada.
 - Possibilitar o *download* da imagem e do padrão do ARM
- A Gestão de Objectos Multimédia deve:
 - Inserir, editar, eliminar e ordenar os objectos multimédia
 - Permitir a pré-visualização do objecto sem a necessidade de publicar as alterações no Magic Catalog.
 - Associar um marcador ao objecto multimédia.
 - Editar a Escala e Rotação do Objecto no ambiente 3D.
 - O objecto multimédia deve suportar ficheiro do tipo JPG, PNG, GIF, SWF, FLV e ZIP (com os ficheiros do modelo 3D em formato COLLADA).
- A Gestão de Catálogos do ARCMS (entenda-se por catálogo, uma conjunto ordenado de objectos multimédia, onde é associado um marcador composto por uma imagem e o ficheiro com o respectivo padrão) deve:
 - Publicar a informação dos objectos multimédia do catálogo, para o Magic Catalog.
 - Informar quando um catálogo não está sincronizado com o Magic Catalog.
 - Pré-visualizar a informação dos objectos multimédia que vão alimentar o Magic Catalog (só para administradores).
 - Gerar o código necessário para carregar um novo catálogo ao Magic Catalog (só administradores).
 - Permitir configurações específicas do Magic Catalog consoante o catálogo em utilização (só para administradores).
- Possibilidade de Integração com outras aplicações *Web* através de uma API
 - O código da API deve ser incluído quando é gerado o catálogo.

4.2 Descrição Técnica

As principais ferramentas de produção utilizadas para desenvolver a solução final foram:

- PHPEdit V4.0 para edição dos ficheiros PHP, HTML, CSS, Javascript e XML.
- Adobe Flash CS5 e o FlashDevelop V3.3.4 para editar e compilar os ficheiros AS3.
- PhpMyAdmin para gestão da base de dados MySQL.
- Adobe Photoshop CS5 para desenhar a interface da aplicação.

O Magic Catalog foi desenvolvido para funcionar em ambiente Windows e Linux, a produção foi realizada em ambiente Windows e cada nova subversão era publicada para o ambiente Linux.

Aspectos técnicos dos servidores utilizados:

- Servidor Windows (WAMP)
 - Sistema Operativo Windows 7 SP1
 - Apache 2.2.11
 - PHP 5.3.0
 - GD 2 (para manipulação de imagens)
 - PDO (para acesso a base de dados pela Zend Framework)
 - ZIP
 - MySQL 5.1.36
 - Zend Framework 1.10 (biblioteca como sistema de ficheiros no projecto)
- Servidor Linux (LAMP)
 - Ubuntu 10.04 LTS
 - Apache 2.2.14
 - PHP 5.3.2
 - GD 2 (para manipulação de imagens)
 - PDO (para acesso a base de dados pela Zend Framework)
 - ZIP
 - MySQL 5.1.4
 - Zend Framework 1.11 (biblioteca como extensão do PHP no servidor)

4.2.1 Magic Catalog

No desenvolvimento deste projecto foram envolvidas múltiplas tecnologias para obter o resultado desejado. A Ilustração 28 apresenta um esquema abstracto das tecnologias usadas nos componentes que constituem o Magic Catalog.

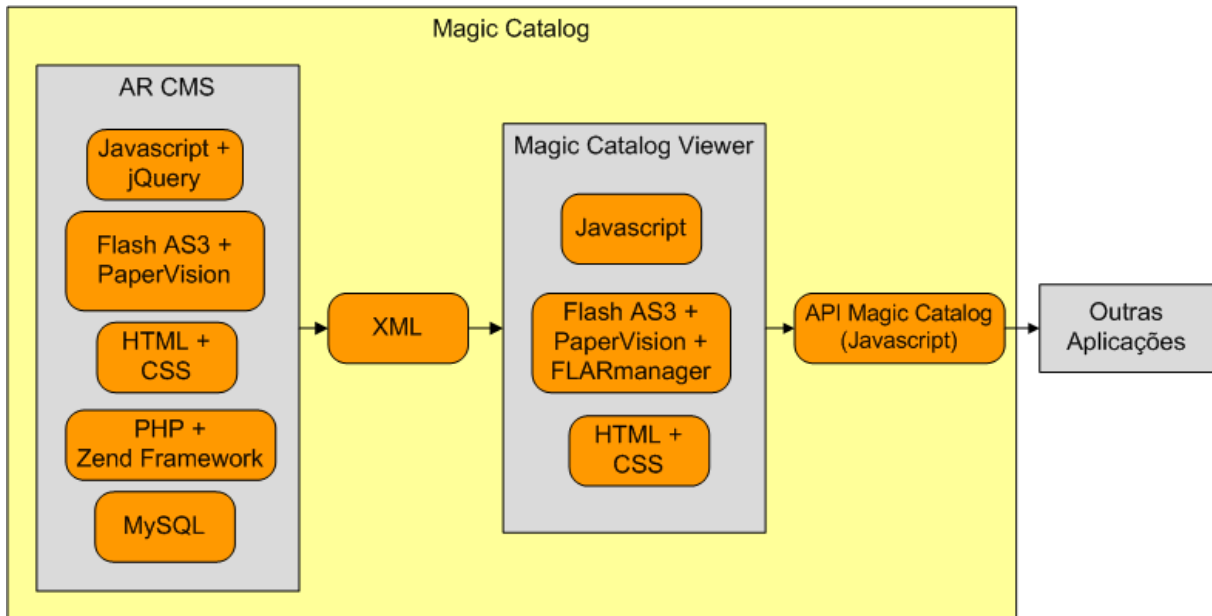


Ilustração 28 - Tecnologias/Linguagens de programação utilizadas

O Magic Catalog é constituído pelo Magic Catalog Viewer e pelo ARCMS. O Magic Catalog Viewer é o *FrontOffice* da aplicação que detecta os ARMs utilizados pelos utilizadores e apresenta os conteúdos multimédia associados. O ARCMS é o *BackOffice* que gere os conteúdos que alimentam o Magic Catalog Viewer. A API Javascript do Magic Catalog permite a integração com outras aplicações via *browser*.

Como o Magic Catalog Viewer é alimentado por XML é possível instalar o Magic Catalog sem o ARCMS, suprimido assim a gestão dos conteúdos, no entanto pode-se instalar o ARCMS a posteriori caso seja necessário.

4.2.2 Magic Catalog Viewer

O Magic Catalog Viewer (Ilustração 29) é um componente Flash AS3 (SWF) que necessita da *plugin* Flash Player 10 para funcionar no *browser*. É responsável por capturar as imagens da *webcam*, processá-las identificando o ARM e a sua posição espacial, seguidamente procura no XML qual o conteúdo multimédia associado e apresenta no ecrã o resultado. Os eventos lançados durante a detecção dos ARMs são enviados para a API Javascript (ver capítulo 4.2.3) para que seja possível efectuar integrações com outras aplicações que corram em simultâneo no *browser*.

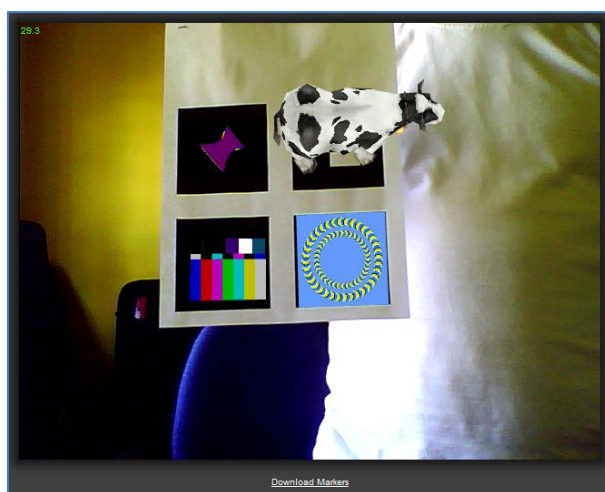


Ilustração 29 – Magic Catalog Viewer

Este componente foi compilado para SWF através do editor FlashDevelop, configurado para usar o Adobe Flex SDK V4. O SWF é embebido no HTML pelo método SWF Object e o identificador do catálogo a ser carregado é enviado por FlashVars para o Magic Catalog Viewer (ver Excerto de código 1). O código HTML para visualizar cada catálogo pode ser gerado automaticamente pelo ARCMS.

```

<script language="JavaScript" type="text/javascript">
    var hasRightVersion = DetectFlashVer(requiredMajorVersion, requiredMinorVersion,
requiredRevision);
    if(hasRightVersion) { // if we've detected an acceptable version
        // embed the flash movie
        AC_FL_RunContent(
            'codebase',
'http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=10,0,0,0',
            'width', '100%',
            'height', '100%',
            'src', 'magic_catalog',
            'quality', 'high',
            'pluginspage', 'http://www.adobe.com/go/getflashplayer',
            'align', 'top',
            'play', 'true',
            'loop', 'true',
            'scale', 'showall',
            'wmode', 'window',
            'devicefont', 'false',
            'id', 'magic_catalog',
            'bgcolor', '#000000',
            'name', 'magic_catalog',
            'menu', 'true',
            'allowFullScreen', 'false',
            'allowScriptAccess', 'sameDomain',
            'movie', 'magic_catalog',
            'salign', 'lt',
            'flashvars', 'catalog=music'
        ); //end AC code
    } else { // flash is too old or we can't detect the plugin
        var alternateContent = 'Alternate HTML content should be placed here.'
            + 'This content requires the Adobe Flash Player.'
            + '<a href="http://www.adobe.com/go/getflashplayer/">Get Flash</a>';
        document.write(alternateContent); // insert non-flash content
    }
</script>

```

Excerto de código 1 – Inclusão do Magic Catalog Viewer pelo método SWF Object

O Magic Catalog Viewer necessita de ter acesso a uma *webcam* para começar a detectar os ARMs e para mostrar os objectos multimédia no ecrã. Internamente o componente está dividido nos seguintes subsistemas:

- FLARManager v1.1 (que usa o FlarToolkit) - Detecção e posicionamento espacial do ARM.
- Motor gráfico Papervision3D v2.1.932 – Para apresentação espacial dos objectos multimédia.
- Motor Magic Catalog Viewer – Motor da aplicação que interliga todos os outros subsistemas.
- Notificador de Eventos da API Javascript – Conjunto de funções que fazem parte do motor da aplicação e que enviam a informação dos ARMs detectados para API Javascript.

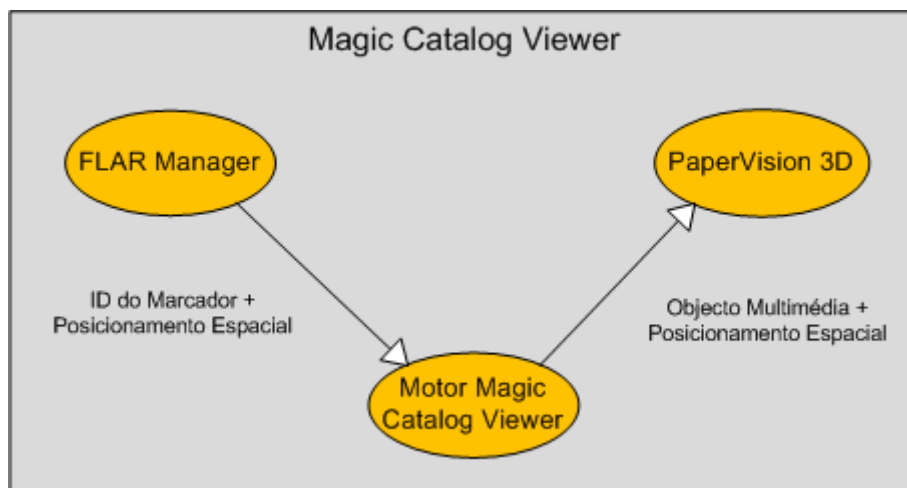


Ilustração 30 - Subsistemas Magic Catalog Viewer

A Ilustração 30 resume o método de criação de realidade aumentada do Magic Catalog Viewer. O FLAR Manager é o primeiro processo a ser iniciado, carrega a informação dos padrões dos ARMs e atribui um identificador único. Os dados são carregados a partir de um XML de configuração (ver Excerto de código 3 página 51), que entre outras informações, contém a definição sobre a confiança da detecção (parâmetro *minConfidence*), esta propriedade define qual a percentagem mínima necessária para assumir que o padrão detectado é um ARM conhecido da aplicação. Se excluirmos as condições ambientais e de iluminação, devemos ponderar o aumento da percentagem de confiança, quanto maior for o número de marcadores presentes no catálogo. Assim asseguramos que existe uma maior probabilidade que o padrão encontrado pertence ao ARM identificado. Outra propriedade relevante é a suavidade dos movimentos dos objectos (parâmetro *smoothing*), que representa o número de coordenadas enviadas entre duas detecções. Quando um ARM é movido rapidamente, as coordenadas intermédias são calculadas por uma previsão linear, esta opção permite minimizar o aparecimento e desaparecimento “brusco” dos objectos no ecrã. Algumas das opções disponíveis no XML de configuração podem ser editadas no ARCMS, onde cada catálogo tem o seu próprio ficheiro de configuração.

A posição espacial e o identificador do ARM são enviados para o motor do Magic Catalog Viewer, que é responsável por carregar os objectos multimédia e desenhá-los no ecrã com o motor PaperVision 3D, que em simultâneo, notifica a API Javascript com os respectivos eventos.

O motor do Magic Catalog Viewer tem um algoritmo de previsão que carrega dinamicamente os objectos desenhados pelo PaperVision 3D. O algoritmo permite balancear o volume de informação lida e carregar os dados à medida que são necessários. Este algoritmo rege-se pelo seguinte princípio, se um padrão de um ARM foi identificado, é provável que o próximo

ARM seja o marcador do objecto anterior ou posterior ao marcador do objecto actual encontrado da lista de objectos do catálogo (ver Ilustração 31). Dito de uma forma prática o algoritmo tenta carregar três objectos de uma só vez, se estes ainda não foram carregados, na tentativa de minimizar o tempo de carregamento entre objectos adjacentes ao objecto que está a ser visualizado. O algoritmo foi elaborado, tendo o cuidado, para não implicar um peso acrescido na performance do sistema.

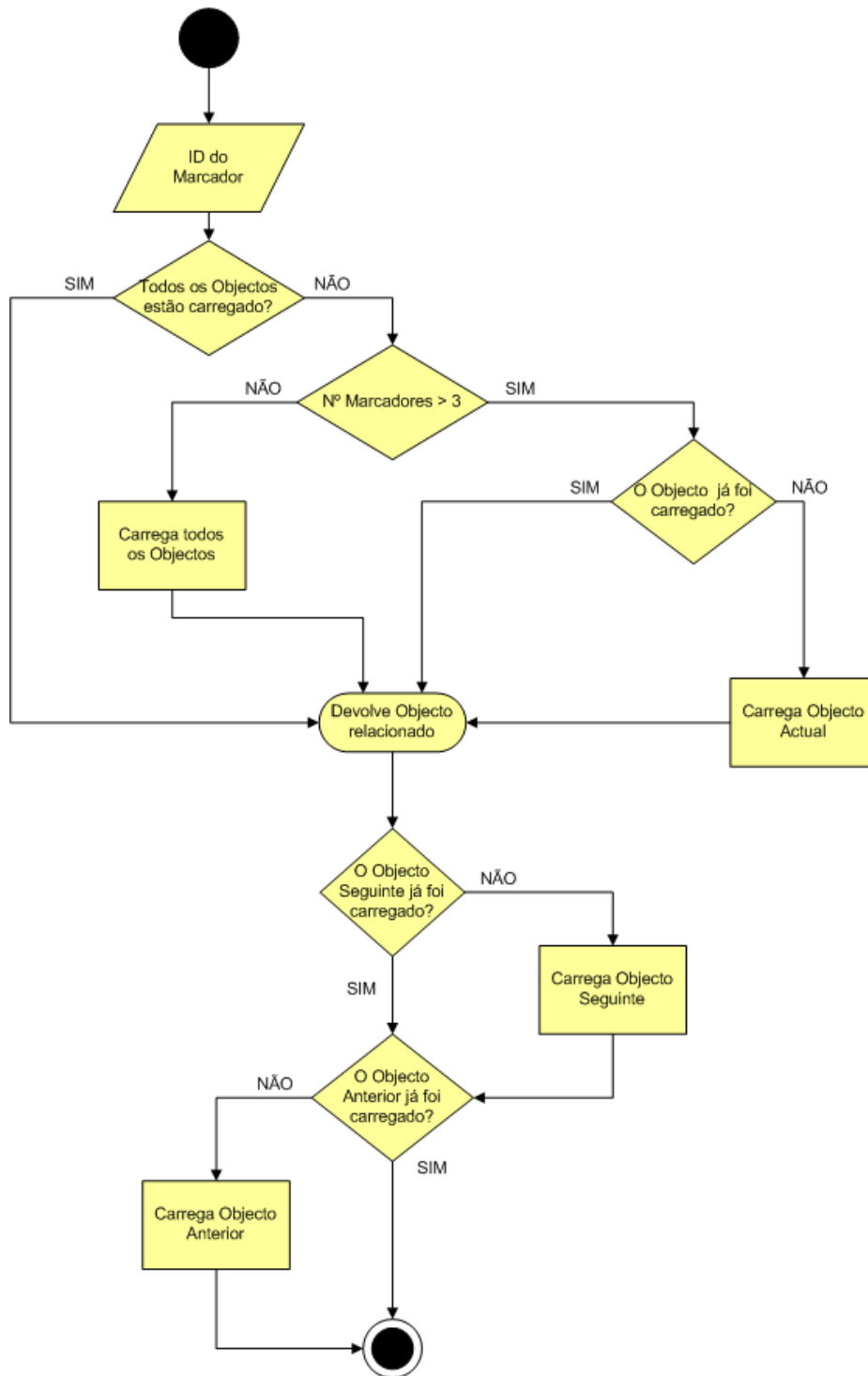


Ilustração 31 – Algoritmo de carregamento de Objectos Multimédia

Os eventos de notificação da API Javascript são despoletados pelo motor apenas quando o SWF é executado no *browser*. Como o *browser* não consegue tratar a elevada quantidade de eventos disparados em simultâneo pelo motor do Magic Catalog Viewer, foi necessário implementar uma fila de espera que chama as funções em intervalos de tempos distintos. No entanto, foi retirada a função que notifica as alterações de posição do ARM, uma vez que esta notificação era executada centenas de vezes por segundo enchendo a fila de eventos de actualização. Como as funções são disparadas em intervalos de tempos fixos, origina um atraso significativo entre os eventos de adição e remoção um ARM, o que implica que as funções não são chamadas em tempo real, criando assim um problema na integração com outras aplicações.

O motor do Magic Catalog Viewer foi implementado para otimizar a visualização dos objectos multimédia e em conjunto com FLAR Manager, que gere os ARMs, possibilita que a aplicação corra suavemente dependendo apenas de factores externos para funcionar correctamente. Alguns destes factores são: a qualidade dos ARMs impressos, as características da máquina do utilizador, a velocidade de ligação a internet (quando aplicável) e das condições ambientais em que a aplicação é utilizada.

4.2.3 API Javascript Magic Catalog

A API Javascript é um objecto javascript que é chamado pelo Magic Catalog Viewer sempre que existe um evento de adição ou remoção de um ARM.

```
MagicCatalog = {
  addMarker: function(marker_id) {
    //alert('addMarker:'+marker_id);
  },
  removeMarker: function(marker_id) {
    //alert('removeMarker:'+marker_id);
  }
}
```

Excerto de código 2 – API Javascript

A API Javascript expõe duas funções que os programadores podem usar:

- `addMarker` – recebe como parâmetro o ID do ARM detectado e é lançada sempre que um marcador novo é detectado.
- `removeMarker` – recebe como parâmetro o ID do ARM quando este deixa de ser detectado.

A função `updateMarker` tinha sido exposta na API, mas por razões de performance foi retirada devido a fazer sucessivas chamadas num curto intervalo de tempo. Esta situação impedia que o *browser* conseguisse tratar todos os eventos, e na maior parte dos casos os

eventos de adicionar e remover os ARMs não eram processados pelo *browser* colocando em risco a funcionalidade da API.

4.2.4 ARCMS

O ARCMS é o gestor de conteúdos do Magic Catalog, a partir deste *BackOffice* é possível criar vários catálogos de objectos multimédia para serem utilizados no Magic Catalog Viewer, criar novos marcadores, editar e pré-visualizar os objectos multimédia antes de os publicar na aplicação.

O ARCMS foi programado em PHP 5, foi desenvolvido para funcionar nos *browsers* mais recentes e está optimizado para ecrãs de resolução 1024x768. É compatível com o Firefox 3.0+, Google Chrome 4.0+, Safari 3.1+, Internet Explorer 9.0+ e Opera 10.5+. Funciona com o Internet Explorer 7 e 8 mas existe algumas limitações gráficas impostas pelo uso de CSS3 nos estilos das páginas HTML, que pode causar alguns problemas de funcionalidade. O Javascript do *browser* tem que estar activo para que o SWF Object consiga incluir objectos Flash e para seja possível realizar pedidos AJAX nos formulários dos módulos. Os pedidos AJAX são realizados pelos métodos disponibilizados na API jQuery, que permitem a compatibilidade *cross-browser* e simplificam a escrita do código das funções.

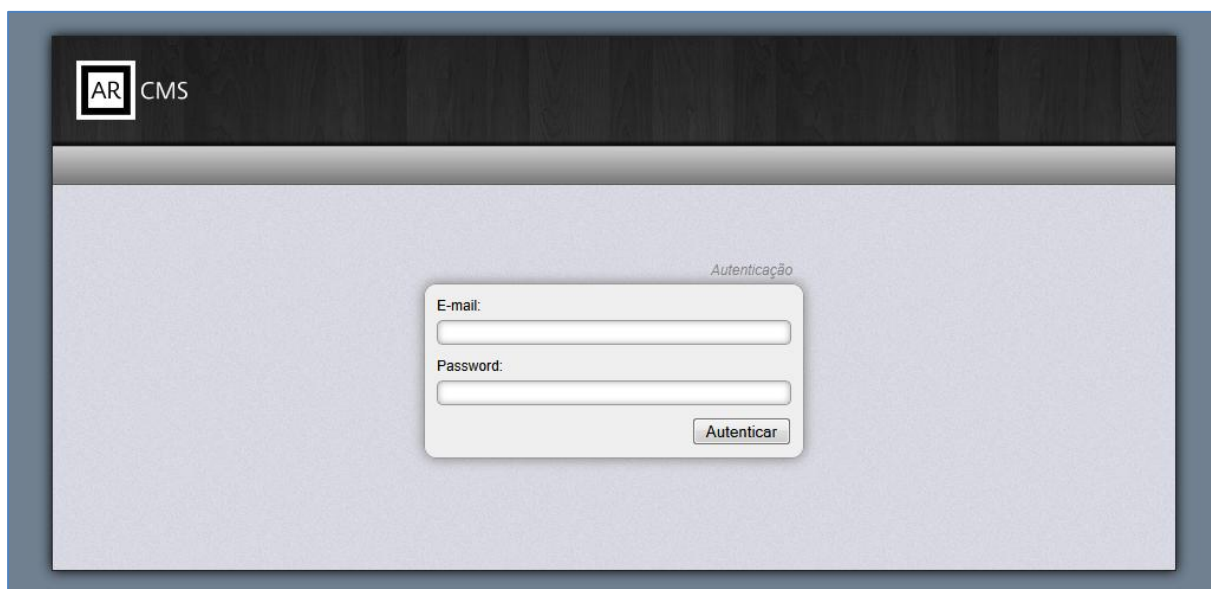


Ilustração 32 - ARCMS Autenticação

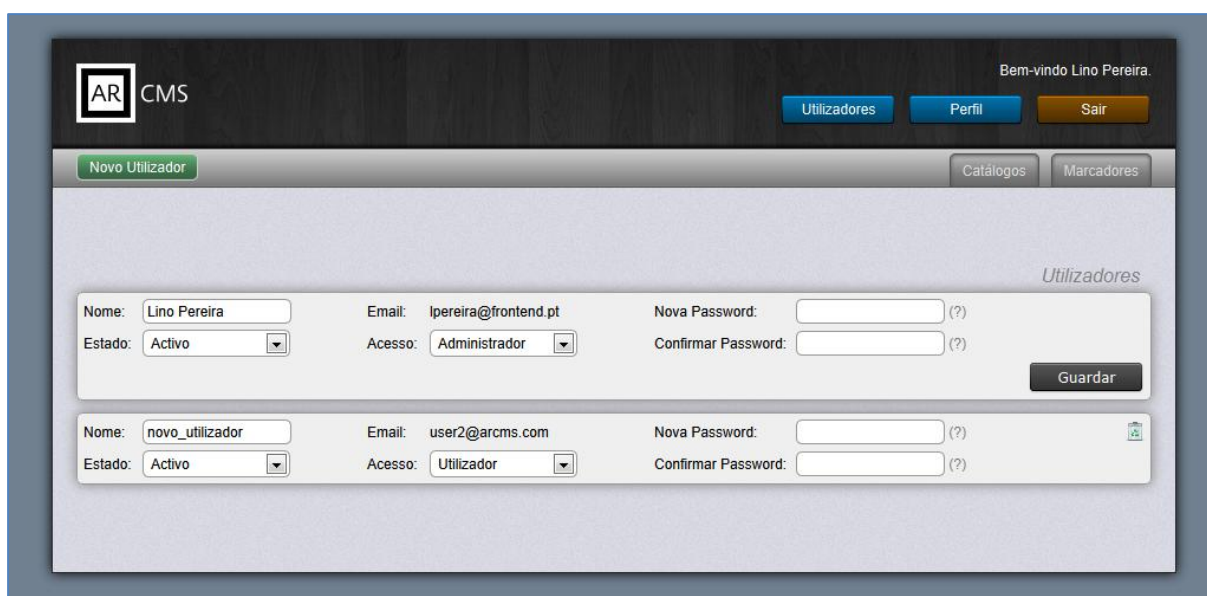
O *BackOffice* ARCMS está dividido em 3 módulos principais:

- Gestão de Utilizadores
- Gestão de Marcadores
- Gestão de Catálogos

Gestão de Utilizadores

Neste módulo (ver Ilustração 33) é possível gerir as credenciais dos utilizadores do sistema e o seu nível de acesso. Existem 3 níveis de acesso:

- Administrador de Sistema – Tem acesso total ao sistema, não aparece na listagem dos utilizadores, nem é possível criar este acesso a partir do *BackOffice*. Tem como propósito a realização de testes de funcionalidade e a recuperação das contas de administradores que ficaram inacessíveis.
- Administrador – Tem acesso a gestão de utilizadores e a todas as funcionalidades disponíveis nos outros módulos.
- Utilizador – Tem acesso aos módulos de Gestão de Marcadores e de Catálogos. No entanto, não tem permissão para alterar as configurações avançadas do catálogo, nem para gerar o código de inclusão do catálogo numa nova página HTML.



The screenshot displays the 'Utilizadores' (Users) management interface in the ARCMS system. At the top, the 'AR CMS' logo is on the left, and the user 'Lino Pereira' is logged in, with a 'Bem-vindo Lino Pereira.' message. Navigation buttons for 'Utilizadores', 'Perfil', and 'Sair' are visible. Below the navigation, there are tabs for 'Novo Utilizador', 'Catálogos', and 'Marcadores'. The main content area is titled 'Utilizadores' and contains a form for adding a new user. The form has two rows of input fields. The first row is for 'Lino Pereira' with email 'lpereira@frontend.pt' and access level 'Administrador'. The second row is for 'novo_utilizador' with email 'user2@arcms.com' and access level 'Utilizador'. Each row includes fields for 'Nome', 'Email', 'Estado' (Active), 'Acesso', 'Nova Password', and 'Confirmar Password'. A 'Guardar' button is located at the bottom right of the form.

Ilustração 33 – ARCMS Gestão de Utilizadores

Gestão de Marcadores

O módulo de gestão de Marcadores (ver Ilustração 34) permite adicionar novos marcadores para serem utilizados nos catálogos. A aplicação é instalada com 64 Marcadores pré-configurados, mas é possível criar marcadores personalizados. Para isso, cada marcador tem que ter uma imagem PNG (ARM) que o identifica e o respectivo ficheiro com padrão da imagem. O ficheiro com o padrão da imagem pode ser gerado na ferramenta online “ARToolKit Marker Generator Online Multi” no site <http://flash.tarotaro.org/blog/2009/07/12/mgo2/> [2011-09-18] ou transferir a versão AIR desta ferramenta *site* <http://saqoosha.net/lab/FLARToolKit/MarkerGenerator/MarkerGenerator.air> [2011-09-18].

Apesar do processo de criação de um marcador ser relativamente simples, este é uma das peças fundamentais para o correcto funcionamento do catálogo construído pelo utilizador, uma vez que ficheiro com o padrão da imagem é a assinatura visual do objecto multimédia que vamos associar no catálogo.

Na sua essência, este módulo, é um repositório de marcadores que podem ser reutilizados nos diversos catálogos.

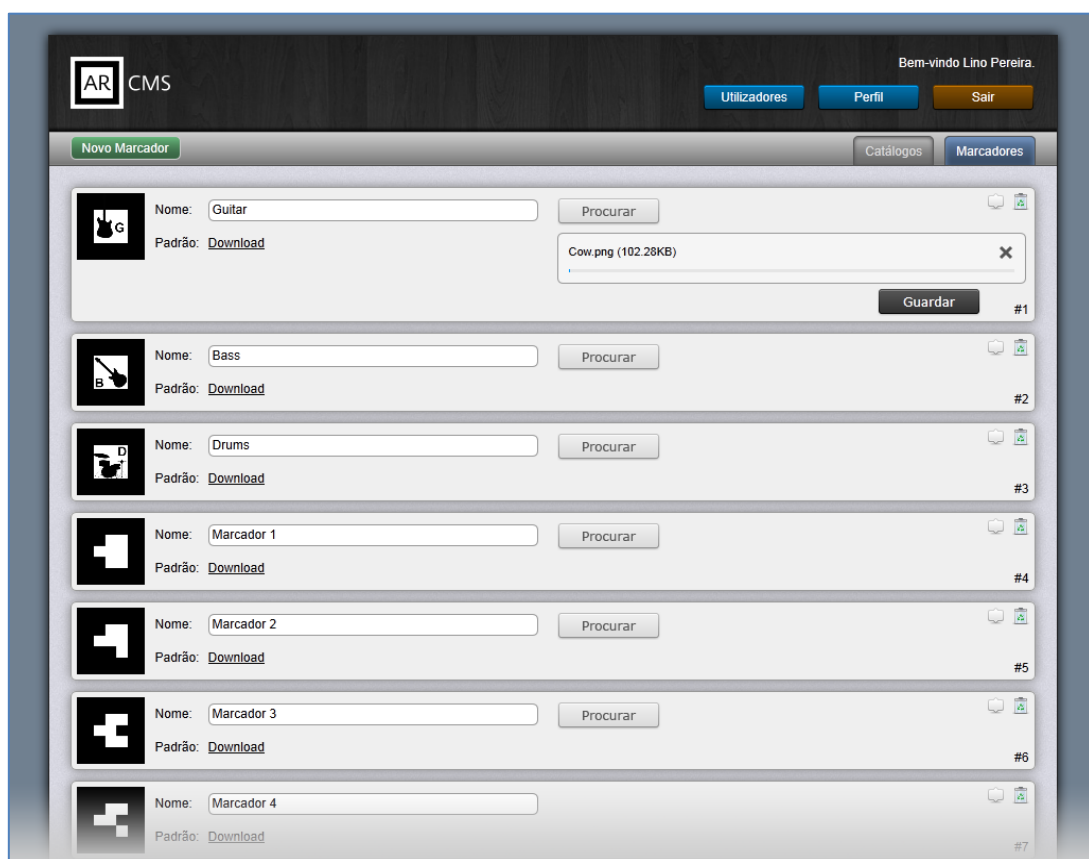


Ilustração 34 – ARCMS Gestão de Marcadores

Gestão de Catálogos

O gestor de catálogos (ver Ilustração 35) é o módulo central da aplicação sendo responsável pela agregação da informação associada aos marcadores e aos objectos dos catálogos, para que no final, cada catálogo possa ser usado individualmente no Magic Catalog.

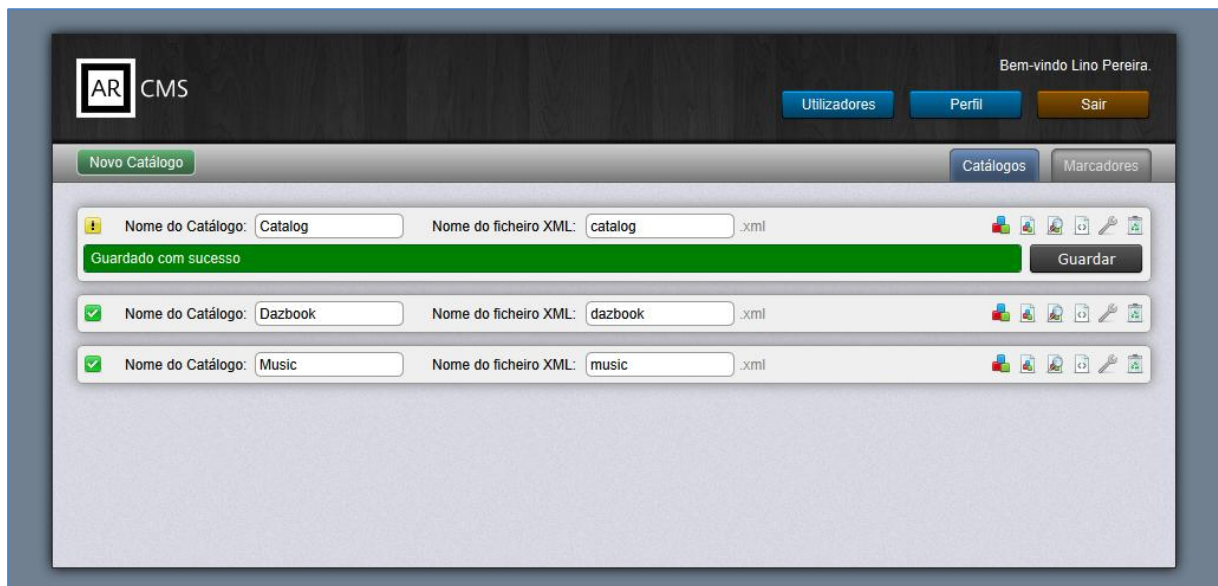


Ilustração 35 – ARCMS Gestão de Catálogos

Neste gestor é possível verificar se o XML do catálogo está desactualizado em relação à informação guardada na base de dados e consultar o XML dos objectos multimédia. Os utilizadores com permissões podem aceder e editar às configurações avançadas (ver Ilustração 36) que são guardadas no ficheiro de configuração de cada catálogo (ver Excerto de código 3 – XML de configuração).

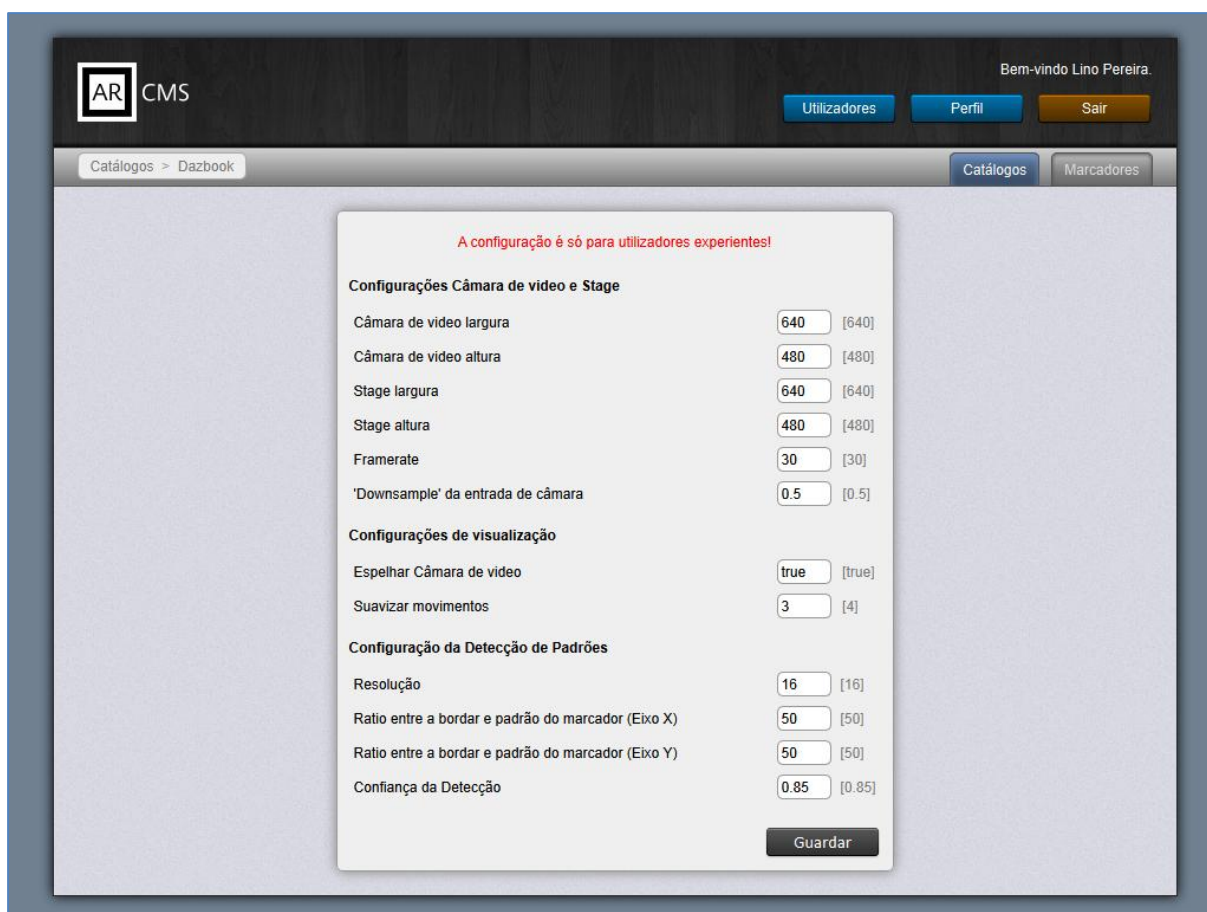


Ilustração 36 – ARCMS Configuração avançada do catálogo

Existe a possibilidade de gerar automaticamente o código HTML com a informação do Magic Catalog, neste código estão contidas todos os parâmetros para carregar o catálogo no Magic Catalog Viewer e as funcionalidades da API Javascript para integração com outras aplicações.

A vista de gestão de objectos multimédia do catálogo (ver Ilustração 37) permite editar as propriedades de cada objecto. Cada objecto tem um marcador associado e um tipo de media, os objectos podem ser Imagens (jpg, gif e png), Vídeos (fla), Objectos Flash (swf) e Modelos 3D COLLADA. Como os modelos COLLADA são normal constituídos por vários ficheiros e pastas, é necessário Zipar o modelo antes de submeter no formulário, facilitando assim o envio do ficheiro e processamento dos ficheiros.

O MC Media Viewer permite alterar a rotação e escala dos objectos imediatamente após o seu envio no formulário. A dimensão dos objectos é calculada automaticamente, excepto os modelos COLLADA, com recurso à extensão GD do PHP e às classes que lêem a meta informação dos ficheiros.

Através da funcionalidade de arrastar com o rato é possível ordenar os objectos. A ordem dos objectos é responsável pela atribuição do ID do ARM quando carregado no Magic Catalog Viewer. Na óptica do programador, o ID do ARM é igual à ordem do objecto menos um, com este identificador e com a API Javascript os programadores podem estender as funcionalidades base do Magic Catalog.

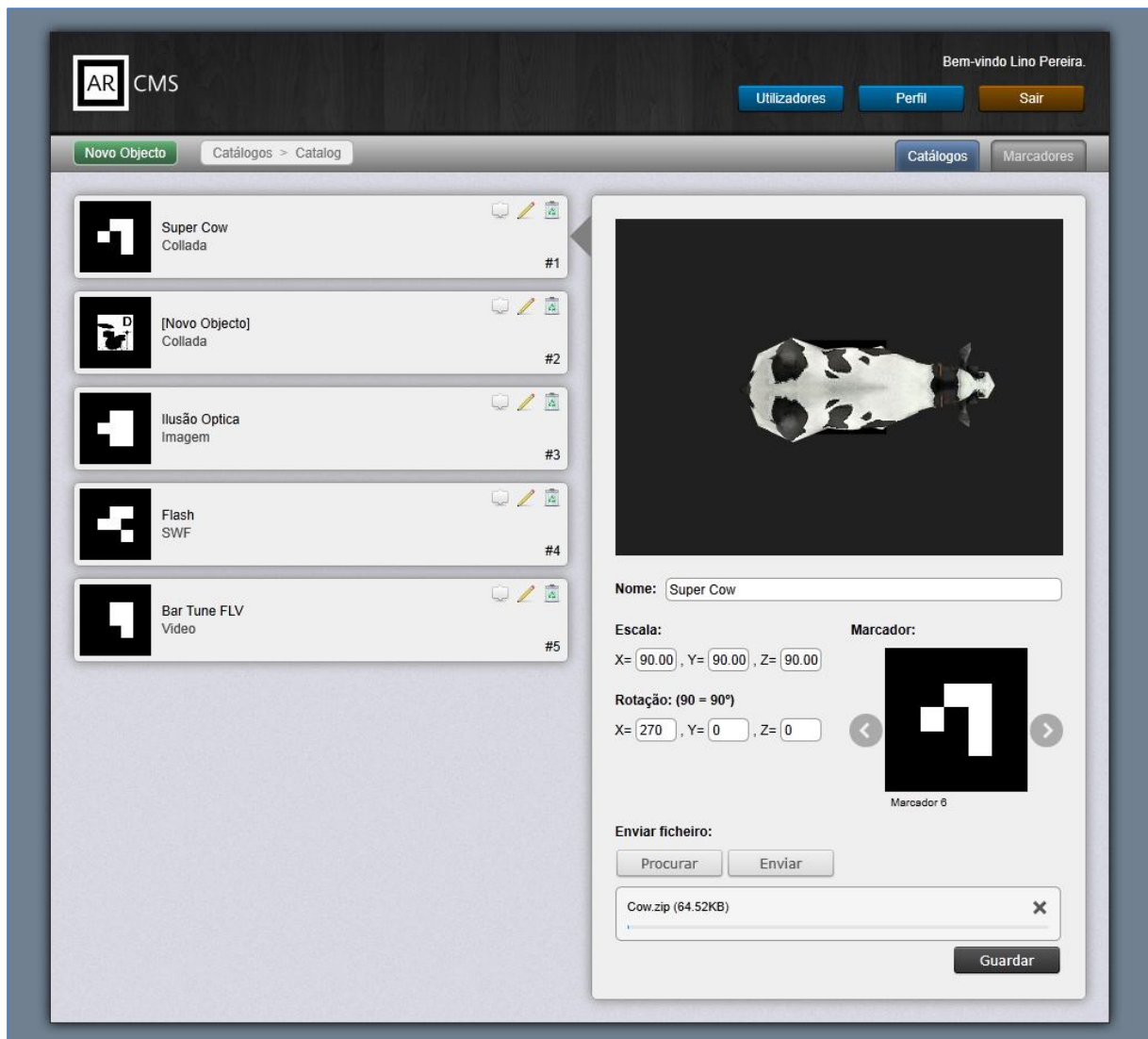


Ilustração 37 – ARCMS Gestão de Objectos Multimédia

Estrutura de Informação do ARCMS

A base de dados da aplicação está implementada num servidor MySQL com as tabelas no formato InnoDB para permitir relações entre campos.

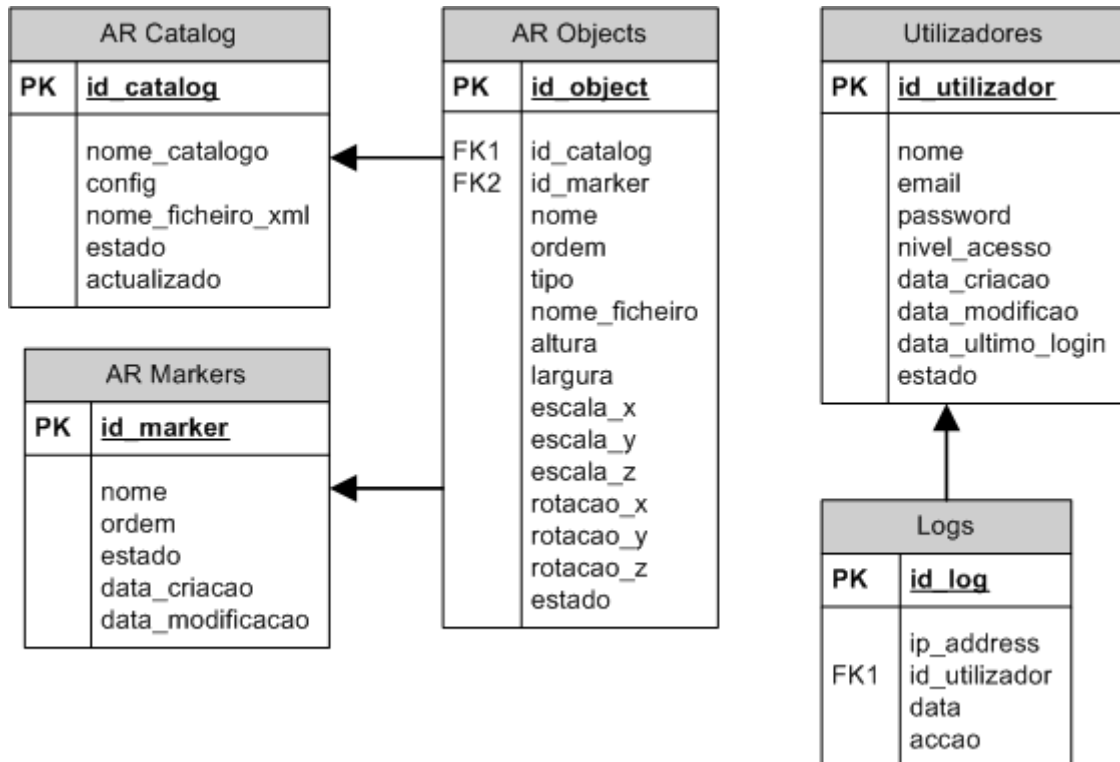


Ilustração 38 – ARCMS Base de Dados

Tabela AR Catalog

A tabela AR Catalog guarda a informação dos catálogos. A cada registo são associados dois ficheiros XML, o ficheiro de configuração do FLAR Manager (que também contém os caminhos para os padrões do objectos) e o ficheiro com as propriedades dos objectos multimédia, ambos são gerados quando se actualiza o catálogo e são criados na estrutura de directórios usando o campo “nome_ficheiro_xml” como nome do ficheiro.

O campo “config” da tabela guarda a informação serializada das configurações do catálogo. Esta solução possibilita uma maior longevidade do modelo de base de dados, uma vez que se constatou que podem existir alterações nas propriedades dos ficheiros de configuração em novas versões do FLAR Manager.

Por exemplo, um catálogo com o nome 'myband' gera os seguintes ficheiro na estrutura de directórios da aplicação:

- Ficheiro configuração: “_resources/flar/myband.xml”

```
<flar_config>
  <flarSourceSettings
    sourceWidth="640"
    sourceHeight="480"
    displayWidth="640"
    displayHeight="480"
    framerate="30"
    trackerToSourceRatio="0.5" />
  <flarManagerSettings mirrorDisplay="true" smoothing="4" >
    <smoother className="FLARMatrixSmoother_Average" positionToRotationRatio="0.5" />
    <thresholdAdapter className="DrunkHistogramThresholdAdapter" speed="0.3" />
  </flarManagerSettings>
  <trackerSettings>
    <flarToolkitSettings cameraParamsFile="_resources/flar_toolkit/FLARCameraParams.dat" >
      <patterns resolution="16"
        patternToBorderRatioX="50"
        patternToBorderRatioY="50"
        minConfidence="0.85" >
        <pattern path="_resources/patterns/1.pat" />
        <pattern path="_resources/patterns/6.pat" />
      </patterns>
    </flarToolkitSettings>
  </trackerSettings>
</flar_config>
```

Excerto de código 3 – XML de configuração de um catálogo

- Ficheiro de Objectos: “_resources/media/xml/myband.xml”

```
<?xml version="1.0" encoding="utf-8"?>
<catalog>
  <media>
    <id>0</id>
    <type>collada</type>
    <source>_resources/media/objects/6/cow.dae</source>
    <width>0</width>
    <height>0</height>
    <scale_x>50.00000000</scale_x>
    <scale_y>50.00000000</scale_y>
    <scale_z>50.00000000</scale_z>
    <rotation_x>270</rotation_x>
    <rotation_y>180</rotation_y>
    <rotation_z>-90</rotation_z>
  </media>
  <media>
    <id>1</id>
    <type>image</type>
    <source>_resources/media/objects/4.jpg</source>
    <width>300</width>
    <height>300</height>
    <scale_x>1.00000000</scale_x>
    <scale_y>-1</scale_y>
    <scale_z>1.00000000</scale_z>
    <rotation_x>0</rotation_x>
    <rotation_y>0</rotation_y>
    <rotation_z>-90</rotation_z>
  </media>
</catalog>
```

Excerto de código 4 – XML dos objectos multimédia de um catálogo

Tabela AR Markers

A tabela AR Markers guarda a informação sobre os marcadores do Magic Catalog, onde a cada registo é associado o ficheiro da imagem do ARM e o ficheiro com o padrão da imagem.

Os ficheiros associados a cada registo são guardados no sistema de ficheiros utilizando a seguinte regra:

- Imagem ARM: “_resources/markers_images/<id_marker>.png”
- Padrão da imagem: “_resources/patterns/<id_marker>.pat”

Tabela AR Objects

A tabela AR Objects guarda a informação sobre os objectos multimédia e os marcadores associados a cada objecto. Por convenção, nos objectos do tipo COLLADA, os campos altura e largura são colocados a Zero porque não são usados quando carregados no Magic Catalog Viewer.

Os ficheiros dos objectos são guardados utilizando a seguinte regra:

- Ficheiros de Imagem, Vídeo e Flash (SWF):
“_resources/media/objects/<id_object>.extensao_ficheiro”
- Os ficheiros do modelo COLLADA:
“_resources/media/objects/<id_object>/nome_original.extensao_ficheiro”

Tabela Utilizadores

A tabela Utilizadores guarda as informações dos utilizadores, assim como o seu nível de acesso dentro da aplicação.

Tabela Logs

A tabela de Logs guarda todos os registos das acções realizadas pelos utilizadores no ARCMS.

4.2.5 MC Media Viewer

O MC Media Viewer (ver Ilustração 39, Ilustração 40 e Ilustração 41) faz parte do ARCMS e tem como papel a visualização dos objectos multimédia simulando a sua posição num ARM sem a necessidade de recorrer ao processo de detecção pela câmara, o que permite que os utilizadores realizem directamente as alterações num ambiente controlado. Para ajudar no posicionamento dos objectos multimédia, a simulação do ARM tem os eixos X, Y e Z desenhados (ver Ilustração 40).

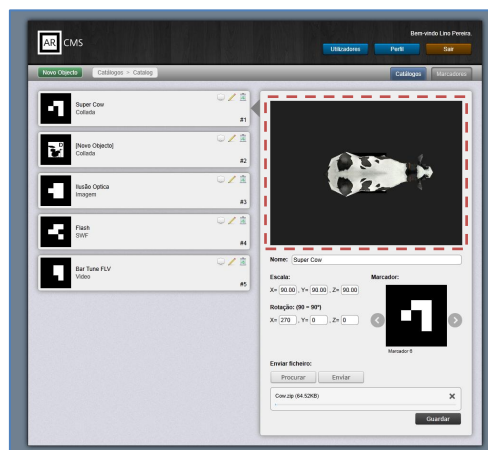


Ilustração 39 – MC Media Viewer na edição de um objecto

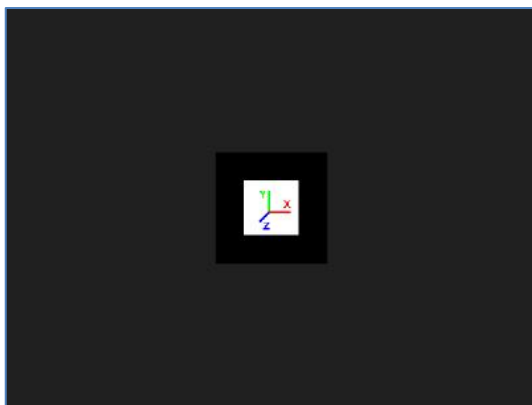


Ilustração 40 – MC Media Viewer sem objecto

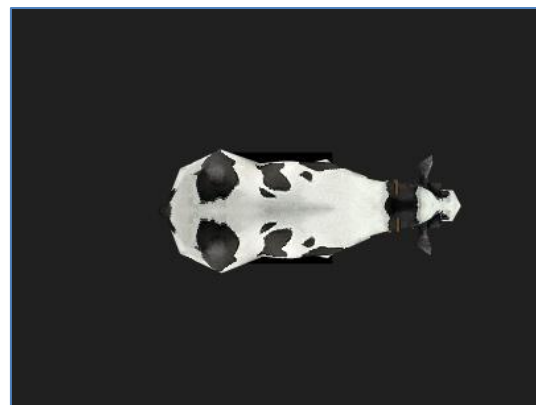


Ilustração 41 – MC Media Viewer com objecto 3D

O MC Media Viewer é um componente SWF embebido no HTML do BackOffice. Os parâmetros para carregar um objecto multimédia no MC Media Viewer são enviados por javascript pelas FlashVars do SWF (ver Excerto de código 5).

```

function get_mc_media_viewer(media_type, media_url, width, height, sx, sy ,sz, rx, ry, rz)
{
    var flashvars = {
        "media": media_url,
        "mediatype": media_type,
        "w": width,
        "h": height,
        "sx": sx,
        "sy": sy,
        "sz": sz,
        "rx": rx,
        "ry": ry,
        "rz": rz
    };
    var params = {
        menu: "false",
        scale: "noScale",
        allowFullscreen: "true",
        allowScriptAccess: "always",
        bgcolor: "#202020"
    };
    var attributes = {
        id: "mc_media_view"
    };
    swfobject.embedSWF("mc_media_view.swf", "media_viewer", "400", "300", "10.0.0",
"expressInstall.swf", flashvars, params, attributes);
}

```

Excerto de código 5 – Código javascript para carregar o MC Media Viewer

Na função que carrega o MC Media Viewer é passado por parâmetro o tipo de media que vamos carregar, a localização física do ficheiro, a dimensões do objectos (são calculadas automaticamente no *upload* do ficheiro), a escala e a rotação inicial. Algumas opções são ignoradas dependendo do tipo de objecto em questão. Após o objecto ser carregado, podemos interagir com ele alterando a rotação e escala até ficarem com o aspecto desejado. Estas modificações são realizadas pela interface javascript do MC Media Viewer que comunica com o SWF.

```

function update_scale()
{
    if(document.getElementById("mc_media_view")){
        var x = parseFloat($j('#scale_x').val());
        var y = parseFloat($j('#scale_y').val());
        var z = parseFloat($j('#scale_z').val());

        if(!isNaN(x) && !isNaN(y) && !isNaN(z)){
            document.getElementById("mc_media_view").mcmv_update_scale(x, y, z);
        }
    }
}

function update_rotation()
{
    if(document.getElementById("mc_media_view")){
        var x = parseFloat($j('#rotation_x').val());
        var y = parseFloat($j('#rotation_y').val());
        var z = parseFloat($j('#rotation_z').val());

        if(!isNaN(x) && !isNaN(y) && !isNaN(z)){
            document.getElementById("mc_media_view").mcmv_update_rotation(x, y, z);
        }
    }
}

```

Excerto de código 6 - Funções da Interface Javascript do MC Media Viewer

Como o MC Media Viewer não utiliza o FLAR Manager, embora o motor gráfico seja o mesmo, o sistema de coordenadas é diferente do Magic Catalog. Neste caso o sistema de coordenadas é relativo ao ecrã e não ao ARM detectado. Para corrigir esta situação é aplicada uma correcção às coordenadas quando estamos a gerar o XML do catálogo. Também é importante referir que esta componente apenas mostra uma simulação do que irá aparecer no Magic Catalog, uma vez que, a escala do objecto irá depender da distância física entre o ARM e a *webcam*.

O Excerto de código 7 ilustra o funcionamento da classe principal do MC Media View. Algumas secções indicadas por “/* (...) */” foram retiradas para aumentar a legibilidade do código.

```

package MCMediaView
{
    /* IMPORTS ...**/ (..) */

    /**
     * @author Lino Pereira
     */
    [Frame(factoryClass = "MCMediaView.Preloader")]
    [SWF(width="400", height="300", frameRate="30", backgroundColor="#202020")]
    public class Main extends Sprite
    {
        /* (...) */

        private var planoMarcador:Plane; //Plano que contém a imagem do marcador
        private var planoObjecto:Plane; //Plano para onde são carregados os objectos de media
        private var modelo:DAE; //Modelo para carregar objectos Collada
        private var object_is_model:Boolean = false; //Verifica se o objecto é um modelo ou um
plano
        private var object_is_loaded:Boolean = false; //Variável para verificar se o objecto esta
carregado
        private var media_url:String = "-100"; //Endereço do Objecto de media
        private var media_type:String = "-100"; //tipo de media (collada, swf, video, image)
        /*Correcção aplicada ao modelos para ficarem do "mesmo" tamanho da aplicação final*/
        private var scaleCorrection:Number = 2;
        private var rotationCorrectionX:Number = 0;
        private var rotationCorrectionY:Number = 180;
        private var rotationCorrectionZ:Number = -90;
        /*Escala aplicada e passada como parametro*/
        private var escalaX:Number = 1;
        private var escalaY:Number = 1;
        private var escalaZ:Number = 1;
        /*Rotação aplicada e passada como parametro*/
        private var rotacaoX:Number = 0;
        private var rotacaoY:Number = 0;
        private var rotacaoZ:Number = 0;
        /*Dimensoes aplicadas como parametro*/
        private var dimWidth:Number = 80;
        private var dimHeight:Number = 80;
        private var imgLoader:Loader;
        private var swfLoader:Loader;
        private var video:Vid;

        public function Main():void
        {
            if (stage) init();
            else addEventListener(Event.ADDED_TO_STAGE, init);
        }

        private function init(e:Event = null):void
        {
            removeEventListener(Event.ADDED_TO_STAGE, init);

            setupPV3D();
        }
    }
}

```

```

if (is_browser())
{
    this.media_url = getFlashVars().media || "-100";
    this.media_type = getFlashVars().mediatype || "-100";

    this.dimWidth = getFlashVars().w || 80;
    this.dimHeight = getFlashVars().h || 80;

    this.escalaX = getFlashVars().sx || 1;
    this.escalaY = getFlashVars().sy || 1;
    this.escalaZ = getFlashVars().sz || 1;

    this.rotacaoX = getFlashVars().rx || 0;
    this.rotacaoY = getFlashVars().ry || 0;
    this.rotacaoZ = getFlashVars().rz || 0;

    //Chamadas externas de javascript
    ExternalInterface.addCallback("mcmv_update_scale", update_object_scale);
    ExternalInterface.addCallback("mcmv_update_rotation", update_object_rotation);
}

addObject();
addEventListener(Event.ENTER_FRAME, update_objects);
}

private function setupPV3D():void
{
    /* (...) */
}

private function addObject():void
{
    ////////////////////////////////////////////////////
    //Similar Marcador
    var loader:Loader = new Loader();
    loader.load(new URLRequest("arcms/templates/images/marker.png"));
    var movMat:MovieMaterial = new MovieMaterial(loader, false, true);
    this.planoMarcador = new Plane(movMat);
    this.planoMarcador.z = 50;
    this.scene.addChild(planoMarcador);
    ////////////////////////////////////////////////////

    switch(media_type)
    {
        case "video":

            video = new Vid();
            video.vid.source = String(this.media_url);

            var movMat1:MovieMaterial = new MovieMaterial(video, false, true, false, new
Rectangle(0,0,this.dimWidth,this.dimHeight));
            this.planoObjecto = new Plane(movMat1, this.dimWidth, this.dimHeight, 1, 1);

            this.planoObjecto.scaleX = this.escalaX;
            this.planoObjecto.scaleY = this.escalaY;
            this.planoObjecto.scaleZ = this.escalaZ;
            this.planoObjecto.rotationX = this.rotacaoX;
            this.planoObjecto.rotationY = this.rotacaoY;
            this.planoObjecto.rotationZ = this.rotacaoZ;

            this.scene.addChild(this.planoObjecto);

            this.object_is_loaded = true;
            this.object_is_model = false;
            break;

        case "image": /* (...) */ break;
        case "swf":/* (...) */ break;

        case "collada":

            this.modelo = new DAE(true, "model", true);
            this.modelo.load(this.media_url);

            /* (...) */

```

```

        this.scene.addChild(modelo);
        this.object_is_loaded = true;
        this.object_is_model = true;
        break;
    }
}

private function update_objects(e:Event):void
{
    if (this.object_is_loaded){
        if (this.object_is_model){/* (...) */}else{/* (...) */}
    }
    this.renderEngine.render();
}

private function update_object_scale(scale_x:Number, scale_y:Number, scale_z:Number):void{
    /* (...) */
}

private function update_object_rotation(rotation_x:Number, rotation_y:Number,
rotation_z:Number):void
{
    /* (...) */
}

private function is_browser(): Boolean
{
    if (Capabilities.playerType == "ActiveX" || Capabilities.playerType == "PlugIn"){
        return true;
    }
    return false;
}

private function getFlashVars():Object {
    return Object( LoaderInfo( this.loaderInfo ).parameters );
}
}
}

```

Excerto de código 7 – Código AS3 do MC Media Viewer

4.3 Análise e apresentação de resultados

Neste capítulo é descrito o método de avaliação do trabalho desenvolvido, fazendo uma análise crítica às limitações das tecnologias escolhidas, às ferramentas de desenvolvimento utilizadas e à aplicação como resultado final deste projecto, com a perspectiva de explicar os problemas encontrados e como foram ultrapassados.

A avaliação dos protótipos em conjunto com os sete colaboradores e dois clientes da Front-End, permitiu delinear quais as tecnologias a serem usadas no desenvolvimento deste projecto. A prototipagem do componente principal da aplicação, o Magic Catalog Viewer, em diferentes tecnologias, serviu para identificar os problemas e as dificuldades quando manipulado pelos utilizadores finais. Alguns dos problemas encontrados na fase de prototipagem ajudaram a resolver situações que surgiram durante o desenvolvimento da aplicação.

4.3.1 Testes de hardware

Os testes de hardware foram elaborados com o objectivo de identificar qual a combinação de hardware que permitia uma melhor performance da aplicação. Algumas das aplicações desenvolvidas são utilizadas em ambiente controlado, como por exemplo em exposições, lojas, apresentações e workshops, nestas situações interessa maximizar o potencial da aplicação.

Para minimizar o erro da análise devido a factores externos, como iluminação ambiente (que influencia a qualidade da imagem na webcam) e ao *software* a executar paralelamente, os testes foram sempre realizados no mesmo intervalo horário (10h00 ao 12H00) e com os serviços mínimos a executar no computador.

A primeira conclusão tirada, como previsto, foi que quanto maior era a capacidade de processamento e de memória do computador melhor era o desempenho (fluidez de movimentos e o número de frames por segundo da *webcam*).

Após este resultado, os testes foram orientados para descobrir qual a *webcam* que permitia uma melhor qualidade de imagem e performance da aplicação.

Características das *webcams* utilizadas nos testes:

- Creative WebCam Live! ULTRA USB 2.0
- LOGITECH - Pro 9000 Webcam for Business (USB 2.0)
- PlayStation Eye (USB 2.0)

Nos testes realizados não existia uma diferença notável na qualidade de imagem entre os diferentes modelos de *webcams* utilizando a resolução 640x480.

A suavidade na detecção do ARM aumentava com o aumento do contraste das cores da imagem da *webcam*, uma vez que introduz menos ruído na imagem quando esta é convertida para preto e branco no algoritmo de detecção.

Em comparação com as restantes *webcams*, a PlayStation Eye tem um maior ângulo de visão e manteve a captura das imagens nos 30fps sem grandes oscilações na *frame rate*. A ideia de usar a PlayStation Eye surgiu no levantamento do estado da arte ao analisar o PlayStation Move, a câmara tem resultados comprovados e um preço/qualidade acessível.

4.3.2 Testes de software

Os testes à aplicação ocorrem em duas fases, na primeira fase foi avaliado o Magic Catalog Viewer e na segunda fase o ARCMS, ambos foram testados por utilizadores e programadores.

O Magic Catalog Viewer foi testado nos *browsers* mais utilizados⁸: Firefox 3.0+, Google Chrome 4.0+, Safari 3.1+, Internet Explorer 7.0+ e Opera 10.5+ e em diferentes sistemas operativos, Window XP, Window Vista, Window 7, Mac e Ubuntu Linux. Desde que o Flash Player 10 tivesse instalado a aplicação funcionava correctamente nos diferentes ambientes.

Os testes de carga ao Magic Catalog Viewer foram realizados com o Firefox 5.0 em Windows 7, com a finalidade de avaliar o desempenho da aplicação (rapidez da detecção, suavidade na movimentação dos objectos pelo ecrã, etc.) utilizando múltiplos objectos multimédia em comparação com um só objecto multimédia no mesmo ambiente.

Foram realizados dois testes diferentes e utilizou-se a seguinte escala na comparação:

- 10 - Para um desempenho igual quando comparado com um só ARM.
- 5 - Piscando uma a duas vezes durante a movimentação no ecrã e detectar o ARM em menos de dois segundos.
- 1 - Para um desempenho muito pior quando comparado com um só ARM.

Tabela 2 – Resumo dos resultados dos testes ao Magic Catalog Viewer

N.º de ARMs	Teste	
	Apresentar os ARMs um a um	ARMs em simultâneo no ecrã
1	<i>Valor de referência</i>	<i>Valor de referência</i>
2	10	10
3	10	10
4	10	9.9
8	10	8.6
16	9.9	7.9
24	9.7	6.5
32	8.8	5.4
64	6.8	3.1

⁸ Browser Statistics - http://www.w3schools.com/browsers/browsers_stats.asp

Tendo em consideração os resultados obtidos (ver Tabela 2) e em conjunto com o *feedback* dos entrevistados chegou-se às seguintes conclusões:

- Até 8 ARMs a aplicação tem um óptimo desempenho
- Entre 8 e 16 ARMs a aplicação tem bom desempenho
- Entre 16 e 32 ARMs a aplicação tem um desempenho razoável
- Com os 32 ARMs a aplicação demorava mais tempo a carregar os objectos
- Com 32 ARMs foi difícil colocá-los num ângulo visível para a câmara (foram impressos 6 ARMs por folha A4 e coladas entre si)
- Com 64 ARM's tornou-se difícil dispô-los em simultâneo dada a oscilação inerente às folhas.
- Mostrar os ARMs sequencialmente tem um melhor desempenho do que em simultâneo. Tecnicamente pode ser explicado por duas razões: quando se mostra os ARMs sequencialmente, o algoritmo de carregamento de objectos multimédia vai pré-carregar os objectos adjacentes, uma vez detectado o ARM só se tem que mostrar o objecto. Paralelamente, quando são utilizados vários ARMs em simultâneo, o próprio método de detecção é mais demorado devido ao número de ARMs e aos objectos multimédia vão ser todos carregados ao mesmo tempo.

Segundo o levantamento dos projectos desenvolvidos pela Front-End que utilizam este tipo de ARMs com realidade aumentada, 90% dos casos usam um ou dois ARMs e um máximo de seis ARMs em simultâneo. Tendo em conta este cenário, a aplicação vai de encontro com as necessidades da empresa.

O material em que o ARM é impresso pode ajudar na detecção, razão pela qual devemos considerar os seguintes aspectos:

- Papel muito brilhante reflecte os pontos de luz no papel, o que implica que podem aparecer círculos ou faixas brancas quando a câmara captura a imagem e analisa no algoritmo de detecção, que impedem o padrão de ser reconhecido.
- O ARM deve ser impresso num material que não dobre, se o papel dobrar disforma a imagem que é capturada pela câmara e o padrão não é reconhecido.

Após alguma investigação e na tentativa de encontrar o melhor material, sempre que possível devemos imprimir os ARMs em papel e tinta fosca com uma gramagem igual ou superior a 150 g/m². Para casos especiais, em que existe muitos pontos de luz na sala, o melhor material encontrado foi uma cartolina de 180 g/m² de superfície rugosa com o padrão recortado em “papel” autocolante que na superfície tem uma espécie veludo preto. As

características desta superfície impediam que a luz fosse reflectida minimizando o aparecimento de “artefactos” na imagem.

Todos os testes e experiencias realizadas serviram para apurar o âmbito, as limitações e as melhorias a futuras implementações do Magic Catalog Viewer.

Na fase de testes do ARCMS os programadores e utilizadores efectuaram testes com diferentes objectivos, enquanto os utilizadores testaram a funcionalidades e a usabilidade do BackOffice, os programadores testaram a capacidade de integração com outras aplicações e as funcionalidades na óptica do desenvolvimento.

Realizou-se um inquérito aos utilizadores em que cada tópico era avaliado de 1 a 5, sendo 5 melhor nota.

Tabela 3 – Resumo do Inquérito de Qualidade do ARCMS

Tópico	Nota Média
Como classifica o aspecto de cada módulo da aplicação	4.5
A iconologia dos botões corresponde as acções realizadas	4.8
Como classifica o aspecto geral da aplicação	4.8
A aplicação é intuitiva de utilizar	4.5
A aplicação explica convenientemente os erros encontrados	4
Como classifica a visualização da informação de cada módulo	5
Como classifica a rapidez do processo de aprendizagem na utilização da aplicação	4.5
Como classifica o funcionamento geral	4.5

Dos resultados da Tabela 3 não se consegue avaliar os melhoramentos devido à proximidade de resultados. No entanto reduzindo o intervalo da avaliação de 4 a 5, identificamos que pode existir uma forma de melhorar o *feedback* que se dá aos utilizadores em caso de erro na aplicação. Este inquérito será novamente realizado quando os utilizadores estiverem mais familiarizados com a aplicação e consigam apontar erros por repetição de operações.

Foi pedido a três programadores da empresa para instalar o Magic Catalog no servidor e criar dois catálogos com as respectivas páginas HTML, a instalação foi bastante linear, uma vez que só é necessário copiar os ficheiros do projecto e criar a base de dados através do ficheiro MySQL. Uma vez instalados, os catálogos foram criados com objectos multimédia, gerou-se o HTML para cada um deles e enviou-se o ficheiro para a raiz do projecto no servidor. Com o Magic Catalog instalado e a funcionar, foi lançado o desafio de usar a API Javascript para manipular outros objectos HTML. Devido à subjectividade destes testes, em lugar de um inquérito, realizou-se uma reunião com os programadores para discutir a avaliação da aplicação e apresentar as novas funcionalidades pensadas durante o desenvolvimento. Deste encontro surgiram novas ideias e melhorias que estão descritas no capítulo 5.1 Desenvolvimento Futuro.

4.3.3 Demonstrações de utilização

Este subcapítulo descreve algumas das implementações realizadas com Magic Catalog onde são apresentados os aspectos técnicos e funcionalidades de cada demonstração.

Demo Magic Catalog

A Demo Magic Catalog é a apresentação de todas as funcionalidades do Magic Catalog, nesta demonstração são apresentados os tipos de objectos multimédia suportados e como se podem gerir no ARMCS. O acesso ao BackOffice ARCMS para gestão dos catálogos é facultado consoante o tipo de projecto.

Na Ilustração 42 estão presentes os seguintes objectos:

1. Animação Flash (SWF)
2. Objecto 3D COLLADA
3. Vídeo FLV
4. Imagem JPG

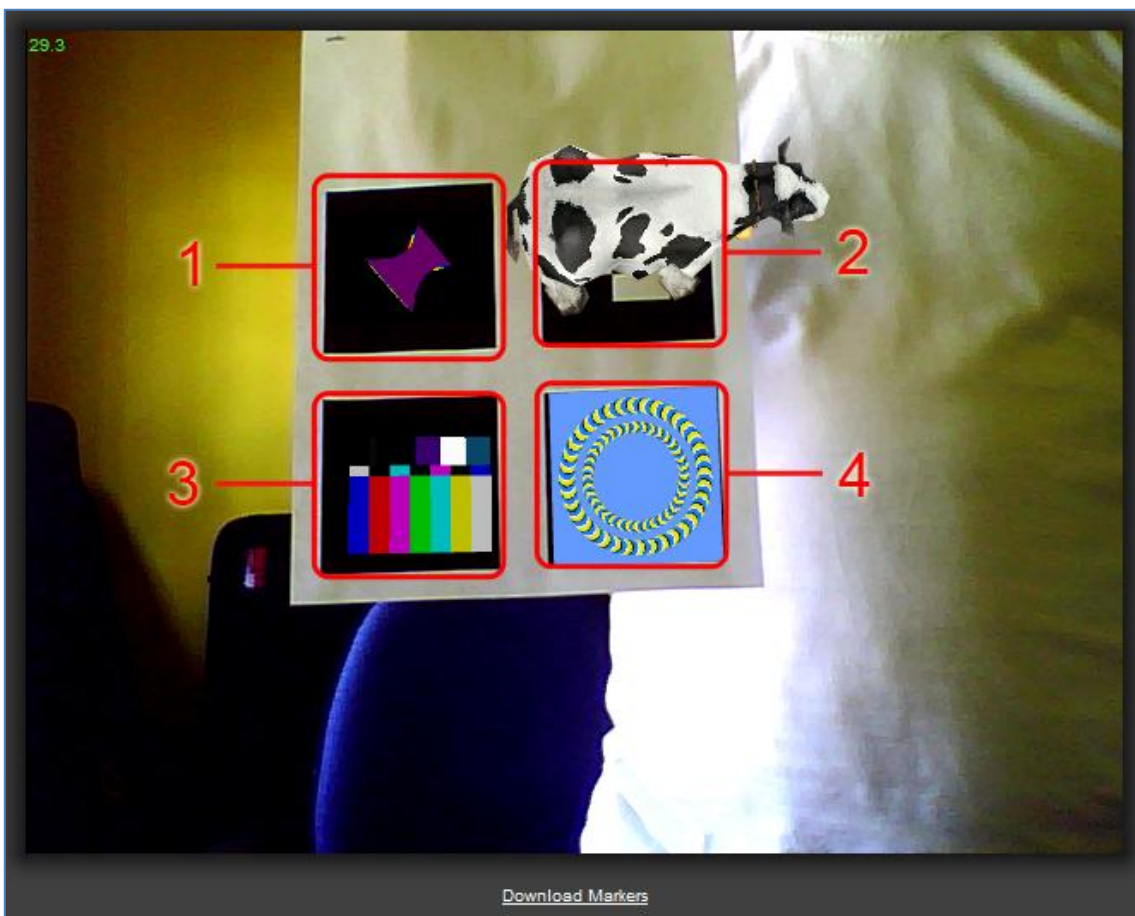


Ilustração 42 – Demo Magic Catalog

My Band

A “My Band” (à esquerda na Ilustração 43) é uma aplicação Flash que está integrada com o Magic Catalog (à direita na Ilustração 43) pela API Javascript. Foi desenvolvida no âmbito deste projecto de tese para demonstrar as possibilidades de implementação dos componentes desenvolvidos.

A “My Band” é composta por uma música dividida em 3 faixas: Guitarra+Voz, Baixo e Bateria, assim que detecta os três instrumentos (ARMs) a música começa tocar, se o instrumento não for detectado, é desactivado e a faixa correspondente pára de tocar.

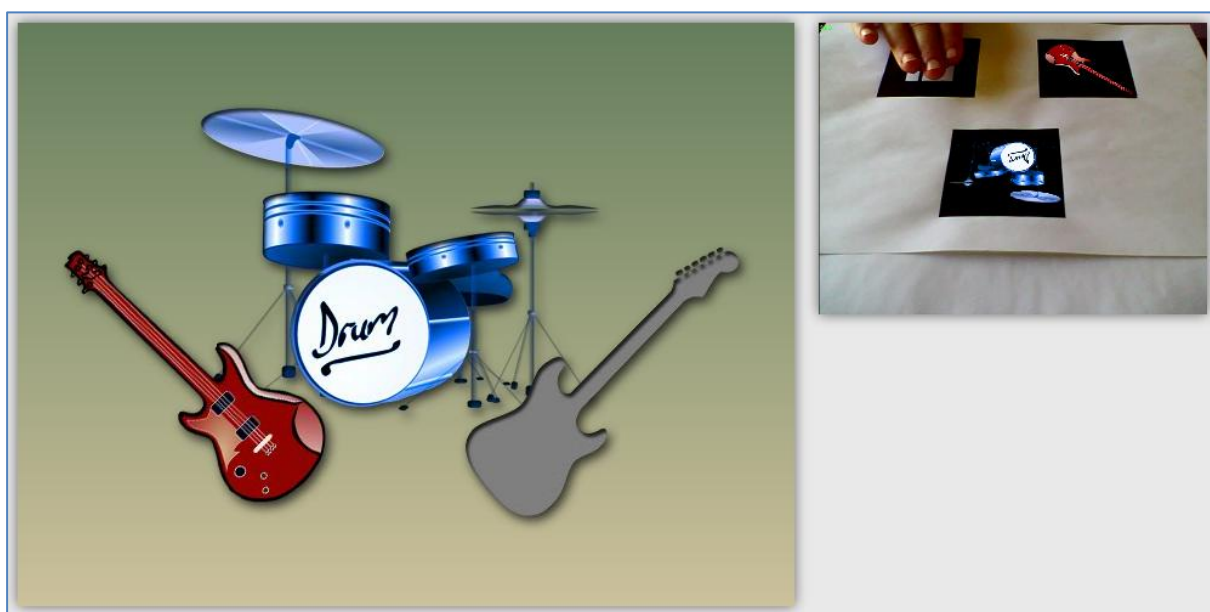


Ilustração 43 – My Band

Dazbook Catalog

O Dazbook Catalog é uma aplicação desenvolvida para um cliente da Front-end que também esteve envolvido na fase de testes do Magic Catalog.

Neste projecto o Magic Catalog é integrado com o Dazbook (*flippage* digital desenvolvido pela Front-end), no Dazbook é apresentado o catálogo de produtos da empresa e no Magic Catalog o vídeo de apresentação dos produtos de cada tema do catálogo. Sempre que o Magic Catalog detecta um novo ARM, o Dazbook é desfolhado até a página do tema correspondente.

A empresa cliente, optou por incluir nos catálogos que distribui em formato papel postais com a impressão do ARM e, uma pequena explicação de como podem ser utilizados. Esta aplicação será disponível ao público em Dezembro de 2011

O novo catálogo em papel do cliente vai incluir os ARMs nas respectivas páginas, o que não aconteceu no actual, dado já ter sido publicado aquando do início do desenvolvimento deste projecto.

Nesta implementação do Magic Catalog a empresa tem a liberdade para gerir os conteúdos da aplicação, o que lhe possibilita a modificação e destacamento dos produtos por tema.

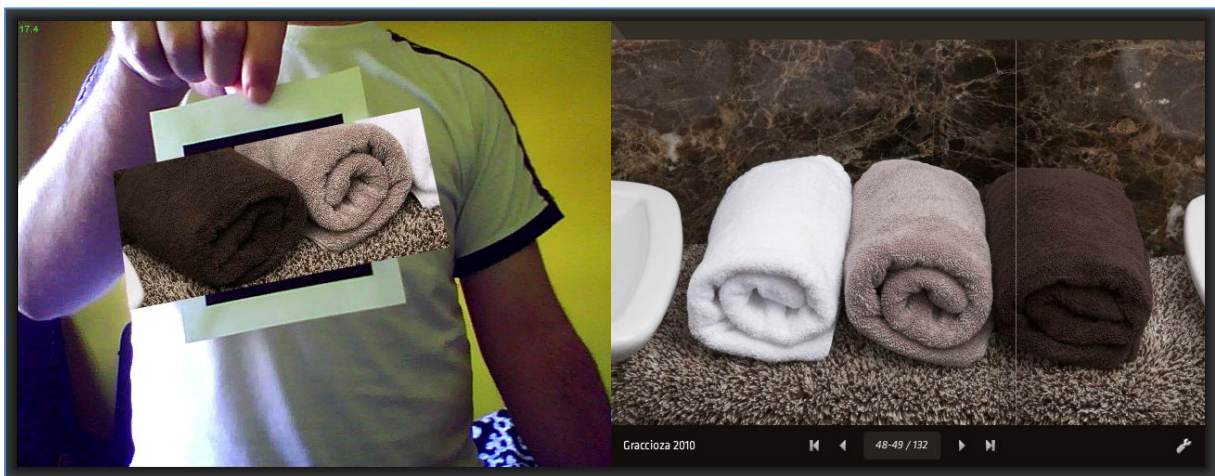


Ilustração 44 – Dazbook Catalog

4.3.4 Limitações das tecnologias utilizadas

O meio envolvente é a principal limitação à tecnologia de detecção de padrões do Magic Catalog. Como a detecção é realizada com imagens a preto e branco, o lugar deve ser bem iluminado para permitir um bom contraste entre o fundo e o ARM. Quanto mais claro for o lugar mais rápida é a detecção do ARM, uma vez este ignora as grandes áreas só com branco e pesquisa nas áreas com pixels pretos. Também existe uma distância limite para a detecção do ARM, dependendo das condições, os ARMs de 8 centímetros conseguem ser detectados até um distância de 1 a 2 metros da câmara. ARMs simples aumentam a distância máxima com que conseguem ser reconhecidos. A escolha dos ARMs e do espaço envolvente é determinante para o bom funcionamento da aplicação.

Na publicação online da aplicação é necessário ponderar a quantidade de ARMs em simultâneo e a complexidade dos objectos multimédia que são utilizados, uma vez que, se

excluirmos os factores externos, o desempenho da aplicação depende das características do computador do utilizador.

Neste momento, a maior barreira tecnológica do Magic Catalog está na API Javascript de integração com outras aplicações, o facto de os *browsers* não conseguirem processar todas as chamadas efectuadas pelo Magic Catalog Viewer, implica que não é possível enviar informação sobre as actualizações efectuadas ao ARM, como por exemplo enviar as coordenadas espaciais, a rotação e as coordenadas de projecção X e Y relativas a área de visualização do Magic Catalog Viewer.

Considerando que as limitações impostas pela tecnologia de detecção de padrões e o meio envolvente onde a aplicação é usada, são mais complexas de resolver, o desenvolvimento futuro irá inferir sobre as funcionalidades desenvolvidas durante este projecto.

5 Conclusões

Dada a crescente utilização das aplicações de realidade aumentada no quotidiano da sociedade actual, torna-se cada vez mais vulgar o uso de conteúdos virtuais que possibilitam a extensão e propagação da interacção humana com o meio envolvente.

Neste sentido, torna-se fulcral produzir tecnologias capazes de interpretar e analisar as diferentes vertentes do meio envolvente, assim como garantir a criação de ferramentas que possibilitem a eficaz gestão dos conteúdos virtuais resultantes da interacção apresentada.

O objectivo principal do presente trabalho é desenvolver e demonstrar a possibilidade de utilização de uma única ferramenta a uma panóplia de projectos de âmbitos diversos. Esta aplicação apenas será potenciada se for garantido o mínimo de intervenção técnica e paralelamente seja transferido para os clientes o ónus da gestão dos conteúdos multimédia, para que ganhos de eficiência na alocação de recursos e produção de conteúdos sejam atingidos pelas empresas que desenvolvem este tipo de aplicações.

A investigação desenvolvida permitiu identificar um leque de possíveis soluções para a resolução dos problemas supra expostos. Uma das soluções propostas passa pela implementação de protótipos que satisfazem os requisitos e que incorporam o domínio técnico da empresa de desenvolvimento. Adicionalmente, a avaliação por parte dos utilizadores e programadores dos protótipos desenvolvidos, facilitou a escolha do protótipo que serviu de base tecnológica para o desenvolvimento da aplicação final. Esta apreciação permitiu identificar antecipadamente problemas num ambiente controlado que poderiam atrasar o desenvolvimento da aplicação, caso fossem encontrados na fase de implementação.

O produto final cumpre todos os objectivos traçados inicialmente, possibilita estender as funcionalidades e integrar com outras aplicações, adicionalmente permite aos clientes a criação de novas aplicações com conteúdos próprios, sem a necessidade de programar uma única linha de código.

Esta versatilidade e escalabilidade do produto fomentam o desenvolvimento de novas funcionalidades e ferramentas. Subsequentemente, a generalização dos recursos e técnicas de realidade aumentada são disponibilizados a alto nível, o que possibilita que a aplicação seja utilizada como base de desenvolvimento por outras empresas do ramo da programação ou na produção de conteúdos multimédia.

A participação dos clientes e colaboradores da Front-End, durante a fase de teste e de implementação do produto, foram cruciais para o desenvolvimento de um projecto que

estabelece a ponte entre as tecnologias utilizadas na criação de aplicações com realidade aumentada e a gestão dos seus conteúdos.

5.1 Desenvolvimento Futuro

Face aos resultados obtidos com a primeira versão do Magic Catalog e após reflexão sobre a avaliação realizada à aplicação, estão previstos os seguintes desenvolvimentos e melhorias:

- Possibilidade de modificar, durante a execução, a posição, escala e rotação dos objectos no Magic Catalog Viewer através da API Javascript.
- Na perspectiva de tornar o Magic Catalog uma ferramenta mais transversal para vender a outras empresas da mesma área, será acrescentado o suporte para os motores gráficos: Alternativa3D, Away3D e Sandy 3D.
- Colocar objectos multimédia no Magic Catalog que não dependam do sistema de detecção de padrões e que fiquem sempre presente no ecrã.
- Notificação de colisões entre os objectos presentes no Magic Catalog Viewer. Detecção de dois tipos de colisão: colisões no espaço tridimensional e colisões no plano do “ecrã” (ignorando a profundidade a que os objectos estão do ecrã).
- Na gestão de marcadores no BackOffice, gerar um ficheiro com o padrão na submissão da imagem, extinguindo a necessidade de recorrer a aplicações externas ao ARCMS para realizar este procedimento.

6 Bibliografia

- Augmented Reality in Education*. (12 de Novembro de 2011). Obtido de Augmented Reality in Education: http://wik.ed.uiuc.edu/index.php/Augmented_Reality_in_Education
- PS Move Community*. (7 de Novembro de 2011). Obtido de PlayStation Move: <http://us.playstation.com/ps3/playstation-move/move-community/>
- What is FLARToolKit*. (12 de Novembro de 2011). Obtido de Spark Project: <http://www.libspark.org/wiki/saqoosha/FLARToolKit/en>
- ABS-CBN Regional Network Group. (8 de Novembro de 2011). *TV Patrol Southern Mindanao*. Obtido de TV Patrol Southern Mindanao: http://en.wikipedia.org/wiki/TV_Patrol_Southern_Mindanao
- Azuma, R. T. (29 de Novembro de 2011). A Survey of Augmented Reality. pp. 4-5.
- Bergig, O., Hagbi, N., El-Sana, J., & Billinghurst, M. (2009). In-Place 3D Sketching for Authoring and Augmenting. *8th IEEE International Symposium on Mixed and Augmented Reality (ISMAR 2009)*, pp. 87-94.
- Billinghurst, M., Kato, H., & Poupyrev, I. (2011). The MagicBook - Moving Seamlessly between Reality and Virtuality. *Computer Graphics and Applications*, pp. 2-4.
- Coca-Cola. (8 de Novembro de 2011). *Coca-Cola*. Obtido de Coca-Cola: <http://www.coca-cola.com.sg/cokezeroavatar/index.html>
- Cooper, M. (23 de Novembro de 2011). *Augmented Reality*. Obtido de Lectures: <http://webstaff.itn.liu.se/~matco/TNM053/Lectures/Lecture%2009%20-%20Augmented%20Reality.pdf>
- Drascic, D., Grodsk, J., Milgram, P., Ruffo, K., Wong, P., & Zhai, S. (s.d.). ARGOS: A Display System for Augmenting Reality. *ACM SIGGRAPH Tech Video Review, Vol 88*, p. 521.
- Edwards, E., Rolland, J., & Keller, K. (s.d.). Video see-through design for merging of real and virtual environments. *Proc. IEEE Virtual Reality International*, pp. 223-233.
- Fakultät für Informatik. (12 de Novembro de 2011). *A Look into the Body*. Obtido de Fakultät für Informatik: <http://www.in.tum.de/en/research/research-highlights/augmented-reality-in-medicine.html>

- Georgia Tech. (8 de Novembro de 2011). *ARGON*. Obtido de ARGON:
<http://www.argon.gatech.edu/>
- Henderson, S., & Feiner, S. (8 de Novembro de 2011). *Augmented Reality for Maintenance and Repair*. Obtido de ARMAR:
<http://graphics.cs.columbia.edu/projects/armar/index.htm>
- Kammann, T. D. (12 de Novembro de 2011). *Football Matches beyond the screen – Surround AR TV*. Obtido de Augmented.org:
<http://www.augmented.org/blog/2010/04/football-matches-beyond-the-screen-surround-ar-tv/>
- Kato, H., & Billingham, M. (7 de Novembro de 2011). *ARToolKit Documentation*. Obtido de ARToolKit: <http://www.hitl.washington.edu/artoolkit/documentation/vision.htm>
- Lee, J. C. (7 de Novembro de 2011). *Johnny Chung Lee Wii Projects*. Obtido de Johnny Chung Lee: <http://johnnylee.net/projects/wii/>
- Lion, D., Rosenberg, C., & Barfield, W. (1993). Overlaying three-dimensional computer graphics with stereoscopic live motion video. *SID Conf. Proceedings*.
- Metaio. (8 de Novembro de 2011). *adidas Originals 3D Neighborhood*. Obtido de Metaio:
<http://www.metaio.com/projects/web/adidas-originals-3d-neighborhood/>
- Metaio. (8 de Novembro de 2011). *LEGO DIGITAL BOX Augmented Reality Kiosk*. Obtido de Metaio: <http://www.metaio.com/projects/kiosk/lego/>
- Microsoft. (7 de Novembro de 2011). *Kinect for Windows*. Obtido de Kinect:
<http://kinectforwindows.org/features/>
- Milgram, P., Takemura, H., Utsumi, A., & Kishino, F. (23 de Novembro de 2011). *AUGMENTED REALITY: A CLASS OF DISPLAYS ON THE REALITY-VIRTUALITY CONTINUUM*. Obtido de Milgram et al: AR - A Class of Displays:
http://etclab.mie.utoronto.ca/people/paul_dir/SPIE94/SPIE94.full.html
- MIT. (8 de Novembro de 2011). *Football Matches beyond the screen – Surround AR TV*. Obtido de the augmented reality blog:
<http://www.augmented.org/blog/2010/04/football-matches-beyond-the-screen-surround-ar-tv/>
- R., T., C., K., & R., S. (2006). Livro do Pré-Simpósio. Fundamentos e Tecnologia de Realidade Virtual e Aumentada. *VIII Symposium on Virtual Reality*.

- Ray Ban. (8 de Novembro de 2011). *Ray Ban*. Obtido de Ray Ban Virtual Mirror:
<http://www.ray-ban.com/usa/science/virtual-mirror>
- Revista Sábado. (8 de Novembro de 2011). *Revista Sábado*. Obtido de Revista Sábado:
[http://www.sabado.pt/Multimedia/FOTOS/-span--b-Sociedade-b---span--\(1\)/Realidade-aumentada-promocao-revista.aspx?id=218530](http://www.sabado.pt/Multimedia/FOTOS/-span--b-Sociedade-b---span--(1)/Realidade-aumentada-promocao-revista.aspx?id=218530)
- Socolofsky, E. (12 de Novembro de 2011). *FLARManager: Augmented Reality in Flash*. Obtido de transmote speaks...: <http://words.transmote.com/wp/flarmanager/>
- Technische Universität München . (8 de Novembro de 2011). *Research in Medical Augmented Reality* . Obtido de Chair for Computer Aided Medical Procedures & Augmented Reality: <http://campar.in.tum.de/Chair/ResearchIssueMedAR>
- TISSOT. (8 de Novembro de 2011). *Tissot Reality*. Obtido de Tissot:
<http://www.tissot.ch/reality/>
- Total Immersion. (8 de Novembro de 2011). *Total Immersion*. Obtido de Total Immersion:
<http://blog.t-immersion.com/2010/06/29/new-ar-books-dinosaurs-alive-and-fairyland-magic-come-alive/>
- Wagner, D. (12 de Novembro de 2011). *History of Mobile Augmented Reality*. Obtido de History of Mobile Augmented Reality:
<https://www.icg.tugraz.at/~daniel/HistoryOfMobileAR/>
- Wikipedia. (7 de Novembro de 2011). *Augmented reality*. Obtido de Wikipedia:
http://en.wikipedia.org/wiki/Augmented_reality