

PROJETO WEBM, UMA ALTERNATIVA PARA O VIDEO NA WEB

Marcelo Félix, N° 59075

Instituto Superior Técnico
Av. Rovisco Pais, 1049-001 Lisboa, Portugal
E-mail: marcelo.felix@ist.utl.pt

RESUMO

Neste artigo descreve-se o projecto WebM. Introduce-se brevemente os eventos que permitiram a sua implementação. Descreve-se a arquitectura dos seus *codecs*, nomeadamente o VP8 para o vídeo e o Vorbis para o áudio, baseando-se na estrutura de agregação e sincronização da Matroska.

É feita uma análise à sua licença, a sua influência no mercado, os apoios que provocou e como irão ajudar a sua integração no mercado, para além dos modelos de negócio a serem utilizados.

É também realizada uma comparação com o H.264, tanto a nível de mercado, como de arquitectura.

Palavras-Chave – WebM, VP8, Vorbis, H.264, AAC;

1. INTRODUÇÃO

Este projecto apresenta uma nova opção para a codificação de vídeos. É algo muito recente, mas com o patrocínio da Google rapidamente foi discutido pelo mundo tecnológico.

Os *codecs* utilizados pelo projecto foram lançados sem *royalties*, segundo uma licença BSD. A libertação da licença levou este projecto a ter o apoio de muitas empresas que irão ajudar o desenvolvimento deste *codec* e a sua integração no mercado.

Este projecto tenta competir com o H.264, na internet, a sua implementação está a ser dedicada a este mundo e tenta ser o futuro.

Existe a necessidade de perceber se o patrocínio da Google e do formato aberto do VP8 serão suficientes para reunir o apoio para desenvolver o *codec* até uma versão segura de poder destronar o H.264 no mercado.

Para conseguir competir no mercado terá de ter uma qualidade semelhante à do H.264, que tem o apoio das empresas que recebem as suas *royalties*.

2. HISTÓRIA

Este projecto foi iniciado em 19 de Maio de 2010, com a aquisição, por parte da Google, da empresa On2 Technologies por \$133M, que é a patrocinadora do mesmo.

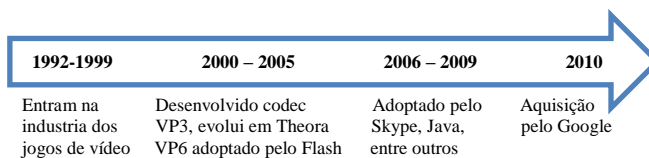


Figura 1 – A evolução histórica da On2. ^[1]

Esta empresa tem um longo historial na cominação por vídeo. Foi fundada no início dos anos 90, desenvolveram o seu primeiro *codec*, com o nome de TrueMotion S, que foi usado em alguns jogos da Sega. No início dos anos 2000 foi adquirida a empresa Metavision, o que permitiu lançar o primeiro *codec* da série VP X, o VP3, que serviu de fundação para os futuros *codecs*. Passado um ano foi decidido libertar as patentes da tecnologia de compressão da versão 3.2 VP para a comunidade *open-source*, o que originou o *codec* Theora.

No mesmo ano foi lançado o VP4 que teve licença exclusiva por parte da RealNetworks. Em 2005 a Macromedia (posteriormente adquirida pela adobe) decidiu integrar o VP6 na tecnologia Flash e a Skype inclui este *codec* para permitir ter uma comunicação vídeo directa. Muitos vídeos foram inseridos no formato VP7 no Youtube, da Google, que agora controla a On2 e o seu último *codec*, o VP8. ^[2]

A cronologia da On2 pode ser observada figura 1.

Para além do VP8, este projecto usa o *codec* Vorbis, da Xiph para o áudio e *containers* da Matroska, ambas empresas de formato aberto e livre.

Com capacidade publicitária da Google na Web, a sua aposta no youtube e nos smartphones, este projecto facilmente reuniu a atenção e o apoio de várias empresas importantes que fazem promessas de usar estes *codecs* nas suas próximas versões e ajudar no seu desenvolvimento, de modo a obter uma versão estável e globalmente utilizada.

3. ARQUITECTURA

3.1. VP8

Como a maior parte das tecnologias de compressão modernas, o VP8 é baseado na decomposição de tramas em sub-blocos de pixéis, fazendo uma predição destes através dos blocos anteriormente construídos e ajustadas através da DCT. Estes sistemas reduzem o *bitrate* explorando a correlação temporal e espacial dos sinais de vídeo. É mais eficiente especificar a localização de uma porção visualmente semelhante de uma trama anterior do que especificar o próprio valor de um pixel.

Ao contrário dos antigos formatos MPEG, o VP8 indica os valores exactos dos pixéis reconstruídos. O algoritmo usa exclusivamente operação com inteiro de processação fixa. Isto facilita a verificação da exactidão da implementação de um decodificador, bem como evita as incoerências visuais difíceis de prever entre tais implementações.^[3]

3.1.1. Descrição do formato

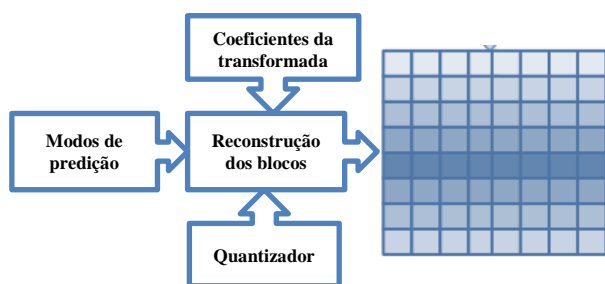


Figura 2 – Esquema dos elementos a ter em consideração na reconstrução de um bloco.^[4]

O VP8 funciona exclusivamente com o formato de imagens 8-bits YUV 4:2:0. Neste formato a crominância, com um bloco de pixéis 2x2, são representadas 4 amostras de luminância. Existe ainda espaço reservado no cabeçalho para um segundo formato de cores. Como é habitual, os pixéis são uma grande matriz de *bytes*, armazenados de baixo para cima, da esquerda para a direita. Para um *bitrate* baixo, o sistema pode decidir reduzir a resolução, para aumentar a eficiência da compressão. Antes da saída o sinal sofre um aumento de escala na sua reconstrução.

Internamente, o VP8 decompõe cada trama de saída numa matriz de macroblocos. Estes são uma matriz de pixéis, com Y de dimensões 16x16 e U e V com dimensões de 8x8. Estes macroblocos são futuramente decompostos em sub-blocos 4x4. Os pixéis são sempre tratados, pelo menos, ao nível dos sub-blocos, que são as estruturas mais básicas a ser usadas no algoritmo do VP8. A DCT é usada para os sub-blocos de 16Y, 4U e 4V. É ainda usada uma outra transformada de Fourier, a WHT (Walsh-Hadamard Transform) para codificar a matriz 4x4 comprimindo a média da intensidade dos 16 sub-blocos Y dos macroblocos,

que não é usada em MPEG. A ilustração da reconstrução de blocos pode ser verificada na imagem 2.^[3]

3.1.2. Tipos de tramas comprimidas



Figura 3 – Selecção dos modos de predição, por parte do codificador.^[5]

Apenas existem dois tipos de tramas no VP8. As tramas intra (I) são descodificadas sem ter como referência outras tramas da sequência. Estas são as tramas que permitem ter pontos de acesso aleatório, na reprodução do vídeo. As tramas inter (P) são descodificadas com referência a tramas anteriores, incluindo a última intra. Uma descodificação correcta desta trama depende da descodificação correcta de todas as tramas que usa como referência. Consequentemente o algoritmo de descodificação não é tolerante a tramas perdidas ou corruptas. Neste ambiente a descodificação correcta não seria possível até a recepção de uma trama intra, pois é independente.^[6]

Ao contrário do MPEG, não existe predição bidireccional. Assim não existem tramas que usem como referência uma trama temporalmente futura a esta. Estas são conhecidas como B, no MPEG. Por outro lado, o VP8 aumenta estas noções com a introdução de tramas de predição alternativas, com o nome tramas golden e *altref* (*alternative reference*). Os blocos inter podem ser preditos usando os blocos da trama imediatamente anterior, bem como na trama golden ou *altref* mais recente. Estas alternativas podem ser utilizadas para superar a intolerância as tramas perdidas. A escolha dos modos de predição é ilustrada na figura 3.^[3]

3.1.3. Descrição do formato dos dados comprimidos

A entrada de um decodificador VP8 é dada por uma sequência de tramas comprimidas, em que a sua ordem é correspondente à ordem temporal. Problemas como a duração das tramas, o correspondente áudio e a sincronização são resolvidos pelo ambiente de reprodução e irrelevantes à decodificação.

A primeira trama a ser descomprimida é intra, que pode ser seguida por um número de tramas inter. A reconstrução correcta destas depende de todas as anteriores, até a próxima intra recomençar o processo. A separação em partições dos dados da predição e coeficientes permite uma maior flexibilidade na implementação da descompressão, pois pode ser armazenada informação para toda a trama e depois ser decodificada, transformada e adicionado o sinal residual a toda a trama.^[3]

3.1.4. Descrição do processo de decodificação

O decodificador do VP8 precisa de ter 4 YUV buffers para tramas de resolução pelo menos igual à codificada. Estes mantem a trama a ser reconstruída, a trama anterior, a última trama gloden e a última trama altref.

A maior parte das implementações vai desejar usar estes buffers com pixéis “invisíveis”, que estendem um número moderado de pixéis para além das margens da imagem visível. Isto simplifica a predição das tramas inter, permitindo utilizar todos os dados para predição dos blocos da imagem.^[3]

3.2. Vorbis

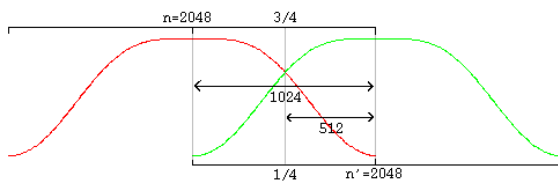


Figura 4 – Sobreposição de duas janelas.^[7]

Este codec é baseado na transformada MDCT (Modified Discrete Cosine Transform). A saída de uma MDCT em janela de uma trama é visível na figura 4. Possui um codificador desenhado segundo um complexo modelo psicoacústico e um decodificador de baixa complexidade. A decodificação é mais simples que a do mp3, mas necessita de mais memória, pois não possui um modelo probabilístico.

O áudio introduzido é dividido em tramas individuais, que são comprimidas em bruto, em pacotes não formatados. O decodificador aceita em pacotes em sequência, decodifica-os e sintetiza as tramas de áudio através destes, volta a juntar estas tramas à imagem da trama original. O Vorbis é da forma VBR (Variable Bit Rate) e assim os pacotes não

têm tamanho mínimo, máximo ou mesmo esperado. Os pacotes são desenhados de modo poderem ser cortados e mesmo assim decodificados. O mecanismo de transporte e decodificação deve permitir que o pacote seja de qualquer tamanho e que acabe antes ou depois de o decodificador o esperar.

Os pacotes em Vorbis são assim esperados serem usados com um mecanismo de transporte que fornece uma forma livre de enquadramento, sincronização, posicionamento e correcção de erros, de acordo com os objectivos do projecto.^[7]

3.3. Matroska

A Matroska Multimedia Container é uma norma aberta para um container de formato livre. Este formato suporta um número ilimitado de faixas de vídeos, músicas, imagens e legendas num único ficheiro. Tem como objectivo fornecer um formato universal para armazenamento de conteúdos multimédia comuns, como programas de TV. A Matroska tem um conceito semelhante a outros, como AVI, MP4 ou ASF, mas é completamente aberta a especificações, em que as suas implementações consistem principalmente em softwares open-source.^[8]

O formato do container do WebM tem como base o perfil do formato do container da Matroska, juntamente com os formatos de vídeo VP8 e de áudio Vorbis.^[9]

4. LICENÇA

A empresa responsável por este projecto, a Google, garante uma licença perpétua, mundial, não exclusiva, isenta de royalties e irrevogável para fazer, usar, oferecer ou vender implementações deste projecto.^[10]

A meio do ano 2010 a OSI (Open Source Initiative) expressou as suas dúvidas sobre a licença deste projecto ser open-source, pois não foi submetido para aprovação na OSI [39]. Em resposta, o Projecto WebM desassociou as patentes de copyright, oferecendo o código sob a licença BSD padrão e sob uma concessão de patentes em separado^[11]. A Free Software Foundation deu o seu aval ao WebM e ao VP8 e considera a licença do seu software compatível a GNU GPL. No início de 2010 esta mesma fundação anunciou o seu apoio oficial ao projecto WebM.^[12]

Apesar da Google ter lançado todas as patentes num formato livre de royalties, a MPEG LA, licenciadora do grupo de patentes H.264, manifestou interesse em criar um grupo de patentes para o VP8. Como resultado desta ameaça, o departamento de justiça dos estados unidos iniciou uma investigação sobre a MPEG LA e o seu papel, numa possível sufocação da concorrência.^[13]

5. MODELOS DE NEGOCIO

5.1. Apoios



Figura 5 – Principais empresas a apoiar o Porjecto WebM, desde o seu início. ^[1]

Este projecto já conta com um enorme apoio de fabricantes de *hardware*, o que permitirá desenvolver os codificadores do VP8 e diminuir o tempo de codificação, que é um dos aspectos em que o *codec* se encontra mais atrasado. Como todo este projecto está aberto, a adaptação deste ao *hardware* e a possibilidade de *hardware* dedicado irá impulsionar este projecto para o futuro. Para além do apoio do *Hardware*, existe o apoio de muitos fabricantes de *Software*. Estes irão fornecer as ferramentas necessárias para poder codificar este *codec*. Os principais apoiantes no início deste projecto podem ser observados na Figura 5. Depois do anúncio do projecto e da libertação das patentes do VP8, várias empresas mostraram a sua intenção de contribuir também para o seu desenvolvimento. ^[1]

No início do lançamento do Projecto, Kevin Lynch, da Adobe, mencionou no Google I/O, que irão incluir o *codec* de vídeo VP8 no reprodutor de Flash na sua próxima versão, que irá fornecer aos utilizadores acesso a uma semelhante qualidade de vídeos de alta definição em todos os dispositivos ligados por internet. ^[14]

Com o suporte do HTML5, a Microsoft e o decidiu apoiar a reprodução do *codec* de vídeo VP8 no IE9, bastando apenas este estar instalado no Windows. ^[15]

O Skype, que lançou recentemente a versão 5.0 do seu *software*, que usa o *codec* de vídeo VP8, onde 40% das suas chamadas são de vídeo. ^[16]

O site de vídeos Youtube, da Goolge, está a codificar todos os novos *uploads* em WebM e está também a trabalhar em codificar todo o seu catálogo de vídeos em WebM. Tendo em conta o tamanho deste catálogo (perto de 6 anos de vídeo são inseridos no site todos os dias), este processo irá ser demorado. Até agora conseguiram codificar os vídeos que originam 99% das visualizações e 30% de todos os vídeos. ^[17] Para além disto, no início de 2011, a Google

decidiu que o seu *browser*, o Chrome, irá deixar de suportar o *codec* de vídeo H.26, na continuação do seu plano de desenvolver os *codec* abertos. ^[18]

5.2. Desenvolvimento

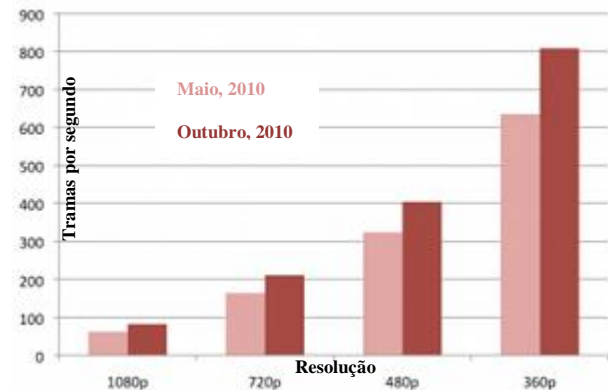


Figura 6 – Evolução da velocidade de descodificação em *single-core*. ^[19]

Com o apoios de todas estas empresas e o patrocínio da gigante Google, a evolução dos descodificadores tem se tronado bastante rápida. Em Outubro de 2010 foi lançado o *codec* SDK. Aylesbury. A evolução, em termos de velocidade de descodificação, pode ser observada na imagem 6. ^[19]

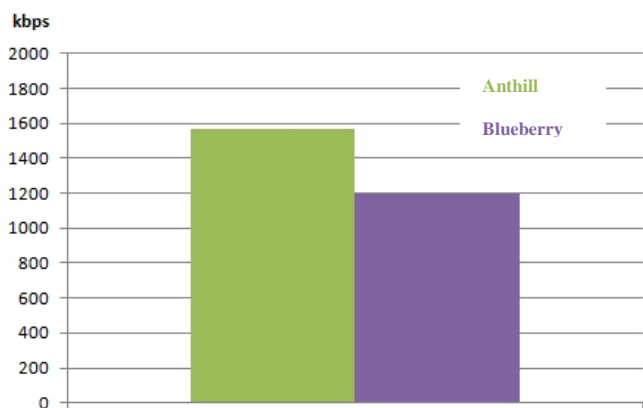


Figura 7 – Comparação entre os descodificadores Anthill e Blueberry. ^[20]

Em Maio de 2011 foi lançado uma nova versão do descodificador. A Comparação entre as duas versões é ilustrada na Figura 7. É visível o *bitrate* necessário para necessário a uma qualidade de 720p, a 30 fps, para ter 33dB de nível de qualidade PSNR. ^[20]

5.3. Novas Tecnologias

Outra ajuda para a globalização do WebM é a nova versão do HTML, o HTML5. O HTML5 introduz *tags* de média, áudio e vídeo, o que permite que o *browser* tenha um reprodutor de média sem a adição de *plug-ins*. O Youtube já faz uso destas especificações. Apesar de ainda estar em fase experimental, já podemos ver vídeos sem a instalação do Flash Player, proprietário da adobe e que a Apple quer remover por completo de todos os seus dispositivos.

Neste o momento as versões mais recentes dos *browsers* que suportam o HTML5, representam 99% do mercado. Quanto ao próprio WebM, também será suportado por estes mesmos *browsers*, à excepção do Safari, da Apple, que é a única grande empresa no mundo das comunicações e da informática que não mostrou qualquer apoio a este projecto. [1]

O desenvolvimento deste projecto está num nível incrivelmente rápido. Existem evoluções semanais significativas no *blog* oficial do WebM, onde a sua última novidade, é a introdução de vídeos 3D webM na *web*, na NVIDIA 3D Visions, com o suporte da NVIDIA. O 3D está a vingar no mercado e a crescer rapidamente. O HTML5 irá ter um reprodutor e o Youtudo vídeos em 3D. [21]

O projecto WebM possui ainda um projecto irmão, com o nome de WebP. Este projecto também está a ser desenvolvido pelo Google e é um novo formato de imagens, para compressão *lossy*. A tecnologia é baseada no *codec* de vídeo VP8 e também foi lançado segundo a licença BSD. Este pretende substituir as imagens em JPEG, oferecendo uma eficácia de compressão para a mesma qualidade de 40%. A reconstrução destas imagens é baseada nas tramas intra das especificações do VP8. [22]

6. COMPARAÇÃO COM O H.264

O Projecto WebM, têm apenas um ano e possui um potencial de crescimento enorme. Ainda é difícil comparar com as alternativas existentes no mercado.

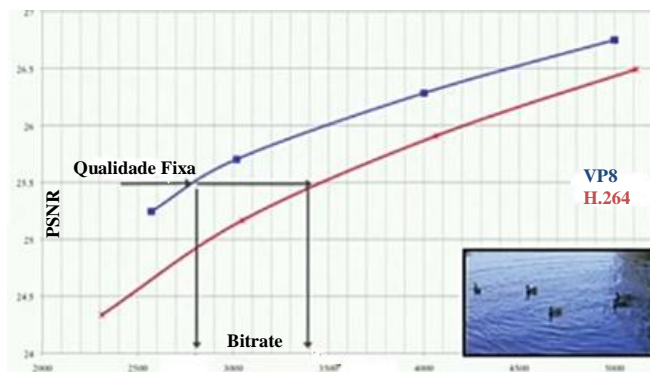


Figura 8 – Comparação do VP8 com o H.264, em nível de qualidade por ritmo binário. [1]

Foram realizados alguns testes com o desenvolvimento inicial do *codec* VP8, contra o *codec* que este está a tentar concorrer, o H.264. A Figura 8 mostra um gráfico que mede o PSNR (uma medida de qualidade) e o *bitrate* em kilobits por segundo. Neste teste com especificações feitas pelo Google (1920x1080@50fps), o VP8 usa menos 600 kilobit do que o H.264, para a mesma qualidade de imagem. [1]

- O perfil 0 da WebM, o que tem maior qualidade, é mais lento a descodificar, mas corre 15 a 20% mais rápido que o H.264.
- O perfil 1, que compete com o Baseline, é mais lento nas novas máquinas, embora seja melhor nas mais antigas, onde não se usas o SS2 e o SS3 (Conjunto de instruções de *Hardware*). [1]

6.1. Vantagens

- Formato livre de royalties, com a licença BSD;
- A qualidade dos vídeos é semelhante à do H.264;
- Tem uma óptima qualidade na reprodução dos vídeos; [1]

6.2. Desvantagens

- Apesar deste *codec* suportar *cores* múltiplos para a descodificação, não é muito eficaz a fazê-lo;
- Ainda não usa aceleração por *Hardware*, ao contrário do H.264, que utiliza nos *PCs* e telemóveis;
- O descodificador ainda não está totalmente otimizado;
- Não estão a ser utilizados os últimos conjuntos de instruções da Intel e AMD (SSE2 e SSE3); [1]

7. CONCLUSÃO

O projecto WebM possui um bom *codec*, o VP8, a qualidade de imagem é muito semelhante ao H.264, apenas os especialistas notam diferenças e não o utilizador comum. Devido aos desenvolvedores se dedicarem especialmente à qualidade de imagem, este *codec* possui um tempo de codificação/descodificação muito superior ao H.264, mas conta com o suporte de grandes empresas desenvolvedoras de *Hardware*, que irão impulsionar a evolução destas especificações.

A integração do *codec* H.264 é enorme, muito superior ao VP8, mas a Google está a ajudar o seu *codec*, principalmente pelo Youtube e Chrome. Para além do apoio da Mozilla e da Adobe. Com a libertação das patentes do VP8, agora com uma licença BSD, este projecto está a reunir ainda os apoios das empresas que não desejam pagar as *royalties* que o H.264 obriga.

Neste momento a competição é muito difícil, e a total funcionalidade do *codec* não está completa, mas uma coisa é certa, este é um *codec* para ter em conta no futuro do vídeo da internet.

8. REFERÊNCIAS

- [1] Acesso em 20/05/2011, disponível em:
<http://www.youtube.com/watch?v=poHqoBKiseY&feature=youtu.be>
- [2] Acesso em 20/05/2011, disponível em:
<http://www.tuxel.fr/?p=307>.
- [3] Acesso em 21/05/2011, disponível em:
<http://tools.ietf.org/html/draft-bankoski-vp8-bitstream-01>
- [4] Acesso em 21/05/2011, disponível em:
<http://multimedia.cx/eggs/vp8-forward-transform/>
- [5] Acesso em 21/05/2011, disponível em:
<http://www.eetimes.com/design/signal-processing-dsp/4017748/The-VP8-video-codec-High-compression-low-complexity/>
- [6] Acesso em 22/05/2011, disponível em:
<http://blog.webmproject.org/2010/07/inside-webm-technology-vp8-intra-and.html>
- [7] Acesso em 22/05/2011, disponível em:
http://www.xiph.org/vorbis/doc/Vorbis_I_spec.html#x1-30001.1
- [8] Acesso em 26/05/2011, disponível em:
<http://www.matroska.org/technical/specs/index.html#track>
- [9] Acesso em 25/05/2011, disponível em:
<http://blog.webmproject.org/2010/05/introducing-webm-open-web-media-project.htm>
- [10] Acesso em 22/05/2011, disponível em:
<http://www.webmproject.org/license/bitstream/>
- [11] Acesso em 22/05/2011, disponível em:
<http://www.computerworlduk.com/community/blogs/index.cfm?entryid=2973>
- [12] Acesso em 23/05/2011, disponível em:
<http://webmproject.blogspot.com/2010/06/changes-to-webm-open-source-license.html>
- [13] Acesso em 24/05/2011, disponível em:
<http://www.fsf.org/news/supporting-webmSmith>
- [14] Acesso em 20/05/2011, disponível em:
http://blogs.adobe.com/flashplatform/2010/05/adobe_support_for_vp8.html
- [15] Acesso em 21/05/2011, disponível em:
<http://windowsteamblog.com/windows/b/bloggingwindows/archive/2010/05/19/another-follow-up-on-html5-video-in-ie9.aspx>
- [16] Acesso em 26/05/2011, disponível em:
<http://blog.webmproject.org/2010/11/webm-video-codec-in-skype-50-group.html>
- [17] Acesso em 20/05/2011, disponível em:
<http://youtube-global.blogspot.com/2011/04/mmm-mmm-good-youtube-videos-now-served.html>
- [18] Acesso em 26/05/2011, disponível em:
<http://blog.chromium.org/2011/01/html-video-codec-support-in-chrome.html>
- [19] Acesso em 26/05/2011, disponível em:
<http://blog.webmproject.org/2010/10/vp8-codec-sdk-aylesbury-release.html>
- [20] Acesso em 26/05/2011, disponível em:
<http://blog.webmproject.org/2011/05/blueberry-vp8-hardware-encoder-ip.html>
- [21] Acesso em 26/05/2011, disponível em:
<http://blog.webmproject.org/2011/05/introducing-3d-webm-support-with-nvidia.html>
- [22] Acesso em 26/05/2011, disponível em:
<http://code.google.com/speed/webp/>



Marcelo P. Félix Nasceu em Glória do Ribatejo, Portugal, no dia 30 de Setembro de 1988. Frequenta neste momento o quarto ano do Mestrado Integrado de Engenharia Electrotécnica e de Computadores, especialização em Telecomunicações e Computadores.

Membro da assembleia geral da empresa júnior Systems Group e Presidente da

República estudantil “A Desordem dos Engenheiros”.