UNIVERSIDADE DO ALGARVE
Faculdade de Ciências Humanas e Sociais

PLATAFORMA DE FILMES COM SEQUÊNCIA ALEATÓRIA
ORGANIC FILM

Rui Paulo Mateus António

Projeto

para obtenção do Grau de Mestre em Produção, Edição e Comunicação de Conteúdos - Ramo Multimédia

Trabalho efetuado sob a orientação de:
Mirian Nogueira Tavares e Bruno Mendes da Silva

2012

UNIVERSIDADE DO ALGARVE
Faculdade de Ciências Humanas e Sociais

PLATAFORMA DE FILMES COM SEQUÊNCIA ALEATÓRIA
ORGANIC FILM

Rui Paulo Mateus António

Projeto

para obtenção do Grau de Mestre em Produção, Edição e Comunicação de Conteúdos - Ramo Multimédia

Trabalho efetuado sob a orientação de:
Mirian Nogueira Tavares e Bruno Mendes da Silva

2012

Plataforma de filmes com sequência aleatória
Organic film

Declaração de autoria de trabalho
Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

Dedicatória

Ao meu pai

Agradecimentos

Quero agradecer aos Professores Mirian Nogueira Tavares e Bruno Mendes da Silva pela forma dedicada com que orientaram este projeto, pelo entusiasmo, apoio, disponibilidade e partilha de conhecimentos que me deram.

Os meus agradecimentos a todas as pessoas que estiveram envolvidas no conjunto do trabalho realizado, colegas e docentes, que para ele em algum momento contribuíram com a sua opinião ou apoio, nomeadamente ao Professor Fernando Amaro, Professor Tiago Batista e Professora Adriana Nogueira. O meu sincero agradecimento pela disponibilidade demonstrada.

Quero ainda agradecer à Sandra, ao Francisco e ao Mateus pela paciência e carinho.

Resumo

Este projeto pretende explorar uma possibilidade de visualização fílmica, num contexto de partilha em rede, utilizando a interação em conjunto com a aleatoriedade programada. Assim, o utilizador é convidado a participar através da contribuição de conteúdos, carregando filmes, e também a interagir criando pontos de corte na edição. Esta interação do utilizador em conjunto com a aleatoriedade computacional, irá gerar os filmes com sequência aleatória: os filmes orgânicos.

O aleatório no cinema contraria a linearidade tradicional da montagem preestabelecida, permitindo a criação de uma obra aberta. Os filmes com sequência aleatória permitem múltiplas leituras e recusam qualquer chave última e estável.

Palvras-chave: Partilha de vídeos, sequência aleatória, sistema interativo, plataforma virtual, multimédia.

Abstract

This project aims to explore the possibility of viewing films in the context of network sharing, using the interaction together with the programmed randomness. Thus, the user is invited to participate by contributing with content, loading movies, and also to interact creating cutting points on film editing. This user interaction together with computational randomness, will produce the films with random sequence: the organic films.

The random film contradicts the linearity of the traditional editing, allowing the creation of an open work. Movies with random sequence allow multiple readings and refuse any last key and stability.

Keywords: Sharing videos, random, interactive system, virtual platform, multimedia.

Sumário

1. Introdução

Os avanços tecnológicos digitais e a conectividade global vieram alterar a forma como o homem se relaciona com a informação e o conhecimento. Esses avanços permitiram novas linguagens e sobretudo novas formas de interação, revolucionando os processos criativos. Com a interação, o utilizador deixa de ser mero observador e passa a fazer parte da obra, a participar nela. Embora a interatividade não seja uma novidade, o digital veio potenciar fortemente esse fator. Da mesma forma, também a aleatoriedade já largamente utilizada na criação de obras, beneficia de um alargamento nas possibilidades que só o cálculo computacional permite. É nestas possibilidades que assenta este projeto que pretende de uma forma experimental pesquisar novas formas de visualização fílmica utilizando a interação em conjunto com a aleatoriedade programada, aplicada num contexto de partilha em rede.

2. Caracterização e definição do projeto

2.1 O projeto *Organic Film*

Organic Film é um projeto experimental que apresenta, numa plataforma online, filmes com sequência aleatória. Os filmes são carregados e divididos em segmentos, pelos utilizadores. A plataforma controla, através de instruções programadas, a sequência a dar aos segmentos dos filmes em cada visualização.

2.2 Tecnologia aplicada

O projeto *Organic Film* utiliza comunicação entre três sistemas de programação: a linguagem *Action Script 3*[1], na qual é desenvolvida a interface da plataforma, o MYSQL[2] onde são guardadas e manipuladas as bases de dados e o PHP[3] que faz a ligação entre a interface e as bases de dados. Os dados referentes aos utilizadores e aos filmes são armazenados nas bases de dados.

A plataforma é alojada num servidor bem como os filmes carregados pelos utilizadores.

[1] *Action Script 3* é uma linguagem de programação orientada a objetos.
[2] O *MySQL* é um sistema de gestão de bases de dados (SGBD), que utiliza a linguagem SQL (Structured Query Language) como interface.
[3] PHP (um acrônimo recursivo para "PHP: Hypertext Preprocessor", originalmente Personal Home Page) é uma linguagem interpretada livre e utilizada para gerar conteúdo dinâmico na World Wide Web.

3. Investigação

3.1 A Montagem Fílmica

Os primeiros filmes de cinema não tinham montagem, eram constituídos por um plano único e fixo. Normalmente apresentavam uma situação do quotidiano e tinham uma duração de cerca de um minuto, que correspondia a uma bobina de filme. Estes registos, por se limitarem a imitar a perceção do real, depressa perderam o encanto inicial.

A primeira montagem data de 1896 pelas mãos de Louis Lumièr com o filme "*Démolition d'un mur*". Este filme mostra um plano com um muro a ser derrubado seguido do mesmo plano invertido, ou seja, o murro a reconstruir-se. Embora se trate de uma montagem ainda rudimentar, ficava já estabelecida uma ligação lógica entre dois planos.

Figura 3.1 - "Démolition d'um mur" de Louis Lumièr

A montagem física, isto é, a mera colagem de planos permitia pouco mais do que uma simples reprodução mecânica de imagens (Ramos, 1981, p.22). Como refere Gilles Deleuze a novidade e a essência do cinema passa pela montagem enquanto processo de significação. O plano deixa de ser fixo e espacial para passar a ser temporal (Deleuze, 2009, p 16).

É através de pioneiros como Edwin Porter (1870-1941) e David Griffith (1875-1948) que se dão os primeiros passos na montagem de significação com recurso a diferentes planos e diferentes ângulos de filmagem. Estes compreenderam o potencial da montagem para criar os elementos que distanciaram o cinema do teatro e da literatura. Griffith descobriu também a importância do ritmo cinematográfico, conceito que é posteriormente teorizado por Sergei Eisenstein (1898-1948) ficando conhecido por montagem métrica. Na Rússia, Lev Kuleshov (1899-1970), Dziga Vertov (1896-1954) e Sergei Eisenstein faziam experiências com associação de imagens. Lev Kuleshov concluiu através de uma experiência que a significação de uma sucessão de imagens depende da interpretação subjetiva do espetador. A experiência consistia em alternar um plano do rosto inexpressivo do ator Mosjoukine com outros planos: um caixão com uma criança, um prato de sopa e uma mulher deitada num divã. Na primeira combinação de imagens os espetadores atribuíram ao rosto do ator a sensação de tristeza, na segunda fome e na terceira desejo, embora se tratasse sempre do mesmo plano do rosto do ator. Efetivamente, os espetadores traziam as suas reações emocionais para a sequência de imagens e atribuíam os seus próprios sentimentos ao rosto inexpressivo do ator (Ramos, 1981,

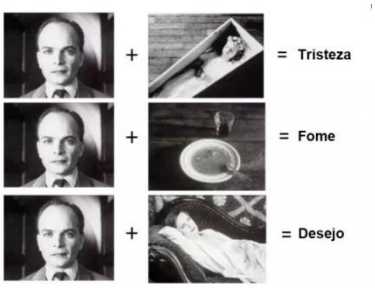


Figura 3.2 - Experiência de Lev Kuleshov

p.23). Segundo Deleuze, a evolução do cinema far-se-á pela montagem, pela câmara móvel e pela emancipação da tomada de vistas (Deleuze, 2009, p. 16). Griffith também contribuiu para a montagem na variação de planos para dar impacto emocional através do grande plano, grande plano geral, plano de pormenor, câmara subjetiva (ponto de vista do personagem) e o *travelling* (deslocação da câmara) (Gosciola, 2003). Desta forma, Griffith pretendia envolver o espetador emocionalmente através de alterações de escala nos planos, dando ao público uma emoção progressiva. Nascem assim, duas tendências de montagem: a montagem narrativa da escola americana e a montagem como produção de sentido teorizada pela escola soviética. Enquanto a primeira procurava minimizar os efeitos dos cortes, levando o espetador a imaginar a narrativa como um todo contínuo, a soviética procurava criar efeitos a partir do choque relativizando o tempo e o espaço.

A montagem fílmica continua a apresentar novas formas e tecnologias e a propor novas questões no que respeita à associação e organização de imagens cinematográficas. Independentemente da técnica utilizada, a montagem terá sempre uma importância determinante na estruturação e significação de um filme.

3.2 O aleatório nas artes

O conceito do aleatório foi já largamente estudado, sobretudo na matemática. Nas últimas décadas vários artistas fizeram experiências na literatura, pintura, música e cinema utilizando processos que fazem uso do fator aleatório na conceção das suas obras (Caires, 2004).

3.2.1 Aleatório na pintura

Segundo Carlos Sena Caires, referindo-se a uma opção de classificação de Claude Faure[4] são identificadas três formas artísticas que utilizam o aleatório nos seus modos de criação: O Expressionismo Abstrato, o Novo Realismo e os artistas que utilizam computadores e métodos de cálculo informáticos avançados como instrumento de trabalho ou pesquisa artística. Jackson Pollock (1912–1956), foi um dos pioneiros do Expressionismo Abstrato, pelo seu trabalho de pintura em movimento (*Action Painting)*. Este movimento surge na década de 1940 nos estados unidos. As telas são esticadas no chão e passam a representar o palco para a ação artística. As tintas são atiradas para a tela com gestos coreografados, utilizando diversos materiais como, pinceis, paus e facas. A pintura resulta de uma relação corporal do artista com a pintura, do encontro entre o gesto do artista e o material. Embora o resultado não seja completamente aleatório, pois existe sempre intenção no atirar da tinta, a mancha resultante é imprevisível ao contrário do que

Figura 3.3 - Jackson Pollock

[4] Claude Faure, fundador da associação Ars Technica e conselheiro artístico da Cité des Sciences et de l'Industrie entre 1983 e 1997.

acontece com o contacto direto do pincel com a tela. Também Franz Kline (1910-1962) e Willem de Kooning (1904-1997) se distinguiram nesta área. Em finais dos anos 50 é criado o Novo Realismo. Este movimento surge na europa e caracteriza-se pela utilização de objetos do quotidiano nas obras de arte. Daniel Spoerri (1930) produziu os seus primeiros *tableaux pièges* (quadros-armadilha), caracterizados por objetos agregados aleatoriamente e colados sobre peças de mobiliário e outros suportes, na posição exata onde tinham sido encontrados. Também Dieter Roth (1930-1998) demonstrou um interesse pelo aleatório, trazendo para os seus quadros, elementos da vida quotidiana como o queijo, o chocolate ou excrementos de animais[5]. Yves Klein (1928-1962) também se interessou pelo aleatório. Numa das suas experiências, ele conduziu um carro à chuva com a tela presa no tejadilho para "gravar" o efeito da chuva[6].

Figura 3.4 - Daniel Spoerri, Restaurant, 1972


Figura 3.5 - Yves Klein, Vent Paris-Nice COS 10

Apenas em 1969 surge o termo Arte Aleatória, por Hans Koetsier (1930-1991)[7], referindo-se à arte baseada nas leis da probabilidade e acaso. E em 1970, Abraham Moles (1920-1992) sugere o termo *Art Permutationnel*, que consiste na criação artística através de uso de permutações e combinações. As tecnologias digitais tiveram aqui um papel fundamental e transformador.

[5] http://www.projectomap.net/#ProjectoMAP [consultado em janeiro 2012].
[6] http://www.visual-arts-cork.com/famous-artists/yves-klein.htm [consultado em fevereiro 2012].
[7] Hans Koetsier foi um artista holandês conceitual do movimento Fluxus (movimento artístico caracterizado pela mistura de diferentes artes, primordialmente das artes visuais mas também da música e literatura. Teve o seu momento mais ativo entre a década de 1960 e década de 1970, declarando-se contra o objeto artístico tradicional como mercadoria e proclamando-se como anti arte).

O impacto das tecnologias digitais veio transformar a forma de fazer arte. Vera Molnar (1924)[8] baseia o seu trabalho numa dialética organizada entre ordem e desordem, previsível e imprevisível, entre forma e abertura, envolvendo, a arte e o novo campo de conhecimento da ciência da computação.

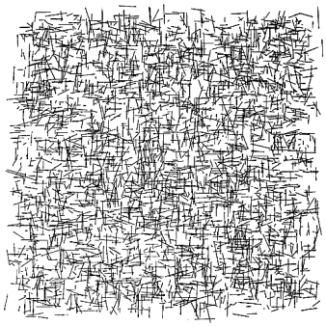


Figura 3.6 - Vera Molnar, Mondrian Dérangé, 1974

Martha Gabriel apresentou em 2007 o projeto *Locative Painting*, um site com processos generativos, que cria uma tela através da participação dos utilizadores de acordo com a sua posição geográfica. O utilizador fornece o código postal e a cor do traço pretendido e recebe via email e sms, a tela produzida resultante da participação cruzada de vários utilizadores em simultâneo na internet[9].

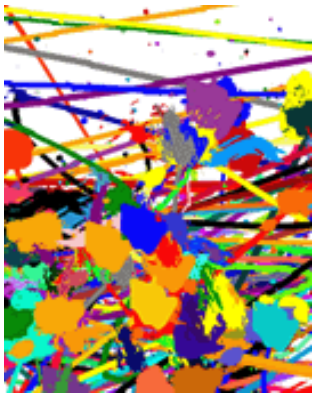


Figura 3.7 - Martha Gabriel, Locative Painting

Joshua Davis (1971) é atualmente uma referência na criação artística com recurso ao computador. As suas obras são criadas através de algoritmos produzidos pelo próprio autor. Nestes algoritmos são definidos alguns parâmetros como a palete de cor e o tipo de traço a ser utilizado. O resultado é aleatório dentro de um limite controlado, tendo em conta as regras definidas pelo autor. Todos os anos, Joshua Davis renova e modifica o seu algoritmo, criando estilos novos. Alguns algoritmos, mais antigos, são disponibilizados na plataforma *hypeFramework*[10], de forma gratuita. Esses algoritmos podem ser reutilizados para novas aplicações ou experiências.

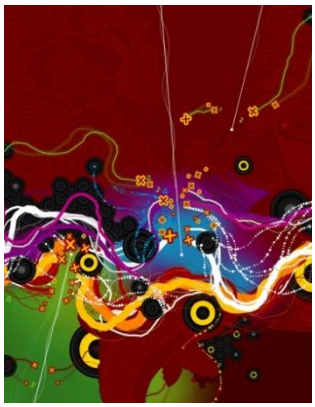


Figura 3.8 - Joshua Davis, Motorola S9, 2007

[8] Pioneira na produção de imagens computacionais nos anos 1960 em França
[9] http://www.martha.com.br/art.htm [consultado em março 2012].
[10] http://www.hypeframework.org/

3.2.2 Aleatório na música

Na música, o aleatório pode ser aplicado na composição da partitura ou na sua interpretação. Remontam ao século XVIII os jogos em que a sequência de execução dos compassos das músicas era feita através de lançamento de dados. A autoria destes jogos, designados por "Musikalisches Würfelspiele", é atribuída a Wolfgang Amadeus Mozart. Mas é a partir da segunda metade do século XX que a música aleatória atinge o seu auge tendo como referência compositores como John Cage (1912-1992), Iannis Xenakis (1922-2001), Pierre Boulez (1925) e Karlheinz Stockhausen (1928-2007). São normalmente obras abertas, nas quais a estrutura é variável e que conta com o acaso. Os procedimentos para esta aleatoriedade podem passar por jogos de dados, baralhos de cartas, cálculos matemáticos, trechos em branco, trechos alternativos e outras formas não tradicionais na composição. No caso das composições com trechos em branco, o executante completa a composição através da improvisação, tornando-se coautor. Na utilização de trechos alternativos o executante escolhe-os livremente, tornando a peça musical sempre diferente em cada atuação, como por exemplo na “Sonata para piano no. 3” (1957) de Pierre Boulez ou os “Jogos Venezianos” (1961) de Witold Lutoslawski.

John Cage utilizava o termo “indeterminação” para definir o carácter aleatório nas suas músicas. Utiliza vários equipamentos de áudio, como por exemplo o rádio, para compor as suas obras musicais. Na utilização do rádio aproveita o fator aleatório do som que é disponibilizado num determinado dia e hora, numa determinada estação. Assim, estes equipamentos passam a ser, instrumentos de

Figura 3.9 - John Cage

composição aleatória. Em 1951, John Cage compõe "Imaginary Landscape No 4" que consiste numa partitura aleatória para doze rádios e doze intérpretes. As sequências, os silêncios, a sintonização de uma estação, etc. são determinados através de um jogo baseado no Livro das Mutações, o I Ching[11]. Através do lançamento de moedas é possível definir cada uma das linhas de cada um dos 64 hexagramas. Outra marca importante nas obras de John Cage é a intervenção do público na interpretação da obra, propondo desta forma a arte participativa. Exemplo disso é a peça "Silent Piece 4'33"" de 1952, em que o público pode completar o silêncio. Este conceito veio trazer aquilo a que chamamos de interatividade, tão comum hoje na relação homem-computador e na cultura digital e multimédia.

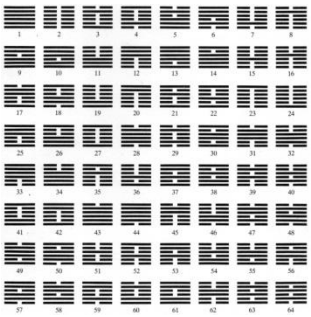


Figura 3.10 - Os 64 hexagramas do I Ching

Por outro lado, Pierre Boulez introduziu o termo música ou composição aleatória para se distinguir de John Cage. Enquanto nas peças de John Cage existia uma liberdade para os intérpretes criarem novos sons, nas peças de Pierre Boulez, apenas tinham a liberdade de escolher a ordem dos trechos previamente escritos pelo compositor. A "Sonata para piano no. 3", por exemplo, é composta por cinco segmentos em que o segmento central é fixo podendo-se alterar a ordem dos restantes, apresentando desta forma oito possibilidades diferentes de execução (ABCDE, ABCED, BACDE, BACED, EDCBA, EDCAB, DECBA e DECAB).

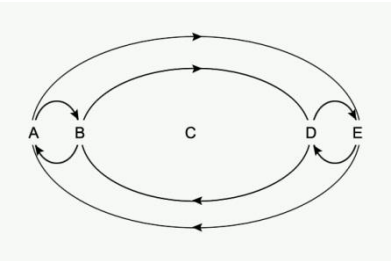


Figura 3.11 - Sonata para piano no.3

Posteriormente Iannis Xenakis começou a desenvolver composição aleatória recorrendo a cálculos complexos e sistemas aleatórios gerados por computador. Os resultados numéricos obtidos por computação tornam-se

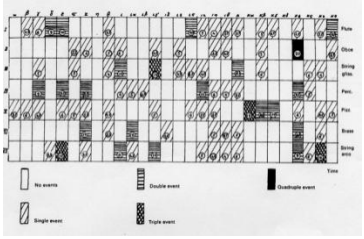


Figura 3.12 - Esquema composicional de Iannis Xenakis

[11] O Livro das mutações está associado aos princípios de combinações binárias, e pode ser lido sem o contexto de narrativa com começo/meio/fim, compreendido na forma de fragmentos aleatórios.

a sua ferramenta para composição musical, que oferecem inúmeras possibilidades de estruturação e transcrição para partituras. Contudo não se trata de um processo completamente aleatório, apenas é indeterminado nos seus detalhes pois o compositor é que rege as regras e determina o previsível. A este método deu-se o nome de processos estocásticos.

Mais recentemente a banda Radiohead compôs um tema que quando tocado ao vivo, inclui som de rádio em tempo real, não sendo previsível o som que se vai ouvir àquela hora naquela estação, à semelhança das experimentações de John Cage.

3.2.3 Aleatório na literatura

Em 1960 surge um grupo de investigação na área da escrita experimental, fundado por escritores e matemáticos chamado OuLiPo (***OU****vroir de* ***LI****ttérature* ***PO****tentiel*). Deste grupo faziam parte autores célebres, tais como Raymond Queneau (1903-1976), François Le Lyonnais , matemático (1901-1984), Jacques Roubaud (1932) George Perec (1936-1982) e Italo Calvino (1923-1985), entre outros. Este grupo cria uma corrente literária que propõe a ligação da literatura à matemática originando assim novas possibilidades de escrita. A criação da obra literária é feita seguindo regras definidas pelos autores, como por exemplo, escrever um romance utilizando apenas uma vogal (Les révenentes de Georges Pérec). O livro *La vie mode d'emploi* (PEREC, 1997) foi escrito seguindo as regras do jogo de xadrez e uma estrutura matemática, em que a passagem de um capítulo para outro é feita seguindo uma regra precisa,

a *poligrafia do cavaleiro* ou o *algoritmo do cavaleiro*. Outro projeto deste grupo foi o S+7 em que cada substantivo do texto é substituído pelo sétimo que aparece depois dele no dicionário, havendo a variante em que se podia utilizar em vez do dicionário, a Bíblia, O Capital, de Marx, etc.. Em 1961 Raymond Queneau publica o livro "*Cent mille milliards de poèmes*", que tem apenas dez folhas que correspondem a dez sonetos. Em cada folha há catorze tiras horizontais que correspondem a versos. O leitor pode criar sonetos pela combinatória de versos, podendo chegar a cem bilhões de combinações.

Figura 3.13 - Cent mille milliards de pòemes de raymond Queneau

Mais tarde, em 1981, Jacques Roubaud e Paul Braffort (1923) propõem um novo grupo – O Alamo (A*telier de Littérature Assistée par la Mathématique et les Ordinateurs*). O seu objetivo assenta no recurso à informática (procedimentos computacionais) para produção de textos (Alencar; Moraes, 2005, p.22).

Em 1989 surge o grupo LAIRE (*Lecture, Art, Innovation, Recherche*) fundado por Claude Maillard (1937), Tibor Papp (1947-2003), Philippe Bootz (1957) e Jean Marie Dutey (1942). Este grupo publica *Alire,* a primeira revista multimédia europeia e referência de poesia eletrónica, primeiro em disquete, depois em CD-ROM. Com a revista *Alire* procura-se inovar, do ponto de vista literário, pela observação de poemas animados por computador e por poemas produzidos a partir de geradores automáticos de texto.

O computador tem aqui um papel determinante e condicionante na escrita e na leitura dos poemas. Liberta a poesia da página impressa ao mesmo tempo que contribui para o alargamento do campo poético, uma vez que, os processos criativos evoluem no espaço virtual oferecido

pela máquina. O uso do computador por si só não é suficiente para se falar de uma nova poesia. Torna-se necessária a exploração das potencialidades criativas proporcionadas por este meio. As experiências de *Alire* não pressupõem uma poesia que apenas se dirige às novas tecnologias da comunicação e da informação nas estratégias subjacentes à escrita e à leitura destes textos poéticos, mas sim uma poesia que genuinamente adota esses novos meios estabelecendo uma correlação entre escritor, texto e leitor. A literatura aliada aos computadores veio facilitar e abrir novas portas à interatividade, atribuindo ao leitor o papel de coautor e, consequentemente, a rutura da linearidade anteriormente experimentada mas com menos potencialidades.

Em *Sintext* de Pedro Barbosa (1948) o leitor depara-se com textos mutáveis que permitem uma infinidade de combinações possíveis. Essa construção pode ter como base textos preexistentes ou textos criados pelo leitor. Nos textos que servem de base existem algumas regras gramaticais e de semântica em que se identificam verbos e substantivos que posteriormente podem ser substituídos por outros verbos e substantivos de forma aleatória, obtendo-se significados diferentes para a mesma matéria-prima.

Um outro autor, Júlio Cortázar (1914-1984) escreveu a obra "Rayuela" que é composta por 155 capítulos. Podem ser lidos de forma sequencial ou numa outra sequência sugerida pelo autor ou ainda por uma outra sequencia qualquer escolhida pelo leitor. Desta forma, o autor afirma que este livro são muitos livros deixando em aberto leituras diferentes consoante a sequência escolhida.

Também José Carlos Fernandes (1964), ilustrador e autor de banda desenhada, fez uso do aleatório nas suas obras. A sua obra "A pior banda do mundo", cuja leitura não é forçosamente sequencial, é composta por pequenas histórias de duas páginas autoconclusivas. O leitor pode assim, abrir o livro em qualquer página aleatoriamente e ler uma história que não está dependente do que vem antes ou depois. José Carlos Fernandes publicou outra obra também com características de leitura aleatória, "A última obra-prima de Aaron Slobodj". Neste livro, as páginas estão divididas em duas partes fisicamente independentes. Na primeira parte encontram-se as ilustrações e na segunda parte encontra-se o texto. O leitor pode ler o livro a partir de qualquer parte de texto, combinando com qualquer parte de ilustração, um pouco à semelhança de "Cent mille milliards de pòemes" de Raymond Queneau, o que torna esta obra num livro/jogo interativo.

3.2.4 Aleatório na arquitetura

Em 2008, os arquitetos Marta Brandão e Mário Sousa autores da *Mima House*, criam um conceito de casa personalizável. A ideia é combinar a estética, funcionalidade com a possibilidade de modificar a casa através de peças amovíveis. O cliente intervém no processo criativo fazendo uso de uma plataforma virtual, tornando-se um arquiteto virtual. A capacidade de adaptação não se esgota na etapa de conceção do projeto. Permite alterações

contínuas mesmo depois de a casa estar construída. A planta da casa pode ser redefinida em qualquer momento. As paredes podem ser movimentadas através das calhas metálicas no chão e teto. Os materiais do chão podem também ser alterados. Esta arquitetura orgânica permite inúmeras combinações que dependem do gosto e da imaginação do seu utilizador[12].

Os arquitetos Luke Ogrydziak e Zoë Prillinger[13] utilizam algoritmos computacionais na criação de alguns elementos dos seus projetos. Eles criam algoritmos que produzem linhas e formas de modo aleatório. De um grande conjunto de elementos gerados, é posteriormente selecionado o elemento que será aplicado no projeto. Um exemplo de construção com utilização destes elementos gerados aleatoriamente, é a "Casa Galeria". Na sua fachada podemos ver uma estrutura metálica que sugere um prolongamento da árvore que se encontra na rua.

Figura 3.14 - Casa Galeria, Ogrydziak e Prillinger

3.2.5 Aleatório no cinema

O aleatório no cinema vem romper com a linearidade tradicional do guião preestabelecido. Aplica-se sobretudo na montagem. Desta forma, em vez da continuidade de sequências temos um conjunto de mosaicos que podem ser combinados de diferentes formas.

[12] http://www.archdaily.com/192043/mima-house-mima-architects/ [consultado em abril 2012].
[13] http://www.oparch.net [consultado em abril 2012].

Em 1959 Charles Eames (1907–1978) apresenta um filme[14] que é projetado em sete ecrãs simultaneamente, deixando ao espetador o papel da montagem aleatória do filme através de escolha dos ecrãs e da imagem visualizada. A escolha é feita com base na imagem que causa mais impacto ao utilizador[15]. Em 2003, Morten Schjødt realiza o filme "Switching" que foi concebido especificamente para ser apresentado em DVD[16]. Trata-se de um filme com uma estrutura não-linear, com sequência aleatória. O utilizador interage da seguinte forma: pode mudar para outra cena em qualquer momento através da tecla *Enter* do comando do leitor de DVD. Contudo, o utilizador não tem controlo sobre qual a nova sequência que lhe será apresentada pois esta escolha é gerada aleatoriamente. A narrativa é circular com constantes repetições e não tem um final, o filme termina quando o utilizador o desejar.

Também em 2003, João Moreira Salles realiza o documentário "Nelson Freire" que está dividido em capítulos/fragmentos. Pode ser visto em DVD de forma aleatória se o espetador assim o desejar. Este documentário não tem uma estrutura sequencial, os capítulos são autónomos.

[14] Glimpses of the USA
[15] http://dignitdigit.com/category/video/ [consultado em fevereiro 2012].
[16] http://www.agencetopo.qc.ca/blog/article/distribution/ [consultado em março 2012].

4. O projeto: Organic Film

Neste projeto, o utilizador é convidado a participar com a contribuição de conteúdos, carregando filmes, e também a interagir adicionando pontos de corte nos seus filmes que resultarão posteriormente na montagem orgânica. Os cortes ficam gravados numa base de dados, para que a plataforma possa gerir a sequência do filme com base nos cortes definidos pelo seu autor. Entende-se por autor, neste contexto, o utilizador que carregou o filme e não o autor do filme original. O autor pode em qualquer momento apagar ou criar pontos de corte. Estes cortes definem os segmentos do filme, sendo que um filme sem cortes tem apenas um segmento, um filme com um corte tem dois segmentos e assim sucessivamente. Na linguagem cinematográfica, estes segmentos são definidos por cenas, termo que será utilizado doravante. A plataforma apresenta

as respetivas cenas do filme com uma sequência aleatória, recorrendo para isso a um algoritmo computacional. O algoritmo tem a especificidade de gerar a sequência aleatória sem, no entanto, repetir qualquer uma das cenas.

O total de arranjos possíveis num filme depende do número de cenas. Um arranjo com n elementos é dado pelo número de sequências que podemos constituir a partir de um conjunto com p elementos, e em que as sequências diferem entre si pela ordem dos elementos que as constituem, não podendo estes serem repetidos nenhuma vez.

$$^{n}A_{p}=\frac{n!}{(n-p)!} \quad p \leq n \wedge p,n \in \mathbb{N} \qquad (1)$$

Uma vez que todas as cenas do filme vão ser apresentadas, temos n=p, pelo que o total de arranjos é-nos dado por (2):

$$n! \qquad (2)$$

O aumento de cortes e cenas num filme aumenta exponencialmente o número de arranjos possíveis. A probabilidade de se visionar a mesma sequência é muito diminuta e esta probabilidade diminui ainda mais à medida que aumentam as cenas num filme. Assim, um filme, com por exemplo 10 cenas tem 3628800 arranjos possíveis. Num filme com 20 cenas o número de arranjos aumenta para 2432902008176640000.

O filme é apresentado sempre de forma diferente. Do fator aleatório resulta uma sequência fílmica que apresenta ou poderá apresentar significados diferentes. Assim, cada filme é muitos filmes à semelhança da obra literária "Rayuela" de Julio Cortázar. Nesta plataforma, um filme é um corpo que respira, que pulsa e que tem vida própria. É,

portanto, um filme orgânico. Esta abordagem não segue o conceito fílmico convencional, em que há um início e um fim, seguindo uma trajetória fixa e predefinida pelo criador da obra. Nesta plataforma de filmes orgânicos, os diferentes segmentos/cenas que constituem o filme são ligados sem ter em conta o espaço e o tempo inicial da obra. Esta ligação entre as diferentes cenas de forma não-linear assemelha-se ao conceito do rizoma de Deleuze. Deleuze define o rizoma como sendo um sistema não hierárquico, não central, sem memória organizadora, unicamente definido por uma circulação de estados. Ele refere ainda que o rizoma funciona por variação, construído de forma desmontável, conectável, reversível, modificável, e com múltiplas entradas e saídas, com uma linha de voo própria. Através do princípio de conexão, Deleuze estabelece que qualquer ponto de um rizoma pode ser conectado com qualquer outro. O projeto *Organic Film* também assenta nesta ideia de antigenealogia ou antimemória, como o rizoma de Deleuze (Deleuze; Guattari, 2004, p. 21-49).

5. Implementação

O projeto foi elaborado em linguagem de programação Action Script 3 em ambiente Flash Builder da Adobe. Foi também utilizado MySQL para gerir as bases de dados e PHP para fazer a ponte entre as bases de dados e a aplicação. Para o alojamento da plataforma seria adequado um servidor dedicado, uma vez que este poderia ser configurado e preparado para implementar o *streaming* do vídeo. O *streaming* iria permitir visualizar os filmes sem necessidade de espera pelo carregamento dos mesmos. Na impossibilidade de pagar um servidor dedicado, foi criado um de raíz[17]. Contudo, devido à fraca largura de banda de *upload* existente nas linhas ADLS os resultados não permitiram a implementação com os resultados esperados. A largura de banda de *upload* limitada a 1Mbps não

[17] www.organicfilm.dynip.sapo.pt

permite a transferência de filmes com resoluções superiores a 320x240 e 16 bits de som, o que inviabiliza por exemplo a visualização em *full screen*. Embora um filme com esta resolução apenas necessite de um bitrate em média de 200 Kbps, os filmes inseridos na plataforma *Organic Film* aumentam substancialmente de tamanho para cerca de 700 Kbps porque necessitam de um *keyframe* em cada *frame* para poderem ser editados *frame* a *frame*. Por esta razão optou-se por alojar a plataforma num servidor partilhado[18] que tem como única desvantagem o facto de ser necessário carregar o filme na íntegra antes da sua visualização. Este tipo de alojamento não limita a qualidade dos filmes carregados para a plataforma, contudo quanto maior o filme, quer em duração quer em resolução, maior será o tempo de espera para a sua visualização.

Os filmes e imagens não são gravados na base de dados, apenas a sua localização no servidor. Desta forma a base de dados está otimizada, permitindo um rápido acesso. Os ficheiros de filmes e imagens ficam gravados numa pasta no servidor. A base de dados é constituída pelas seguintes tabelas: Utilizador, Filmes e Cortes. Na tabela Utilizador são armazenados o nome, o código de identificação e a palavra-chave de cada utilizador. Na tabela Filmes são armazenados o código de identificação do filme, o nome do filme, o código do utilizador que carregou o filme, o caminho do ficheiro do filme, o caminho do ficheiro da imagem que corresponde ao filme. Na tabela Cortes são armazenados os valores que correspondem aos pontos de corte das diferentes cenas. O registo de utilizadores é armazenado na tabela Utilizadores. Quando um utilizador inicia uma sessão, é feita a verificação quanto à existência

[18] www.organicfilm.net

do nome de utilizador e da correspondência com a palavra-chave. As palavras-chave são introduzidas na base de dados utilizando o método de encriptação md5 para uma maior segurança.

Os filmes carregados para a plataforma podem estar nos formatos mais comuns: AVI, MOV, MP4 e MPG, FLV, F4V e WMV. No momento do seu carregamento para a plataforma, é gerada uma imagem que representa a imagem de apresentação do filme. Esta imagem coincide com o momento temporal central do filme. Aquando da edição dos filmes, todos os pontos de corte são guardados na base de dados. É a partir desta informação que é feita a montagem aleatória pela plataforma. Esta montagem é feita utilizando um algoritmo que dispõe os diferentes segmentos do filme de forma aleatória, com permutação de elementos, e portanto sem repetição de segmentos numa apresentação.

5.1 Interface

A página de entrada apresenta-se da seguinte forma:

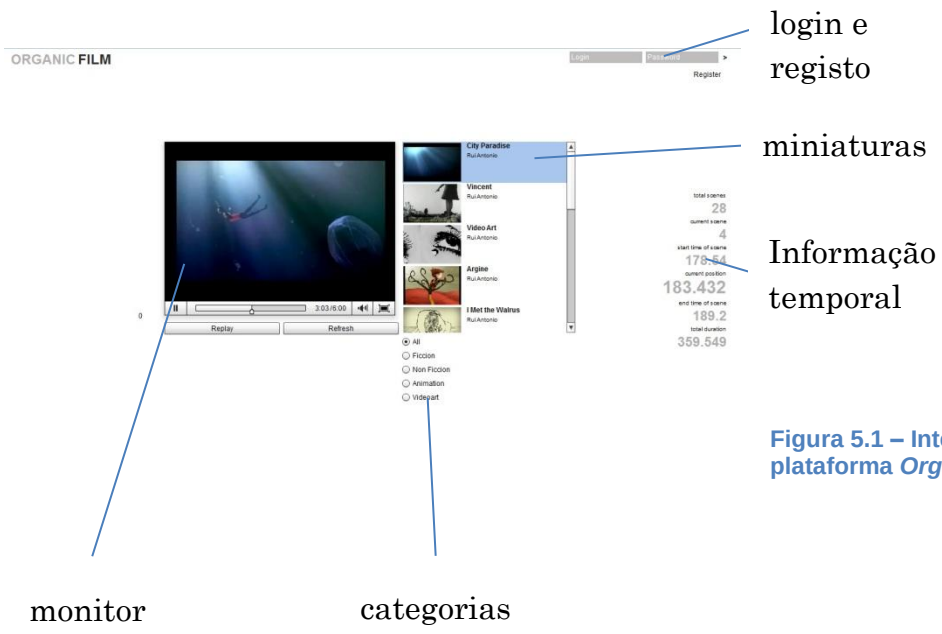


Figura 5.1 – Interface da plataforma *Organic Film*

O design escolhido para o projeto pretende ser simples, de forma a focar o utilizador no essencial: os filmes.

5.1.1 Login e registo

Para carregar ou editar filmes é necessário identificação e autenticação do utilizador através de **Nome de Utilizador** e **Palavra passe**. Os utilizadores poderão registar-se para criarem estes dados necessários ao *login*. Os dados são codificados de forma a serem entendidos apenas pelo sistema informático.

```
<?php

function novoUtilizador ($name,$login,$pass,$email)
{
        mysql_query("INSERT INTO users
        (name_user,login_user,pass_user,email_user)
        VALUES
        ('$name','$login','md5($pass)','$email')");
        }

function login ($login, $pass) {
        $res=mysql_query("SELECT *  FROM users
        WHERE login_user='$login' AND
        pass_user='md5($pass)'");
        $num=mysql_num_rows($res);
        if ($num==0) return 0;
        else return mysql_fetch_object($res);
        }

?>
```

5.1.2 Pesquisa por categorias

A pesquisa por categorias (ficção, não-ficção, animação e vídeo-arte) filtra a lista dos filmes, apresentando apenas os filmes pertencentes à categoria selecionada.

5.1.3 Criação de miniaturas

A função para criação de miniaturas, inserida em *upload.php*, gera uma imagem a partir do filme. Nesta função, é determinada a duração do filme para que a imagem represente um *frame* do meio do filme.

```
function getthumbnail( $in, $out ){
        if(file_exists($out)){unlink($out);}
        $movie = new ffmpeg_movie($in);
        $duration = $movie->getDuration();
        $cmd = "/usr/local/bin/ffmpeg -ss
        ".intval($duration/2)." -i $in -sameq -vframes 1
        -s 300x200 $out 2>&1";
        $fh = popen( $cmd, "r" );
        while( fgets( $fh ) ) { }
        pclose( $fh );
}
```

O resultado será uma imagem no formato JPG[19] com dimensões 300x200 pixéis gravada na mesma pasta dos filmes.

[19] JPG é um método comum usado para comprimir imagens fotográficas.

5.1.4 Informação temporal

A informação temporal apresenta ao utilizador vários valores sobre o filme que está a ser visionado: o número total de segmentos, o segmento atual, a duração do filme, a posição inicial do segmento atual, a posição final do segmento e a posição de leitura atual. Estes valores refletem o funcionamento do algoritmo de leitura da sequência do filme, permitindo em cada instante identificar qual o segmento que está a ser apresentado bem como o número de segmentos que ainda faltam.

5.1.5 Monitor

O monitor dispõe de cinco botões de controlo: visionamento do filme, ajustamento de volume de som, modo ecrã inteiro, repetição da sequência visionada e nova sequência aleatória.

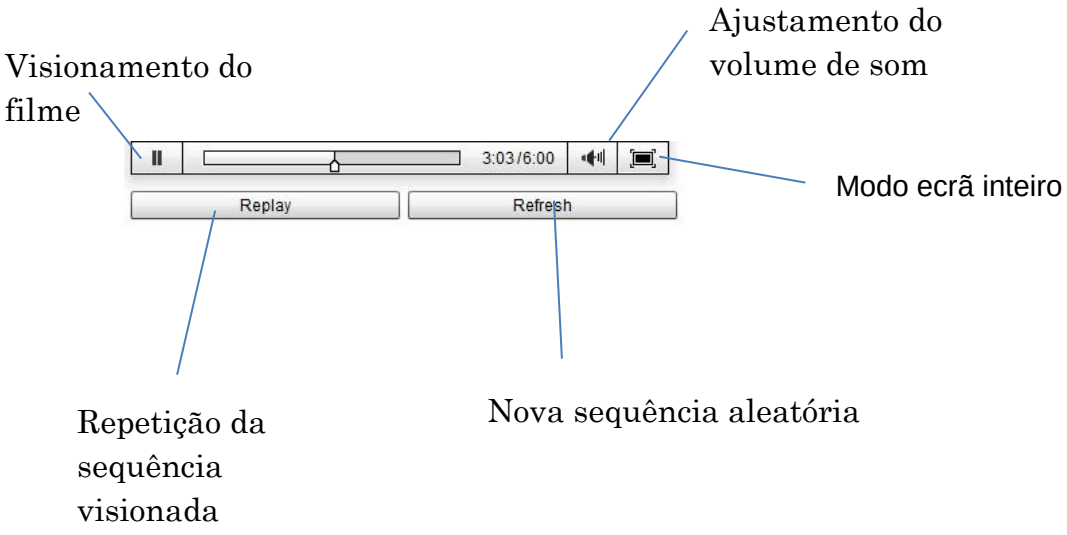


Figura 5.2 –botões de monitor

5.1.6 Edição

Figura 5.3 – Interface de edição

A página de edição é composta por monitor, botões de controlo e lista de cortes. Na edição, o utilizador tem a possibilidade de avançar ou retroceder fotograma a fotograma para efetuar cortes precisos. A lista de cortes permite ao utilizador a deslocação para um determinado ponto de corte no filme e também a sua eliminação.

5.1.7 Carregamento de filmes

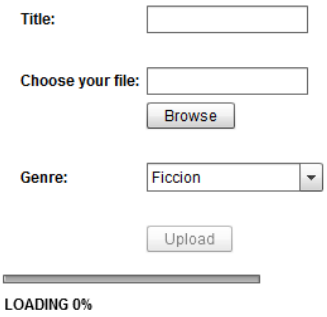


Figura 5.4 – Interface de carregamento de filmes

O carregamento de filmes permite selecionar um ficheiro a partir de um botão de localização, atribuir um título ao filme e especificar o género a que pertence.

5.2 Criação das tabelas de base de dados

São necessárias três tabelas para a plataforma *Organic Films*: Utilizadores (users), filmes (movies) e cortes (splits). Nas tabelas dos utilizadores e filmes existe uma chave primária que identifica cada um deles e é única (não existem dois utilizadores ou filmes iguais).

```
CREATE TABLE `users` (
  `id_user` int(99) NOT NULL AUTO_INCREMENT,
  `name_user` varchar(60) NOT NULL,
  `login_user` varchar(12) NOT NULL,
  `pass_user` varchar(16) NOT NULL,
  `email_user` varchar(20) NOT NULL,
  `foto_user` varchar(40) DEFAULT NULL,
  `data_regist_user` date DEFAULT NULL,
  PRIMARY KEY (`id_user`)
)


CREATE TABLE `movies` (
  `movieId` int(11) NOT NULL AUTO_INCREMENT,
  `userId` int(11) NOT NULL,
  `username` varchar(255) NOT NULL,
  `title` varchar(255) NOT NULL,
  `source` varchar(255) NOT NULL,
  `thumb` varchar(255) NOT NULL,
  `width` int(11) NOT NULL,
  `height` int(11) NOT NULL,
  `genre` varchar(20) NOT NULL,
  PRIMARY KEY (`movieId`)
)


CREATE TABLE IF NOT EXISTS `splits` (
  `movieId` int(11) NOT NULL,
  `split` double NOT NULL,
  KEY `movieId` (`movieId`)
)
```

5.3 Comunicação com as bases de dados

A comunicação entre a interface e a base de dados é feita através de funções em PHP. A interface envia ao PHP a identificação do filme ou do utilizador através do seguinte código:

```
moviesplitsXmlData.send({m_Id:movie_id});
```

Neste exemplo, a variável *movie_id* representa um valor único que identifica um determinado filme.
A função PHP recebe esse valor, extrai a informação necessária da base de dados e devolve o resultado à interface sob a forma de XML[20]:

```
<?php
        require "DB.php";
        $moviebase = 'movies/';
        header( 'content-type: text/xml' );
        $dsn='mysql://organic1_admin:0rg4n1c@localh
        ost/organic1_organicfilm';
        $db = @DB::connect( $dsn );
        if ( PEAR::isError( $db ) ) { die($db-
        >getMessage()); }
?>
<splits>
        <?php
                $v=$_GET['m_Id'];
                $res = $db->query( 'SELECT movieId,
                split FROM splits' );
```

[20] O XML é um formato para a criação de documentos com dados organizados de forma hierárquica.

```
                while( $row = $res->fetchrow( ) ) {
                        if ($row[0]==$v){
        ?>
                                <split movieId="<?php
                                echo( $row[0] ) ?>"
                                split="<?php echo( $row[1] )
                                ?>"  />
                        <?php
                        }
                }
                        ?>
</splits>
```

5.4 Sequência aleatória

A sequência aleatória é definida pelo seguinte algoritmo:

```
for (var ii:int = al; ii>=0; i--){
        var j:int = Math.round(Math.random()*ii);
        shuffledSplits.addItem(movieSegment[j]);
        movieSegment.removeItemAt(j);
}
```

A variável *al* representa o total de elementos do vetor *movieSegment* (vetor que contém todos o valores correspondentes aos cortes de um filme). A função *Math.random* gera um número aleatório compreendido entre 0 e *ii* (variável que inicialmente é igual a *al*) e guarda o resultado na variável *j*. *shuffledSplits* é o vetor que irá

guardar a sequência aleatória. Consideremos um filme com duração de 3 minutos, com 5 cortes:

Cortes no filme

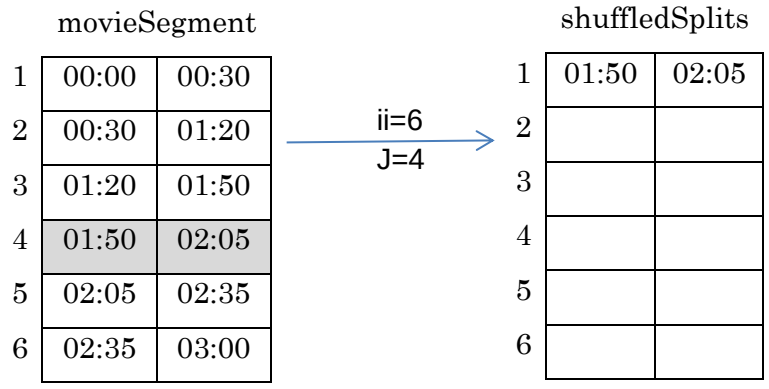
1º corte	00:30
2º corte	01:20
3º corte	01:50
4º corte	02:05
5º corte	02:35

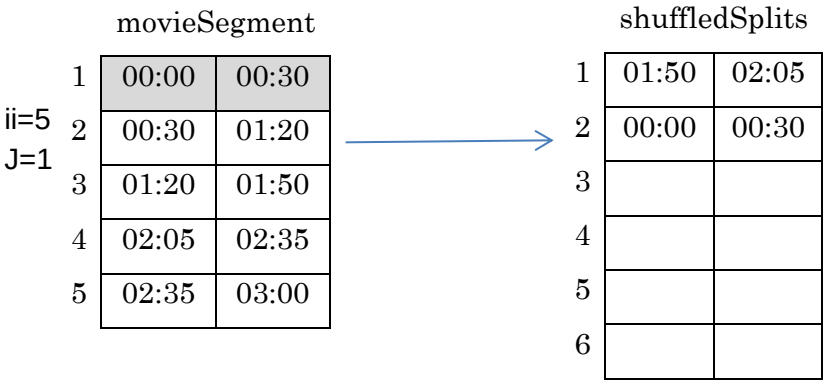
portanto com 6 segmentos:

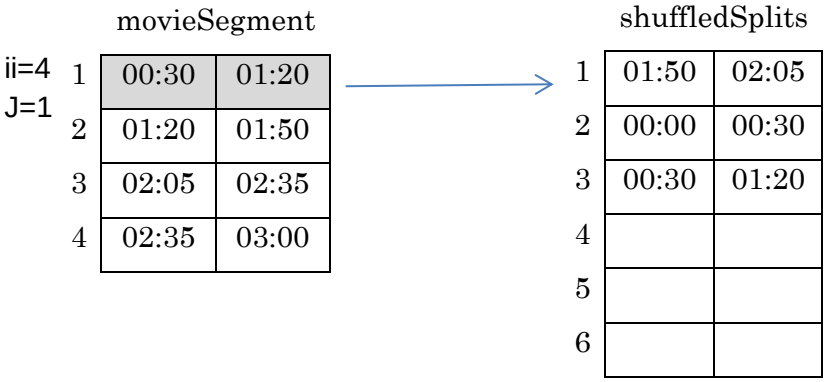
movieSegment

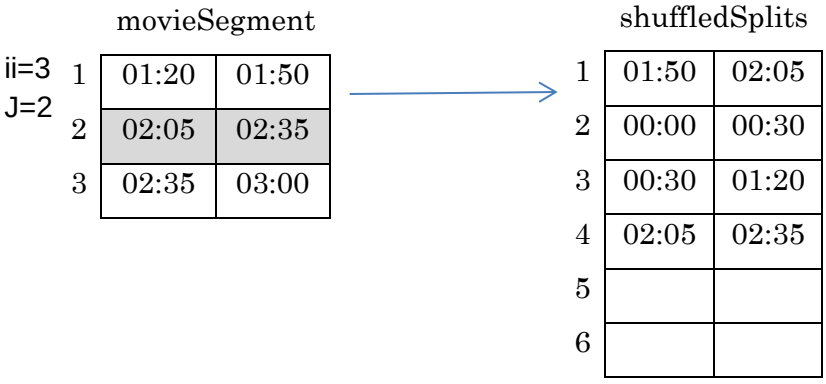
	Início	Fim
1º segmento	00:00	00:30
2º segmento	00:30	01:20
3º segmento	01:20	01:50
4º segmento	01:50	02:05
5º segmento	02:05	02:35
6º segmento	02:35	03:00

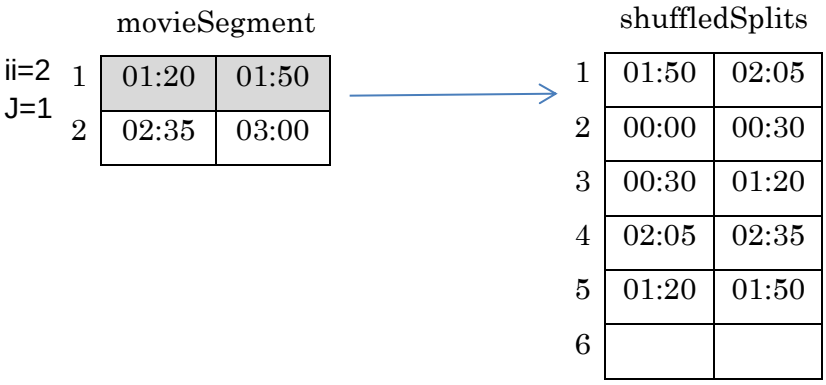
Exemplo de funcionamento do algoritmo para este filme com 6 segmentos:

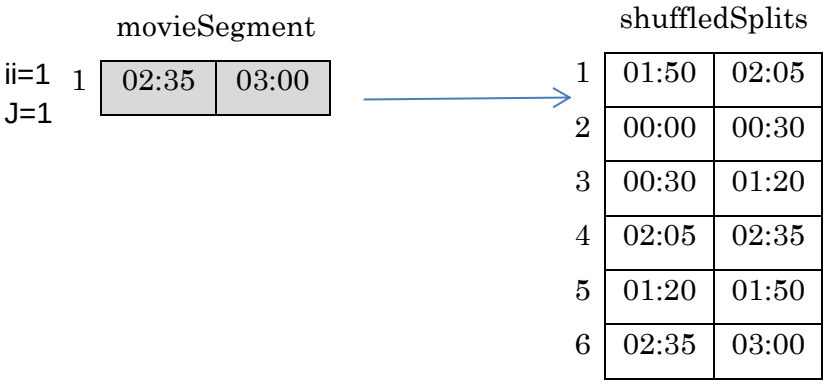



movieSegment
shuffledSplits
ii=5
J=1
1 00:00 00:30
2 00:30 01:20
3 01:20 01:50
4 02:05 02:35
5 02:35 03:00
1 01:50 02:05
2 00:00 00:30
3
4
5
6


movieSegment
shuffledSplits
ii=4
J=1
1 00:30 01:20
2 01:20 01:50
3 02:05 02:35
4 02:35 03:00
1 01:50 02:05
2 00:00 00:30
3 00:30 01:20
4
5
6


movieSegment
shuffledSplits
ii=3
J=2
1 01:20 01:50
2 02:05 02:35
3 02:35 03:00
1 01:50 02:05
2 00:00 00:30
3 00:30 01:20
4 02:05 02:35
5
6


movieSegment
ii=2
J=1
1
2
01:20
01:50
02:35
03:00
shuffledSplits
1
2
3
4
5
6
01:50
02:05
00:00
00:30
00:30
01:20
02:05
02:35
01:20
01:50


movieSegment
ii=1
J=1
1
02:35
03:00
shuffledSplits
1
2
3
4
5
6
01:50
02:05
00:00
00:30
00:30
01:20
02:05
02:35
01:20
01:50
02:35
03:00

5.5 Visualização da sequência aleatória

Existe uma função que controla qual o segmento a ser visionado. Esta função verifica quando chega ao fim de um segmento e faz avançar para o início do próximo.

```
Public function watchTime(){
        if
        (tela.currentTime>=shuffledSplits[countSegme
        nts].endSegment){
                if (countSegments==totalSegments-1){
                        tela.pause();
                        tela.removeEventListener
                        (TimeEvent.CURRENT_TIME
                        _CHANGE, watchTime);
                }
                else{
                        countSegments++;
                        goToPosition=shuffledSplits[
                        countSegments].startSegment;
                        tela.seek(goToPosition);
                        tela.play();

                }
        }
}
```

A função conta o número de segmentos já visionados e para a sua execução quando o número de segmentos visionados atinge o total de segmentos do filme.

6. Conclusões

Este projeto resulta de uma investigação teórica e prática sobre o aleatório e a interatividade aplicados ao visionamento de filmes em ambiente de rede web.

Numa primeira abordagem foi apresentada a importância da sequência fílmica concretizada normalmente através da montagem.

De seguida foi destacada a utilização do aleatório nas áreas de expressão artística como a literatura, a música, a pintura, arquitetura bem como a aplicação de processos aleatórios já realizada no próprio cinema. Como diz John Cage, o objetivo da utilização do aleatório não é escapar a fazer escolhas, mas sim contribuir conceptual e esteticamente na geração de uma obra. Contudo trata-se sempre de uma aleatoriedade controlada, pois segue as regras que o autor dita.

O projeto *Organic Film* pretende experimentar novas formas e possibilidades de visualização de filmes não seguindo, portanto, a linearidade da montagem fílmica convencional. O utilizador, mesmo que não seja o autor do filme passa a ser co-autor nesta experiência, com um papel ativo na edição. O fator aleatório deixa a obra em aberto e mostra-nos um filme diferente em cada visualização. Um filme é muitos filmes. E cada sequência apresentada leva-nos a leituras diferentes sobre o mesmo conjunto de

imagens. A sua estrutura desmontável e consequente ausência de memória organizadora aproximam estes filmes do conceito de rizoma definido por Deleuze, com múltiplas entradas e saídas. Os filmes tornam-se vivos, portanto, orgânicos.

7. Bibliografia

ALBERT, Monserrat - A Música Contemporânea. Rio de Janeiro: Ed. Salvat, 1979.

AZEVEDO, Carlos A. Moreira e Azevedo, Ana Gonçalves – Metodologia Científica: Contributos práticos para a elaboração de trabalhos académicos. Porto: C. Azevedo, 4ª Edição, 1998.

BARBOSA, Pedro e CAVALHEIRO, Abílio - Teoria do Homem Sentado. Porto: Afrontamento, 1996.

CAGE, John - Silence. Middletown: Wesleyan University Press, 2ª Edição, 1974.

CAIRES, Carlos - Título do artigo: O aleatório na imagem vídeo digital: o projecto experimental [fragments _01]. Universidade Paris 8, departamento de Artes Plásticas, 2004.

COPE, David - New Directions in Music. USA: Weveland Press, 2001.

DELEUZE, Gilles - A ImagemMovimento: Cinema 1. Lisboa: Assínio e Alvim, 2004.

DELEUZE, Gilles e GUATTARI, Félix - *Mil Planaltos Capitalismo e Esquizofrenia 2*. Lisboa: Assírio & Alvim, 2004, p. 21-49.

ECO, Umberto - A Obra Aberta - Forma e Indeterminação nas Poéticas Contemporâneas; São Paulo: Ed. Perspectiva, 1991.

ECO, Umberto - Como se faz uma tese em ciências humanas, 15.ª ed., Editorial Presença. Lisboa, 2009.

GAUNTLETT, David e HORSLEY, Ross – Web Studies. London: Hodder Education, 2ª Edição, 2004.

GERE, Charlie - Digital Culture. London: Reaktion Books, 2002.

GOSCIOLA, Vicente - Roteiro para as Novas Mídias: do game à tv interativa. São Paulo: Editora Senac, 2003

GRIFFITHS, Paul - Música Moderna: uma História Concisa e Ilustrada de Debussy a Boulez. Rio de Janeiro: Ed. Jorge Zahar, 1987.

KAC, Eduardo - Media Poetry: An International Anthology. Bristol: Intellect Books, 2007.

MANOVICH, Lev - The Language of New Media. Cambridge, Massachusetts: The MIT Press, 2001.

MARTIN, Marcel - A Linguagem Cinematográfica, Lisboa: Dinalivro, 2005.

RAMOS, Jorge L. - Sergei Eisenstein. Lisboa: Livros Horizonte, 1981.

TERRA, Vera - Acaso e aleatório na música. São Paulo: Educ, Fapesp, 2000.

VIVEIROS, Paulo - A Imagem no Cinema: história, teoria e estética, Lisboa: Edições Universitárias Lusófonas, 2ª edição, vol. 1, 2003.

VV. AA – Videoculturas de fin de siglo. Madrid: Ediciones Cátedra, 2ª Edição, 1989.

XENAKIS, I. - Formalized Music: thought and mathematics in composition. New York: Pendragon Press, 1992

8. Código Fonte

OrganicFilm.mxml

```
<?xml version="1.0" encoding="utf-8"?>


<s:Application xmlns:fx="http://ns.adobe.com/mxml/2009"

xmlns:s="library://ns.adobe.com/flex/spark"

xmlns:mx="library://ns.adobe.com/flex/mx"
                        xmlns:organic="services.organic.*"
                        xmlns:net="flash.net.*"
                        width="100%" height="100%"
backgroundColor="#FFFFFF"
                        creationComplete="initApp()"
                        backgroundColor.State1="#FFFFFF"

backgroundColor.StateAccount="#FFFFFF"

backgroundColor.StateEdit="#FFFFFF">



<fx:Script>
<![CDATA[


import flash.display.StageDisplayState;
import flash.events.FullScreenEvent;
import flash.geom.Rectangle;
import flash.net.FileFilter;
import flash.net.FileReference;

import mx.collections.ArrayCollection;
import mx.collections.Sort;
import mx.collections.SortField;
import mx.controls.Alert;
                    import mx.controls.List;
```

```
import mx.controls.VideoDisplay;
import mx.events.FlexEvent;
import mx.events.ItemClickEvent;
import mx.events.VideoEvent;
import mx.rpc.events.FaultEvent;
import mx.rpc.events.ResultEvent;
import mx.rpc.http.HTTPService;

import org.osmf.events.MediaPlayerStateChangeEvent;
import org.osmf.events.TimeEvent;

private var tempo:Number=0;

private var movie_id : String;
private var movie_genre : String;

private var idUser:Number;
private var nameUser:String;

private var _paused:Number;
private var _pausedSeek:Boolean;

private var endMovie:Number=0;
private var goToPosition:Number=0;

private var totalSegments:int=0;
private var countSegments:int=0;

private var getIn:int=0;

private var countmovies : int=0;
private var countmoviesplits : int=0;
private var firstMovie : String;

[Bindable]
private var results:Object;



private function initApp():void {
        movieXmlData.send();

}
```

```
protected function vpMediaPlayerStateChangeHandler
(event:MediaPlayerStateChangeEvent):void {

}

protected function vpcurrentTimeChange
(event:TimeEvent):void {
                        /* Se playhead for 0, passa a
100ms, assim DateFormatter nao devolve uma string
vazia.*/
        var pT:Number = tela.currentTime || 0.1;
        var tT:Number = tela.duration;

                        /* Converte playheadTime e
totalTime de segundos para milisegundos e cria um novo
objeto Date. */
        var pTime:Date = new Date(pT * 1000);
        var tTime:Date = new Date(tT * 1000);

        labelTime.text=dateFormatter.format(pTime)+ " / " +
        dateFormatter.format(tTime);
}

//Lista de filmes

[Bindable]
private var movies : ArrayCollection = new
ArrayCollection();
private var mymovies : ArrayCollection = new
ArrayCollection();

[Bindable]
private var moviesplits : ArrayCollection = new
ArrayCollection();
private var movieSegment : ArrayCollection = new
ArrayCollection();
private var shuffledSplits:ArrayCollection = new
ArrayCollection();



public function onGetMovieSplits (event : ResultEvent) :
void {

        if (event.result.splits!=null){

                if (countmoviesplits==1){
```

```
                moviesplits= new
                ArrayCollection([event.result.splits.split
                ]);
                }
        else {
                moviesplits=event.result. splits.split;
        }

        if (moviesplits.length>0 &&
        countmoviesplits>1){
                //Ordena o vetor moviesplitsvar
                dataSortField:SortField = new
                SortField();
                dataSortField.name = "split";
                dataSortField.numeric = true;

                var numericDataSort:Sort = new Sort();
                numericDataSort.fields =
                [dataSortField];

                moviesplits.sort =
                numericDataSort;
                moviesplits.refresh();
        }

        if (moviesplits.length>0 &&
        countmoviesplits>1) {

        tempo=event.result.splits.split[0].split;
        }
        else
                tempo=0;


}
else{
        tempo=0;
        getIn=-1;
}


progressBar.source=tela;
}
```

```
public function noSplits (event : ResultEvent) : void {
        tempo=0;
}



public function goToTime
(e:MediaPlayerStateChangeEvent):void {

        tela.seek(tempo);
        if(getIn != -1){
                if (tela.currentTime>=tempo &&
                tela.currentTime>0){
                        tela.removeEventListener(MediaPlayer
                        StateChangeEvent.MEDIA_PLAYER_S
                        TATE_CHANGE, goToTime);

                        // Cria segmento Movie segments - start
                         endMovie=tela.duration;

                        var al:Number = moviesplits.length;

                        for (var i:Number=0; i<=al; i++){
                                if (i==0)
                                        movieSegment.addItem( {
                                        startSegment: 0.04,
                                        endSegment:
                                        moviesplits[0].split} );
                                if (i>0 && i<al)
                                        movieSegment.addItem( {
                                        startSegment:
                                        moviesplits[i-1].split+0.04,
                                        endSegment:
                                        moviesplits[i].split} );
                                if (i==al)
                                        movieSegment.addItem( {
                                        startSegment:
                                        moviesplits[i-1].split+0.04,
                                        endSegment: endMovie} );
                        }

                        // Cria segmento Movie segments - End

                        // Mistura splits - Start
                        for(var ii:int = al; ii>=0; ii--) {
                                var j:int =
                                Math.round(Math.random()*ii);
```

```
                    shuffledSplits.addItem(movieSeg
                    ment[j]);
                    movieSegment.removeItemAt(j);
                }
                //Mistura splits - End
                goToPosition=shuffledSplits[0].startSeg
                ment;
                seekBar.value=goToPosition;
                tela.seek(goToPosition);
                tela.play();

                totalSegments=al+1;
                tela.addEventListener(TimeEvent.CUR
                RENT_TIME_CHANGE, watchTime);

            }

        }
        else {
            tela.removeEventListener(MediaPlayerStateC
            hangeEvent.MEDIA_PLAYER_STATE_CHAN
            GE, goToTime);
            tela.play();
        }

}


public function watchTime(e:TimeEvent):void {

        if (shuffledSplits.length>0){

            endsplit.text=shuffledSplits[countSegments].e
            ndSegment.toString();

            current.text= tela.currentTime.toString();
            duration.text= tela.duration.toString();

            currentsplit.text= countSegments.toString();
            totalsplits.text= (totalSegments-1).toString();

            startsplit.text=shuffledSplits[countSegments].
            startSegment.toString();

            if (tela.currentTime >=
            shuffledSplits[countSegments].endSegment){

                if (countSegments== totalSegments-1){
```

```
                        tela.pause();

                        tela.removeEventListener(TimeE
                        vent.CURRENT_TIME_CHANGE
                        , watchTime);
                    }
                    else {
                        countSegments++;

                        goToPosition=shuffledSplits[count
                        Segments].startSegment;

                        tela.seek(goToPosition);
                        tela.play();
                    }
                }
            }
            else {

                countmymoviesplitsXmlData.send({m_Id:movi
                e_id});

                moviesplitsXmlData.send({m_Id:movie_id});
            }

}


public function onGetCountMyMovies( event :
ResultEvent ) : void {

        countmovies =event.result.counting.count.num;
}


public function onGetCountMyMoviesSplits( event :
ResultEvent ) : void {

        countmoviesplits=event.result.countingsplits.countsp
        lits.num;
}


public function onGetMovies ( event : ResultEvent ) :
void {

        if (event.result.movies!=null){
```

```
                teste.text=countmovies.toString();
                if (countmovies==1){
                        firstMovie =
                        event.result.movies.movie.source.toStrin
                        g();
                        movie_id =
                        event.result.movies.movie.movieId.toStr
                        ing();
                        movies=new
                        ArrayCollection([event.result.movies.mo
                        vie]);

                }
                else {
                        firstMovie  =
                        event.result.movies.movie[0].source.toSt
                        ring();
                        movie_id =
                        event.result.movies.movie[0].movieId.to
                        String();
                        movies = event.result.movies.movie;

                        movieList.selectedIndex = 0;
                }

                tela.source = firstMovie;

                endMovie=tela.duration;

                tela.addEventListener
                (TimeEvent.DURATION_CHANGE,
                showTotalTime);


                countmymoviesplitsXmlData.send({m_Id:movi
                e_id});

                moviesplitsXmlData.send({m_Id:movie_id});
        }
        else
                movies.source = new Array();
}
```

```
public function onGetMyMovies( event : ResultEvent ) :
void {
        if (event.result.movies!=null) {
                var firstMovie : String =
                event.result.movies.movie[0].source.toString();

                tela.source = firstMovie;

                movies = event.result.movies.movie;

                movieListAccount.selectedIndex = 0;
        }
        else
                movies.source = new Array();
}


public function onChange() : void {

        tela.stop();

        tela.source =
        this.movieList.selectedItem.source.toString();

        movie_id =
        this.movieList.selectedItem.movieId.toString();

        countSegments=0;
        totalSegments=0;
        moviesplits.source = new Array();
        movieSegment.source = new Array();
        shuffledSplits.source = new Array();

        getIn=0;

        countmymoviesplitsXmlData.send({m_Id:movie_id});

        moviesplitsXmlData.send({m_Id:movie_id});

}
```

```
public function onChangeAccountToEdit() : void {

        currentState="StateEdit";
        movie_id =
        this.movieListAccount.selectedItem.movieId.toString
        ();

        moviesplits.source = new Array();

        countmymoviesplitsXmlData.send({m_Id:movie_id});

        moviesplitsXmlData.send({m_Id:movie_id});

        tela.source =
        this.movieListAccount.selectedItem.source.toString()
        ;

        tela.addEventListener(TimeEvent.DURATION_CHA
        NGE, showTotalTime);

}


public function showTotalTime(e:TimeEvent):void {

        var pT:Number = tela.currentTime || 0.1;
        var tT:Number = tela.duration;

        /* Converte playheadTime e totalTime de segundos
        para milisegundos e cria novo objeto Date. */
        var pTime:Date = new Date(pT * 1000);
        var tTime:Date = new Date(tT * 1000);

        labelTime.text=dateFormatter.format(pTime)+ " / " +
        dateFormatter.format(tTime);

        tela.removeEventListener(TimeEvent.DURATION_C
        HANGE, showTotalTime);
}


public function onChangeSplit(): void {

        goToPosition=this.splitList.selectedItem.split;
        seekBar.value=goToPosition;
        tela.seek(goToPosition);
        tempo=goToPosition;
}
```

```
public function onNext() : void {

        if ( movieList.selectedIndex >= ( movies.length - 1 ) )
                movieList.selectedIndex = 0;
        else
                movieList.selectedIndex += 1;

        tela.source =
        this.movieList.selectedItem.source.toString();
}


public function onPrevious() : void {

        if ( movieList.selectedIndex == 0 )
                movieList.selectedIndex = movies.length - 1;
        else
                movieList.selectedIndex -= 1;

        tela.source =
        this.movieList.selectedItem.source.toString();
}
//Fim de lista de filmes



//Inicio de Registo

protected function novo():void {

        novoUtilizador(txtName.text, txtLogin.text,
        txtPassword.text, txtEmail.text);

}


protected function novoUtilizador (name:Object,
login:Object, pass:Object, email:Object):void {

        novoUtilizadorResult.token =
        organic.novoUtilizador(name, login, pass, email);

}
```

```
protected function registado():void {
        currentState="State1";
}
//Fim de Registo



//Inicio de Login

protected function login():void {

        loginResult.token = organic.login(textLogin.text,
        textPass.text);

}


protected function
resultadoLogin(event:ResultEvent):void {

        if (event.result==0) textoUser.text="Login Inválido"
        else {
                textoUser.text="Hello
                "+event.result.name_user+"!";

                idUser=event.result.id_user;

                nameUser=event.result.name_user;

                teste.text=idUser.toString();

                textLogin.visible=false;
                textPass.visible=false;
                btLogin.visible=false;
                btRegister.visible=false;
                btMyAccount.visible=true;
                btLogout.visible=true;
        }
}
//Fim de Login
```

```
//Inicio de Upload
private var request:URLRequest;
private var fileRef:FileReference;
private var uploadedFile:String;

// Eventos para o botao browse
private function onBrowse() : void {

        fileRef = new FileReference();
        fileRef.addEventListener(Event.SELECT,
        selectHandler);

        try {
                var fileFilter:FileFilter = new
                FileFilter("Videos: (*.flv, *.f4v, *.mov, *.avi,
                *.mpg, *.mp4, *.wmv)", "*.flv; *.f4v; *.mov;
                *.avi; *.mpg; *.mp4; *.wmv");
                var success:Boolean =
                fileRef.browse([fileFilter]);
        }

        catch (error:Error) {
                trace("Unable to browse for files.");
        }

        function selectHandler(event:Event):void {

                videofile.text = fileRef.name;
                status.text = "Ready for upload";
                uploadedFile = fileRef.name;
                btUpload.enabled = true;

        }
}



private function onUpload():void {

        barraProgresso.label == "";
        request = new
        URLRequest("http://organicfilm.dynip.sapo.pt/up
        load.php");

        var params:URLVariables = new URLVariables();
        params.title = videotitle.text;
        params.idU = idUser;
        params.nameU = nameUser;
```

```
        params.genre = myCB.selectedItem;

        request.method = URLRequestMethod.POST;

        request.data = params;


        // Verifica o progresso de upload
        fileRef.addEventListener(ProgressEvent.PROGRESS
        , progressHandler);

        try {

                fileRef.upload(request);
                status.text = "Uploading " + fileRef.name + "
                ... ";
                browse_btn.enabled = false;
                btUpload.enabled = false;
        }

        catch (error:Error) {
                status.text = "Upload error";
        }
}

// Apresenta o progresso atual do upload
private function
progressHandler(event:ProgressEvent):void {

        barraProgresso.setProgress(Math.floor(event.bytesLo
        aded/1024),Math.floor(event.bytesTotal/1024));
        barraProgresso.label = "Uploading .. " +
        Math.floor(event.bytesLoaded/1024) + " of " +
        Math.floor(event.bytesTotal/1024) + " kbytes";";
        if (event.bytesLoaded == event.bytesTotal) {

                status.text="Upload Completed";
                browse_btn.enabled = true;
                videotitle.text="";
                videofile.text="";
        }
}
//Upload end
```

```
protected function
pla_clickHandler(event:MouseEvent):void {

        tela.play();


}



protected function
pause_clickHandler(event:MouseEvent):void {
        tela.pause();
        var numero:Number=tela.currentTime;
        teste.text=numero.toString();
        tempo=numero;
}



protected function
register_clickHandler(event:MouseEvent):void {

        currentState="StateRegister";

}



protected function
nextF_clickHandler(event:MouseEvent):void {
        if (tempo<tela.duration-0.04) {
                tempo+=0.04;
                tela.seek(tempo);
                seekBar.value=tempo;
                teste.text=tempo.toString();
        }
}


protected function
upload_clickHandler(event:MouseEvent):void {
        currentState="StateUpload";
        teste.text= idUser.toString();
}
```

```
private function seekBarValue():void {

        tempo=seekBar.value;
        tela.seek(tempo);
}


protected function
btMyAccount_clickHandler(event:MouseEvent):void {

        countmymovieXmlData.send({u_Id:idUser});

        mymovieXmlData.send({u_Id:idUser});
        currentState="StateAccount";
        teste.text= idUser.toString()+"teste";
}


protected function
btLogout_clickHandler(event:MouseEvent):void {

        textoUser.text="";
        textLogin.text="";
        textPass.text="";
        //currentState="sEntrada";
        textLogin.visible=true;
        textPass.visible=true;
        btLogin.visible=true;
        btRegister.visible=true;
        btMyAccount.visible=false;
        btLogout.visible=false;
}


protected function
btPass_clickHandler(event:MouseEvent):void {

        currentState="StateChange";
}


protected function changePass():void {


}
```

```
protected function
btBackAccount_clickHandler(event:MouseEvent):void {
        currentState="StateAccount";

        mymovieXmlData.send({u_Id:idUser});
}

protected function
backState1_clickHandler(event:MouseEvent):void {

        currentState="State1";
        countmovies=0;
        countSegments=0;
        getIn=0;
        movieSegment.source = new Array();
        shuffledSplits.source = new Array();
        movieXmlData.send();

}

protected function
splitter_clickHandler(event:MouseEvent):void {
        //Insere novo corte na base de dados

        request = new
        URLRequest("http://organicfilm.dynip.sapo.pt/wr
        ite_splits.php");


        var params:URLVariables = new URLVariables();

        params.split = tela.currentTime;
        params.mId = movie_id;

        request.method = URLRequestMethod.POST;

        request.data = params;

        var loader:URLLoader = new URLLoader();

        loader.addEventListener( Event.COMPLETE,
        loadCompleteHandler );
        loader.load( request );


        //Faz atualizacao na lista de cortes
        function loadCompleteHandler( event:Event ):void {
```

```
                countmymoviesplitsXmlData.send({m_Id:movi
                e_id});

                moviesplitsXmlData.send({m_Id:movie_id});
        }

}


protected function
btPrevFrame_clickHandler(event:MouseEvent):void {

        if (tempo>0.04) {
                tempo-=0.04;
                tela.seek(tempo);
                seekBar.value=tempo;
                teste.text=tempo.toString();
        }
}



protected function
button1_clickHandler(event:MouseEvent):void {

        var request:URLRequest= new
        URLRequest("http://organicfilm.dynip.sapo.pt/de
        lete_split.php");

        request.method = URLRequestMethod.POST;

        var loader:URLLoader = new URLLoader();
        loader.addEventListener( Event.COMPLETE,
        loadCompleteHandler );
        loader.load( request );
}


protected function loadCompleteHandler( event:Event
):void {

        Object(owner).moviesplitsXmlData.send({m_Id:Objec
        t(owner).movie_id});
}
```

```
protected function
btReplay_clickHandler(event:MouseEvent):void {

        countSegments=0;

        goToPosition=shuffledSplits[0].startSegment;
        seekBar.value=goToPosition;
        tela.seek(goToPosition);
        tela.play();

        tela.addEventListener(TimeEvent.CURRENT_TIME
        _CHANGE, watchTime);
}

protected function
btRefresh_clickHandler(event:MouseEvent):void {

        tela.stop();
        countSegments=0;
        //totalSegments=0;
        //moviesplits.source = new Array();
        movieSegment.source = new Array();
        shuffledSplits.source = new Array();

        getIn=0;

        tela.addEventListener(MediaPlayerStateChangeEve
        nt.MEDIA_PLAYER_STATE_CHANGE, goToTime);
}

protected function
btFullScreen_clickHandler(event:MouseEvent):void {



}


private function
handleGenre(event:ItemClickEvent):void {

        if (event.currentTarget.selectedValue == "All") {
                movieXmlData.send();
        }
        else {
        movieByGenreXmlData.send({m_G:event.currentTar
        get.selectedValue});
        }
```

```
}


private function progressBar_complete(evt:Event):void {

        tela.stop();
        tela.play();
                                                        if
        (currentState=="State1" ||
        currentState=="StateEdit")

        tela.addEventListener(MediaPlayerStateChangeEve
        nt.MEDIA_PLAYER_STATE_CHANGE, goToTime);
}


public function httpServiceError(evt:FaultEvent):void {

        Alert.show("no Splits");
}


]]>
</fx:Script>


<s:states>
        <s:State name="State1"/>
        <s:State name="StateRegister"/>
        <s:State name="StateUpload"/>
        <s:State name="StateEdit"/>
        <s:State name="StateChange"/>
        <s:State name="StateAccount"/>
</s:states>


<fx:Declarations>
        <s:CallResponder id="novoUtilizadorResult"
        result="registado()"/>
        <organic:Organic id="organic"

        fault="Alert.show(event.fault.faultString + '\n' +
        event.fault.faultDetail)"

        showBusyCursor="true"/>
```

```
        <s:CallResponder id="loginResult"
        result="resultadoLogin(event)"/>
        <mx:HTTPService method="get"
        url="http://organicfilm.dynip.sapo.pt/movies.php"
        id="movieXmlData" result="onGetMovies( event )" />

        <mx:HTTPService method="get"
        url="http://organicfilm.dynip.sapo.pt/mymovies.php"
        id="mymovieXmlData" result="onGetMovies( event )"
        />

        <mx:HTTPService method="get"
        url="http://organicfilm.dynip.sapo.pt/movie_splits.ph
        p" id="moviesplitsXmlData"
        result="onGetMovieSplits( event )"
        fault="httpServiceError(event)" />

        <mx:HTTPService method="get"
        url="http://organicfilm.dynip.sapo.pt/moviesByGenre.
        php" id="movieByGenreXmlData"
        result="onGetMovies( event )" />

        <mx:HTTPService method="get"
        url="http://organicfilm.dynip.sapo.pt/countmymovies.
        php" id="countmymovieXmlData"
        result="onGetCountMyMovies( event )" />

        <mx:HTTPService method="get"
        url="http://organicfilm.dynip.sapo.pt/count_movie_sp
        lits.php" id="countmymoviesplitsXmlData"
        result="onGetCountMyMoviesSplits( event )" />

        <mx:DateFormatter id="dateFormatter"
        formatString="NN:SS" />

        <s:RadioButtonGroup id="Genre"/>
        <s:RadioButtonGroup id="genre"
        itemClick="handleGenre(event);"/>
</fx:Declarations>


<s:HGroup includeIn="State1,StateAccount,StateEdit"
horizontalCenter="0" verticalCenter="-53">

        <s:VGroup >
                <s:VideoDisplay id="tela"
                includeIn="State1,StateEdit" width="400"
                height="300"
```

```
autoPlay="false" autoRewind="false"
mediaPlayerStateChange="vpMediaPlayerStat
eChangeHandler(event)"
currentTimeChange="vpcurrentTimeChange(e
vent)"/>

<mx:ProgressBar id="progressBar"
includeIn="State1, StateEdit"
complete="progressBar_complete(event);"
width="{tela.width}"
mode="polled"
source="{tela}"
label="%1 of %2 KB loaded (%3%%)"
conversion="1024"
labelPlacement="center" />

<s:HSlider id="seekBar"
includeIn="State1,StateEdit" width="400"
minimum="0" maximum="{tela.duration}"
snapInterval=".1" value="{tela.currentTime}"
click="seekBarValue()"/>

<s:HGroup>
        <s:Button id="btPlay"
        includeIn="State1,StateEdit"
        label="&gt;"
        click="pla_clickHandler(event)"
        x.State1="355" y.State1="409"
        label.State1="Play"
        label.StateEdit="Play"/>

        <s:Button id="btPause"
        includeIn="State1,StateEdit" x="273"
        y="409" label="| |"
        click="pause_clickHandler(event)"
        label.State1="Pause"
        label.StateEdit="Pause"/>

        <s:Button id="btPrevFrame"
        includeIn="StateEdit"
        label="prevFrame"
        click="btPrevFrame_clickHandler(event)
        "/>

        <s:Button id="btNextFrame"
        includeIn="StateEdit" x="355" y="447"
        label="nextFrame"
        click="nextF_clickHandler(event)"/>
```

```
                <s:Button id="splitter"
                includeIn="StateEdit" label="Cut"

                click="splitter_clickHandler(event)"/>

                <s:Button id="btReplay"
                includeIn="State1" label="Replay"
                click="btReplay_clickHandler(event)"/>

                <s:Button id="btRefresh"
                includeIn="State1" label="Refresh"
                click="btRefresh_clickHandler(event)"/>

                <s:Button id="btFullScreen"
                includeIn="State1" label="Fullscreen"
                click="btFullScreen_clickHandler(event)
                "/>

                <s:Label id="labelTime"
                includeIn="State1,StateEdit"
                color="#000000" text="00:00"/>
        </s:HGroup>

        <s:CheckBox id="checkBoxFixSegment"
        includeIn="StateEdit" x="92" y="427"
        label="keep first and last segment fixed"/>

</s:VGroup>

<s:VGroup id="vgMovies" includeIn="State1" >

        <mx:List id="movieList" includeIn="State1"
        width="300" height="330"
        borderVisible="false"  change="onChange()"
        dataProvider="{movies}"
        itemRenderer="MovieItem"
        labelField="title" ></mx:List>

        <s:RadioButton label="All"
        groupName="genre" id="genreAll" value="All"
        selected="true"/>

        <s:RadioButton label="Ficcion"
        groupName="genre" id="genreFiccion"
        value="Ficcion"/>
```

```
        <s:RadioButton label="Non Ficcion"
        groupName="genre" id="genreNonFiccion"
        value="Non-Ficcion"/>

        <s:RadioButton label="Animation"
        groupName="genre" id="genreAnimation"
        value="Animation"/>
        <s:RadioButton label="Videoart"
        groupName="genre" id="genreVideoart"
        value="Videoart"/>

</s:VGroup>


<s:VGroup id="vgMyMovies"
includeIn="StateAccount" >

        <s:Label id="titleMyMovies" width="186"
        color="#B0AEAE" fontWeight="bold"
        text="My Movies" textAlign="right"
        verticalAlign="top"/>

        <mx:List id="movieListAccount" width="300"
        height="330" borderVisible="false"
        change="onChangeAccountToEdit()"
        creationComplete="mymovieXmlData.send({u_
        Id:idUser})"
        dataProvider="{movies}"
        itemRenderer="MovieItem"
        labelField="title" textAlign="right">
        </mx:List>

</s:VGroup>


<s:VGroup id="vgCutsList">
        <s:Label id="TitleCuts" includeIn="StateEdit"
        color="#B0AEAE" fontWeight="bold"
        text="Cuts"/>

        <mx:List id="splitList" includeIn="StateEdit"
        width="150" height="330"
        borderVisible="false"
        change="onChangeSplit()"
        dataProvider="{moviesplits}"
        itemRenderer="SplitItem"
        labelField="split" textAlign="right">
        </mx:List>
```

```
        </s:VGroup>

</s:HGroup>


<s:Label
includeIn="State1,StateAccount,StateChange,StateEdit,Sta
teRegister" x="10" y="10"
color="#B0AEAE" fontSize="24" fontWeight="bold"
text="ORGANIC"
x.State1="15" y.State1="9"
x.StateAccount="15" y.StateAccount="9"
x.StateEdit="15" y.StateEdit="9"/>

<s:Label
includeIn="State1,StateAccount,StateChange,StateEdit,Sta
teRegister" x="126" y="10"
width="264" color="#292929" fontSize="24"
fontWeight="bold" text="FILM"
x.State1="131" y.State1="9"
x.StateAccount="131" y.StateAccount="9"
x.StateEdit="131" y.StateEdit="9"/>


<s:VGroup includeIn="State1,StateAccount,StateEdit,
StateRegister" id="vgLoginRegister" right="6" top="5"
width="313" height="54" horizontalAlign="right">

        <s:HGroup includeIn="State1" x="22" width="291"
        height="23">

                <s:TextInput id="textLogin"
                borderVisible="false" chromeColor="#C7C5C5"
                color="#FFFFFF"
                contentBackgroundColor="#BEBDBD"
                fontStyle="normal" prompt="Login"
                skinClass="spark.skins.spark.TextInputSkin"/
                >

                <s:TextInput id="textPass"
                borderVisible="false" color="#FFFFFF"
                contentBackgroundColor="#BEBDBD"
                fontStyle="normal" paddingBottom="3"
                paddingRight="3" displayAsPassword="true"
                prompt="Password"/>
```

```
            <mx:LinkButton id="btLogin" label="&gt;"
            click="login()" fontWeight="bold"
            paddingLeft="0" textAlign="left"/>

        </s:HGroup>

        <s:HGroup x="24" width="289" height="22" gap="0"
        paddingRight="12" verticalAlign="top">

            <s:Label id="textoUser" width="182"
            height="18" verticalAlign="bottom"/>

            <mx:LinkButton id="btRegister"
            includeIn="State1" width="108"
            label="Register"
            click="register_clickHandler(event)"/>

        </s:HGroup>

        <mx:LinkButton id="btMyAccount"
        includeIn="State1" label="My Account"
        visible="false"
        click="btMyAccount_clickHandler(event)"/>

        <mx:LinkButton id="btLogout" includeIn="State1"
        x="681" y="224" label="Logout" visible="false"
        click="btLogout_clickHandler(event)"/>

        <mx:LinkButton id="backState1"
        includeIn="StateAccount, StateRegister"
        label="back"
        click="backState1_clickHandler(event)"/>

        <mx:LinkButton id="upload"
        includeIn="StateAccount" x="640" y="120"
        label="Upload"
        click="upload_clickHandler(event)"/>

        <mx:LinkButton id="btPass"
        includeIn="StateAccount" label="Change Password"
        click="btPass_clickHandler(event)"/>

</s:VGroup>


<s:VGroup includeIn="State1,StateEdit"
id="vgOrganicData" right="25" top="250"
horizontalAlign="right" verticalAlign="middle">
```

```
        <s:Label id="totalsplitstitle"  fontSize="10"
        text="total scenes"/>

        <s:Label id="totalsplits" color="#B0AEAE"
        fontSize="24" fontWeight="bold" text="total scenes"/>

        <s:Label id="currentsplittime"  fontSize="10"
        text="current scene"/>

        <s:Label id="currentsplit" color="#B0AEAE"
        fontSize="24" fontWeight="bold" text="current
        scene"/>

        <s:Label id="startsplittitle"  fontSize="10"
        text="start time of scene"/>

        <s:Label id="startsplit" color="#B0AEAE"
        fontSize="24" fontWeight="bold" text="start time of
        scene"/>

        <s:Label id="currenttitle"  fontSize="10"
        text="current position"/>

        <s:Label id="current" color="#B0AEAE"
        fontSize="30" fontWeight="bold" text="current
        position"/>

        <s:Label id="endsplittitle"  fontSize="10"  text="end
        time of scene"/>

        <s:Label id="endsplit" color="#B0AEAE"
        fontSize="24" fontWeight="bold" text="end time of
        scene"/>

        <s:Label id="durationtitle"  fontSize="10" text="total
        duration"/>

        <s:Label id="duration" color="#B0AEAE"
        fontSize="24" fontWeight="bold" text="total
        duration"/>

</s:VGroup>


<s:Form id="frmRegister" includeIn="StateRegister"
width="424" height="305" horizontalCenter="3"
verticalCenter="-21">
```

```
        <s:FormHeading label="Register Information"/>

        <s:FormItem label="Name:" required="true">
                <s:TextInput id="txtName"/>
        </s:FormItem>

        <s:FormItem label="Login:" required="true">
                <s:TextInput id="txtLogin"/>
        </s:FormItem>

        <s:FormItem label="Password:" required="true">
                <s:TextInput id="txtPassword"
                displayAsPassword="true"/>
        </s:FormItem>

        <s:FormItem label="Retype Password:"
        required="true">
                <s:TextInput id="txtRPassword"
                displayAsPassword="true"/>
        </s:FormItem>

        <s:FormItem label="Email Address:"
        required="true">
                <s:TextInput id="txtEmail"/>
        </s:FormItem>

        <s:FormItem label="Retype Email Address:"
        required="true">
                <s:TextInput id="txtREmail"/>
        </s:FormItem>

        <s:FormItem>
                <s:Button id="btOK" label="Submit"
                click="novo()"/>
        </s:FormItem>

</s:Form>


<s:Form id="frmChangePass" includeIn="StateChange"
width="424" height="305" horizontalCenter="3"
verticalCenter="-21">

        <s:FormHeading label="Change Password"/>

        <s:FormItem label="Old Password:" required="true">
                <s:TextInput id="txtOldPass"/>
        </s:FormItem>
```

```
        <s:FormItem label="New Pasword:" required="true">
                <s:TextInput id="txtNewPass"/>
        </s:FormItem>

        <s:FormItem label="Retype New Password:"
        required="true">
                <s:TextInput id="txtRetypeNewPassword"
                displayAsPassword="true"/>
        </s:FormItem>

        <s:FormItem>
                <s:Button id="btOkChange" label="Submit"
                click="changePass()"/>
        </s:FormItem>

</s:Form>


<s:HGroup id="caixaUpload" includeIn="StateUpload"
height="244" autoLayout="false"
horizontalCenter="-38" verticalCenter="-102">

        <s:Form id="frmUpload" >
                <s:FormItem label="Title:" id="formitemtitle">
                        <s:TextInput id="videotitle"
                        editable="true" x="10" y="358"/>
                </s:FormItem>

                <s:FormItem label="Choose your file:"
                id="formitem">
                        <s:TextInput id="videofile"
                        editable="false" x="10" y="358"/>
                        <s:Button label="Browse"
                        click="onBrowse()" x="180" y="359"
                        id="browse_btn"/>
                </s:FormItem>

                <s:FormItem label="Genre:"
                id="formitemgenre">

                        <s:ComboBox id="myCB" width="140"
                        selectedIndex="0">

                        <s:dataProvider>
                                <mx:ArrayList>
                                        <fx:String>Ficcion
                                        </fx:String>
```

```
                                <fx:String>Non-
                                Ficcion</fx:String>
                                <fx:String>Animation</fx:S
                                tring>
                                <fx:String>Videoart</fx:Str
                                ing>
                            </mx:ArrayList>
                        </s:dataProvider>

                        </s:ComboBox>
                    </s:FormItem>

                    <s:FormItem id="browsearea" >
                        <s:Button id="btUpload"
                        click="onUpload()" label="Upload"
                        width="68" enabled="false" x="256"
                        y="360"/>
                    </s:FormItem>

                    <mx:ProgressBar id="barraProgresso"
                    width="206"/>

                    <s:Label id="status" color="#A91010"
                    paddingLeft="0"/>
                </s:Form>

</s:HGroup>


<s:Label includeIn="StateUpload" x="15" y="9"
color="#B0AEAE" fontSize="24" fontWeight="bold"
text="ORGANIC"/>

<s:Label includeIn="StateUpload" x="131" y="9"
width="264" color="#292929" fontSize="24"
fontWeight="bold" text="FILM"/>

<s:Label id="teste"
includeIn="State1,StateAccount,StateEdit,StateUpload"
x="236" y="457"text="Label"/>

<mx:LinkButton id="btBackAccount"
includeIn="StateChange,StateUpload, StateEdit"
right="10" top="12" label="back"
click="btBackAccount_clickHandler(event)"/>


</s:Application>
```

Organic.php

```
<?php

class Organic {

	function Organic() {
		mysql_connect("localhost","root","");
		mysql_select_db("organic_film");
		mysql_set_charset("utf8");
	}


	function novoUtilizador($name,$login,$pass,$email){
		mysql_query("INSERT INTO users
		(name_user,login_user,pass_user,email_user)
		VALUES ('$name','$login','$pass','$email')");
	}

	function todos(){
		return mysql_query("SELECT * FROM
		users");
	}

	function login($login, $pass){
		$res=mysql_query("SELECT *  FROM users
		WHERE login_user='$login' AND
		pass_user='$pass'");
		$num=mysql_num_rows($res);
		if ($num==0) return 0;
		else return mysql_fetch_object($res);
	}

}

?>
```

MovieItem.mxml

```
<?php

class Organic {


	function Organic() {
		mysql_connect("localhost","root","");
		mysql_select_db("organic_film");
		mysql_set_charset("utf8");
	}


	function novoUtilizador($name,$login,$pass,$email){
		mysql_query("INSERT INTO users
		(name_user,login_user,pass_user,email_user)
		VALUES ('$name','$login','$pass','$email')");
	}

	function todos(){
		return mysql_query("SELECT * FROM
		users");

	}

	function login($login, $pass){
		$res=mysql_query("SELECT *  FROM users
		WHERE login_user='$login' AND
		pass_user='$pass'");
		$num=mysql_num_rows($res);
		if ($num==0) return 0;
		else return mysql_fetch_object($res);
	}

}

?>
```

SplitItem.php

```
<?xml version="1.0" encoding="utf-8"?>


<mx:Canvas xmlns:mx="http://www.adobe.com/2006/mxml"
height="12" verticalScrollPolicy="off"
horizontalScrollPolicy="off">

	<mx:states>
		<mx:State name="normal" />
		<mx:State name="hovered" />
		<mx:State name="selected" />
	</mx:states>

	<mx:Script>
		<![CDATA[
			import mx.controls.Alert;
			import mx.events.CloseEvent;
			import spark.components.List;
			import flash.net.URLLoader;
			import
			flash.net.URLLoaderDataFormat;
			import flash.net.URLRequest;
			import flash.net.URLRequestHeader;
			import flash.net.URLRequestMethod;
			import flash.net.*;


			protected function send_data():void {

				Alert.show(data.label,
				"Are you sure you want to
				delete this item?",
				Alert.YES|Alert.CANCEL,
				null,
				alrt_closeHandler);
			}



			protected function
			alrt_closeHandler(evt:CloseEvent):void
			{
				switch (evt.detail) {
					case Alert.YES:
					case Alert.OK:
```

```
                                    //Apaga corte na  BD
                                    userRequest.send();
                                    Object(owner).dataProvider
                                    .removeItemAt(Object(owne
                                    r).selectedIndex);
                                    break;
                                    case Alert.CANCEL:
                                    case Alert.NO:
                                    break;
                                    default:
                                    break;
                            }
                    }

            ]]>
        </mx:Script>

        <mx:HBox width="100%"  height="100%">
            <mx:Label text="{data.split}" top="0" left="0"
            fontSize="10" />

            <mx:LinkButton id="btDelete" right="0"
            top="0"  label="x" click="send_data()"
            color="#B43D3D" fontSize="10"
            textAlign="right"/>
        </mx:HBox>

        <mx:HTTPService id="userRequest"
        url="http://192.168.1.35/delete_split.php"
        useProxy="false" method="POST">

            <mx:request xmlns="">

                <movieId>{parentApplication.splitList.s
                electedItem.movieId}</movieId>

                <split>{parentApplication.splitList.selec
                tedItem.split}</split>

            </mx:request>

        </mx:HTTPService>

</mx:Canvas>
```

count_movie_splits.php

```
<?php
require "DB.php";

$moviebase = 'movies/';

header( 'content-type: text/xml' );

$dsn =
'mysql://organic1_admin:0rg4n1c@localhost/organic1_organi
cfilm';
$db = @DB::connect( $dsn );
//if ( PEAR::isError( $db ) ) { die($db->getMessage()); }
?>

<countingsplits>
	<?php
	$v=$_GET['m_Id'];
	$c=0;
	$res = $db->query( 'SELECT movieId, split FROM
	splits' );
	while( $row = $res->fetchrow( ) ) {
		if ($row[0]==$v){
			$c++;
		}
	}
	?>
	<countsplits num="<?php echo( $c ) ?>" />
</countingsplits>
```

countmymovies.php

```
<?php
require "DB.php";

$moviebase = 'movies/';

header( 'content-type: text/xml' );

$dsn =
'mysql://organic1_admin:0rg4n1c@localhost/organic1_organi
cfilm';
$db = @DB::connect( $dsn );
?>
<counting>
        <?php
        $v=$_GET['u_Id'];
        $c=0;
        $res = $db->query( 'SELECT movieId, userId, title,
        source, thumb, width, height, genre FROM movies' );
        while( $row = $res->fetchrow( ) ) {
                if ($row[1]==$v){
                        $c++;
                }
        }
        ?>

<count num="<?php echo( $c ) ?>" />
</counting>
```

delete_split.php

```
<?php

function delete_split( $mId, $split ) {
	$con =
	mysql_connect("localhost","organic1_admin","0rg4n1c
	");
	if (!$con) {
		die('Could not connect: ' . mysql_error());
		echo "nao ligou";
	}

	mysql_select_db('organic1_organicfilm');

	$result=mysql_query("DELETE FROM splits
	WHERE movieId='$mId' AND split='$split'");

	mysql_close($con);
}


delete_split( $_POST['movieId'], $_POST['split'] );

?>
```

movies.php

```
<?php
require "DB.php";

$moviebase = 'movies/';

header( 'content-type: text/xml' );

$dsn =
'mysql://organic1_admin:0rg4n1c@localhost/organic1_organi
cfilm';
$db = @DB::connect( $dsn );
?>
<movies>
        <?php
        $res = $db->query( 'SELECT movieId, userId,
        username, title, source, thumb, width, height, genre
        FROM movies' );
        while( $row = $res->fetchrow( ) ) {
        ?>
                 <movie movieId="<?php echo( $row[0] ) ?>"
                userId="<?php echo( $row[1] ) ?>"
                username="<?php echo( $row[2] ) ?>"
                title="<?php echo( $row[3] ) ?>"
                source="<?php echo($moviebase.$row[4] ) ?>"
                thumb="<?php echo( $moviebase.$row[5] ) ?>"
                width="<?php echo( $row[6] ) ?>"
                height="<?php echo( $row[7] ) ?>" />
        <?php
        }
        ?>
</movies>
```

movie_splits.php

```
<?php
require "DB.php";


$moviebase = 'movies/';

header( 'content-type: text/xml' );

$dsn =
'mysql://organic1_admin:0rg4n1c@localhost/organic1_organi
cfilm';
$db = @DB::connect( $dsn );
?>
<splits>
        <?php
        $v=$_GET['m_Id'];
        $res = $db->query( 'SELECT movieId, split FROM
        splits' );
        while( $row = $res->fetchrow( ) ) {
                if ($row[0]==$v){
                        ?>
                         <split movieId="<?php echo( $row[0] )
                        ?>"  split="<?php echo( $row[1] ) ?>"    />
                        <?php
                }
}
?>
</splits>
```

moviesByGenre.php

```
<?php
require "DB.php";


$moviebase = 'movies/';

header( 'content-type: text/xml' );

$dsn =
'mysql://organic1_admin:0rg4n1c@localhost/organic1_organi
cfilm';
$db = @DB::connect( $dsn );
?>
<movies>
        <?php
        $v=$_GET['m_G'];
        $res = $db->query( 'SELECT movieId, userId, title,
        source, thumb, width, height, genre FROM movies' );
        while( $row = $res->fetchrow( ) ) {
                if ($row[7]==$v){
                        ?>
                        <movie movieId="<?php echo( $row[0] )
                        ?>" userId="<?php echo( $row[1] ) ?>"
                        title="<?php echo( $row[2] ) ?>"
                         source="<?php echo($moviebase.$row[3]
                        ) ?>"
                        thumb="<?php echo(
                        $moviebase.$row[4] ) ?>" width="<?php
                        echo( $row[5] ) ?>"
                        height="<?php echo( $row[6] ) ?>" />
                        <?php
                }
}
?>
</movies>
```

mymovies.php

```
<?php
require "DB.php";

$moviebase = 'movies/';

header( 'content-type: text/xml' );

$dsn =
'mysql://organic1_admin:0rg4n1c@localhost/organic1_organi
cfilm';
$db = @DB::connect( $dsn );
?>
<movies>
	<?php
	$v=$_GET['u_Id'];
	$c=0;
	$res = $db->query( 'SELECT movieId, userId, title,
	source, thumb, width, height, genre FROM movies' );
	while( $row = $res->fetchrow( ) ) {
		if ($row[1]==$v){
			$c++;
			?>
			<movie movieId="<?php echo( $row[0] )
			?>" userId="<?php echo( $row[1] ) ?>"
			title="<?php echo( $row[2] ) ?>"
			source="<?php echo($moviebase.$row[3]
			) ?>"
			thumb="<?php echo(
			$moviebase.$row[4] ) ?>" width="<?php
			echo( $row[5] ) ?>"
			height="<?php echo( $row[6] ) ?>" />
			<?php
		}
	}
?>
</movies>
<counting>
	<count num="<?php echo( $c ) ?>" />
</counting>
```

upload.php

```
<?php
require "DB.php";

function converttoflv( $in, $out ) {
        if(file_exists($out)){unlink($out);}
        $cmd = "/usr/local/bin/ffmpeg -v 0 -i $in -ar 11025 -g 1
        $out 2>&1";
        $fh = popen( $cmd, "r" );
        while( fgets( $fh ) ) { }
                pclose( $fh );
}

function getthumbnail( $in, $out ) {
        if(file_exists($out)){unlink($out);}
        $movie = new ffmpeg_movie($in);
        $duration = $movie->getDuration();
        $cmd = "/usr/local/bin/ffmpeg -ss
        ".intval($duration/2)." -i $in -sameq -vframes 1 -s
        300x200 $out 2>&1";
        $fh = popen( $cmd, "r" );
        while( fgets( $fh ) ) { }
                pclose( $fh );
}

function flv_import( $idU, $nameU, $upfile, $fname, $title,
$genre ) {
        $fname = preg_replace( '/\..*$/', '', basename( $fname
        ) );
        $finalname=str_replace(" ", "", $fname."_");
        $flvpath = "$finalname.flv";
        $thumbpath = "$finalname.jpg";

        getthumbnail( $upfile, "movies/$thumbpath" );



        $dsn =
        'mysql://organic1_admin:0rg4n1c@localhost/organic1_
        organicfilm';
        $db =& DB::connect( $dsn );
        if ( PEAR::isError( $db ) ) { die($db->getMessage()); }

        $sth = $db->prepare( 'INSERT INTO movies
        VALUES ( 0, ?, ?, ?, ?, ?, ?, ?, ? )' );
        $db->execute( $sth, array( $idU, $nameU, $title,
        $flvpath, $thumbpath, 300, 200, $genre ) );
```

```
        converttoflv( $upfile, "movies/$flvpath" );
}


flv_import( $_POST['idU'], $_POST['nameU'],
$_FILES['Filedata']['tmp_name'],
$_FILES['Filedata']['name'], $_POST['title'],
$_POST['genre'] );
move_uploaded_file($_FILES['Filedata']['tmp_name'],
"movies/".$_FILES['Filedata']['name']);


?>
File sucessfully uploaded
```

write_splits.php

```
<?php

require "DB.php";


function write_split( $mId, $split ) {

        $dsn =
        'mysql://organic1_admin:0rg4n1c@localhost/organic1_
        organicfilm';
         $db =& DB::connect( $dsn );
         if ( PEAR::isError( $db ) ) { die($db->getMessage()); }

        $sth = $db->prepare( 'INSERT INTO splits VALUES
        ( ?, ? )' );
        $db->execute( $sth, array( $mId, $split ) );
}

write_split( $_POST['mId'], $_POST['split'] );

?>
```