

Referências Bibliográficas

- [ABNT Ginga-NCL] Norma ABNT 15606-2:2007. **Televisão digital terrestre - Codificação de dados e especificações de transmissão para radiodifusão digital - Parte 2: Ginga-NCL para receptores fixos e móveis - Linguagem de aplicação XML para codificação de aplicações**, 2007. 1.1
- [ActionScript, 2006] **Actionscript 3.0**. Website. Acessado em 03/06/2009, URL: http://help.adobe.com/en_US/AS3LCR/Flash_10.0/. 2.1.5, 2.1.5
- [Adobe Director, 2009] **Adobe director 11**. Website. Acessado em 01/06/2009, URL: <http://www.adobe.com/products/director/>. 2.1
- [Adobe Flash, 2008] **Adobe flash cs4 professional**. Website. Acessado em 03/06/2009, URL: <http://www.adobe.com/products/flash/>. 2.1
- [Barbosa & Silva, 2010] BARBOSA, S. D. J.; SILVA, B. S.. **Interação Humano-Computador**. Campus/Elsevier, 2010. 4, 6.2.1
- [Bulterman & Hardman, 2005] BULTERMAN, D. C. A.; HARDMAN, L.. **Structured multimedia authoring**. ACM Trans. Multimedia Comput. Commun. Appl., 1(1):89–109, 2005. 3.4.3
- [Cesar et al., 2008] CESAR, P.; BULTERMAN, D. C. A. ; SOARES, L. F. G.. **Introduction to special issue: Human-centered television—directions in interactive digital television research**. ACM Trans. Multimedia Comput. Commun. Appl., 4(4):1–7, 2008. 1.1
- [Cesar & Chorianopoulos, 2008] CESAR, P.; CHORIANOPOULOS, K.. **Interactivity and user participation in the television lifecycle: creating, sharing, and controlling content**. In: UXTV '08: PROCEEDING OF THE 1ST INTERNATIONAL CONFERENCE ON DESIGNING INTERACTIVE USER EXPERIENCES FOR TV AND VIDEO, p. 125–128, New York, NY, USA, 2008. ACM. 1.1
- [Chiao et al., 2006] CHIAO, H.-T.; HSU, K.-S.; CHEN, Y.-K. ; YUAN, S.-M.. **A template-based mhp authoring tool**. Computer and Information Technology, Sept. 2006. 2.1

- [Coelho & Soares, 2004] COELHO, R. M.; SOARES, L. F. G.. **Integração de ferramentas gráficas e declarativas na autoria de arquiteturas modeladas através de grafos compostos**. Master's thesis, PUC-Rio, 2004. 2.1.1, 3.4.3
- [Costa & Soares, 1996] COSTA, F. R.; SOARES, L. F. G.. **Um editor gráfico para definição e exibição do sincronismo de documentos multimídia/hipermídia**. Master's thesis, Aug. 1996. 3.4.3
- [Costa & Soares, 2007] SOARES, L. F. G.; COSTA, R. M. R.. **Modelo temporal hipermídia para suporte a apresentações em ambientes interativos**. Webmedia, 2007. 3.4.4, 5.2.2
- [Cypher, 1993] CYPHER, A.. **Watch what I do: programming by demonstration**. MIT Press, 1993. 3.3
- [De Souza, 2005] DE SOUZA, C. S.. **The semiotic engineering of human-computer interaction**. The MIT Press, 2005. 3.1
- [ETSI, 2003] **Multimedia home platform (mhp) specification 1.0.3**. Technical report, ETSI Standard, Dec. 2003. 2.1
- [Gamma et al., 1995] GAMMA, E.; HELM, R.; JOHNSON, R. ; VLISSIDES, J.. **Design Patterns**, volumen 1. Addison-Wesley Pub Co, 1st edition, 1995. 5.2.1, 5.2.1, 5.2.1, 5.2.2, 5.2.2, 5.2.2, 5.2.2, 5.2.3
- [Guimaraes & Soares, 2007] GUIMARÃES, R. L.; SOARES, L. F. G.. **Composer: um ambiente de autoria de documentos ncl para tv digital interativa**. Master's thesis, PUC-Rio, June 2007. 2.1
- [Icareus Author, 2007] **Icareus itv suite author**. Website. Acessado em 08/06/2009, URL: <http://www.icareus.com>. 2.1
- [Kim & An, 1999] KIM, J.; AN, S. S.. **Design and implementation of imat (internet multimedia authoringtool) using a unified spatio-temporal relationship model**. Multimedia Signal Processing, 1999. 2.1
- [Kim et al., 2000] KIM, H. S.; KO, H. ; LEE, K.. **Scene graph for dynamic virtual environment: Spangraph**. THE INTERNATIONAL JOURNAL OF VIRTUAL REALITY, p. 13, 2000. 5.2.2
- [Lieberman, 2001] LIEBERMAN, H.. **Your wish is my command: programming by example**. Morgan Kaufmann Publishers, 2001. 3.3

- [Lundberg & Kornefors, 1994] KORNEFORS, R.; LUNDBERG, L.. **Hyperbook tools, an authoring tool**. Multi-Media Engineering Education Proceedings, p. 281–286, July 1994. 1.1
- [Moreno et al., 2007] MORENO, M.; RODRIGUES, R. F. ; SOARES, L. F. G.. **Ginga-ncl: The declarative environment of the brazilian digital tv system**. Journal of the Brazilian Computer Society, 12(4), 2007. 1.1
- [Moura & Soares, 2001] MOURA, M. S. A.; SOARES, L. F. G.. **Relações espaciais em documentos hipermídia**. Master's thesis, PUC-Rio, Aug. 2001. 3.4.3
- [NCL Eclipse, 2008] Ncl-eclipse. Website, 2008. Acessado em 07/10/2009, URL: <http://laws.deinf.ufma.br/~ncleclipse/>. 2.1.1, 6.2.5
- [Oliveira et al., 2008] OLIVEIRA, M. R.; FILHO, C. B. ; FER, A. F.. **itv project: an authoring tool for mhp and ginga-j based on a web environment**. In: UXTV '08: PROCEEDING OF THE 1ST INTERNATIONAL CONFERENCE ON DESIGNING INTERACTIVE USER EXPERIENCES FOR TV AND VIDEO, p. 179–182, New York, NY, USA, 2008. ACM. 2.1
- [Peterson, 1977] PETERSON, J. L.. **Petri nets**. ACM Comput. Surv., 9(3):223–252, 1977. 5.2.2
- [Rabin & Burns, 1996] RABIN, M. D.; BURNS, M. J.. **Multimedia authoring tools**. In: CHI '96: CONFERENCE COMPANION ON HUMAN FACTORS IN COMPUTING SYSTEMS, p. 380–381, New York, NY, USA, 1996. ACM. 1.1
- [Rosson & Carrol, 2002] ROSSON, M. B.; CARROLL, J. M.. **Usability engineering: scenario-based development of human-computer interaction**. Morgan Kaufmann Publishers, 2002. 3.5
- [Rousseau et al., 2005] ROUSSEAU, B.; JOUVE, W. ; BERTI-EQUILLE, L.. **Enriching multimedia content description for broadcast environments: from a unified metadata model to a new generation of authoring tool**. Multimedia, Seventh IEEE International Symposium, Dec. 2005. 2.1
- [SMIL 3.0, 2008] W3c 2008. **synchronized multimedia integration language (smil) 3.0 - w3c recommendation**. Website. Disponível em <http://www.w3.org/TR/2008/REC-SMIL3-20081201>. 2.1.3, 3

- [Saade, 2003] SAADE, D. C. M.. **Relações em Linguagens de Autoria Hipermídia: Aumentando Reuso e Expressividade**. PhD thesis, PUC-Rio, 2003. 3.4.1
- [Salles et al., 2009] LIMA, B. S.; AZEVEDO, R. G. A. ; DE SALLES SOARES NETO, C.. **Autoria de documentos multimídia baseada na identificação e preenchimento de estruturas recorrentes**. Webmedia, 2009. 1.2
- [Santos & Saade, 2009] SANTOS, J. A. F.; SAADE, D. C. M.. **Linguagem xtemplate 3.0: Facilitando a autoria de programas ncl para tv digital interativa**. Webmedia, 2009. 3.4.1
- [Snyder, 2003] SNYDER, C.. **Paper Prototyping**. Morgan Kaufmann Publishers, 2003. 3.4, 4.1
- [Soares et al., 2003] SOARES, L. F. G.; RODRIGUES, R. F. ; MUCHALUAT-SAADE, D. C.. **Modelo de contextos aninhados – versão 3.0**. Technical report, PUC-Rio, 2003. 2.1.1
- [Soares et al., 2004] SOARES, L. F. G.; COLCHER, S. ; RODRIGUES, R. F.. **Relatório tv digital: Identificação dos cenários tecnológicos de interesse**. Technical report, PUC-Rio, Rio de Janeiro, RJ, 2004. 1.1
- [Soares et al., 2008] DE SALLES SOARES NETO, C.; DE SOUZA, C. S. ; SOARES, L. F. G.. **Linguagens computacionais como interfaces: um estudo com nested context language**. In: IHC '08, p. 166–175, Porto Alegre, RS, Brazil, 2008. Sociedade Brasileira de Computação. 1.1
- [Soares & Barbosa, 2009] SOARES, L. F. G.; BARBOSA, S. D. J.. **Programando em NCL 3.0 - Desenvolvimento de aplicações para o middleware Ginga**. Elsevier, 2009. 5.2.1
- [Soares et al., 2010] DE SALLES SOARES NETO, C.; SOARES, L. F. G. ; DE SOUZA, C. S.. **Tal - linguagem para autoria de templates de documentos hipermídia**. Em processo de publicação, 2010. 3.4.1
- [Soares & Lima, 2010] SOARES, L. F. G.; LIMA, B. S.. **Um ambiente de autoria para aplicações ncl extensível, adaptável e multiplataforma**. Proposta de Dissertação, 2010. 6.2.4
- [Tutorial NCL 3.0, 2007] DE SALLES SOARES NETO, C.; SOARES, L. F. G.; RODRIGUES, R. F. ; BARBOSA, S. D. J.. **Construindo programas**

audiovisuais interativos utilizando a ncl 3.0 e a ferramenta composer. Technical report, PUC-Rio, 2007. 4.1, 5.3, 5.3, A.4

[Viljoen et al., 2003] VILJOEN, D. W.; CALITZ, A. P. ; COWLEY, N. L. O.. A 2-d mpeg-4 multimedia authoring tool. In: AFRIGRAPH '03: PROCEEDINGS OF THE 2ND INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS, VIRTUAL REALITY, VISUALISATION AND INTERACTION IN AFRICA, p. 151–160, New York, NY, USA, 2003. ACM. 2.1

[Wikipedia Middleware, 2009] **Middleware**. Website. Acessado em 23/04/2009, URL: <http://pt.wikipedia.org/wiki/Middleware>. 1

[Wikipedia Lingo, 2009] **Lingo**. Website. Acessado em 01/06/2009, URL: [http://en.wikipedia.org/wiki/Lingo_\(programming_language\)](http://en.wikipedia.org/wiki/Lingo_(programming_language)). 2.1.4

[Wikipedia Tomada (cinema), 2009] **Tomada (cinema)**. Website. Acessado em 16/10/2009, URL: [http://pt.wikipedia.org/wiki/Tomada_\(cinema\)](http://pt.wikipedia.org/wiki/Tomada_(cinema)). 6

A

Material da Avaliação Formativa

Todo o material da avaliação formativa realizada é apresentado neste capítulo.

A.1

Cenário

Marcos é um chefe de família de 30 anos e trabalha como gerente em uma importante produtora de filmes, vídeos e conteúdo multimídia do Brasil. Completou o Ensino Superior e é experiente em edição de vídeos em ferramentas como Avid e Final Cut. Entretanto, a experiência em informática não inclui programação.

Na última semana, o diretor da produtora contou a todos que agora eles trabalhariam produzindo conteúdo interativo voltado para o novo padrão de TV Digital brasileiro. Como gerente de produção de conteúdo, Marcos ficou responsável em escolher o novo *software* que ajudaria a criar o conteúdo interativo.

Após diversos estudos, a sua equipe ficou em dúvida entre duas ferramentas: o Composer e o NCLite. Coube a Marcos decidir qual das duas era a mais adequada. Como não queria ser influenciado pelos integrantes da sua equipe, Marcos decidiu analisá-las sozinho. Para isso, ele imprimiu um exemplo de utilização de cada ferramenta. Feito isso, ele os analisou e tentou entender o que estava acontecendo. Para facilitar, embaixo de cada tela dos exemplos ele escrevia o seu entendimento. Se percebesse um detalhe de uma tela somente na seguinte, voltava e corrigia o que escreveu. Com essa abordagem, Marcos conseguiu escolher a ferramenta com a melhor interface para produtores como ele.

Imaginando que você é o Marcos, qual das duas ferramentas você escolheria após analisar o exemplo? Por que a escolheu? Lembre-se de anotar embaixo de cada tela o seu entendimento do que a ferramenta está fazendo.

A.2

Termo de Consentimento

O **SERG**, Grupo de Pesquisa em Engenharia Semiótica do Departamento de Informática da Puc-Rio, desenvolve pesquisa científica na área de Interação

Humano-Computador (IHC), com o objetivo de contribuir para a crescente melhoria da experiência de usuários com artefatos computacionais.

Convidamos você a participar de um de nossos estudos. Trata-se da avaliação de um protótipo criado por integrantes do **SERG** e de um *software* desenvolvido por um outro laboratório da PUC-Rio, o **Telemídia**. O propósito de ambos é o de ajudar usuários que desejam criar aplicações interativas para o padrão brasileiro de **TV Digital (SBTVD)**.

Em nosso estudo, buscamos conhecer a interpretação dos sistemas apresentados por parte de alguns usuários para identificar oportunidades de aprimoramentos tanto no protótipo quanto no *software*.

Gostaríamos de solicitar que você preencha um questionário com 5 questões para traçar o seu perfil de usuário. Em seguida você será convidado a interagir com os sistemas durante cerca de 30 minutos, tendo por objetivo final detalhar, a partir de orientações fornecidas por nós, o seu entendimento sobre o sistema. Gravaremos sua experiência em áudio e vídeo através de uma câmera. Em nenhum momento o seu rosto será gravado a fim de manter a sua privacidade. Finalmente, ouviremos a sua opinião sobre a atividade realizada, em uma breve entrevista.

Para prosseguirmos, pedimos seu consentimento para realizarmos a avaliação descrita acima, fornecendo a você algumas informações adicionais.

1. Os dados coletados destinam-se **estritamente** a atividades de pesquisa e desenvolvimento, e somente os pesquisadores do *SERG* têm acesso à íntegra do material resultante da avaliação.
2. A divulgação dos resultados de nossa pesquisa em foros científicos e/ou pedagógicos pauta-se no **respeito à privacidade**, e o **anonimato** dos participantes é preservado em quaisquer materiais que elaboramos.
3. Sua participação na avaliação pode ser interrompida a qualquer momento, e estamos disponíveis para quaisquer esclarecimentos que você deseje, antes, durante ou mesmo depois da realização da avaliação.

De posse das informações acima, você:

Consente na realização do experimento.

Não autoriza sua realização.

Rio de Janeiro, _____

Participante

Avaliador responsável

A.3

Questionário do Perfil

1. Qual faculdade está cursando ou cursou?

2. Possui conhecimento de programação? Conhece NCL?

3. Quais ferramentas utiliza para edição de vídeo? Conhece o Composer?

A.4

Telas do estudo

Todas as telas do Composer e da NCLite utilizadas no estudo são apresentadas nesta seção. Vale lembrar que selecionamos o exemplo 9 da segunda edição do Tutorial NCL (Tutorial NCL 3.0, 2007) e o reproduzimos no Composer e na NCLite.

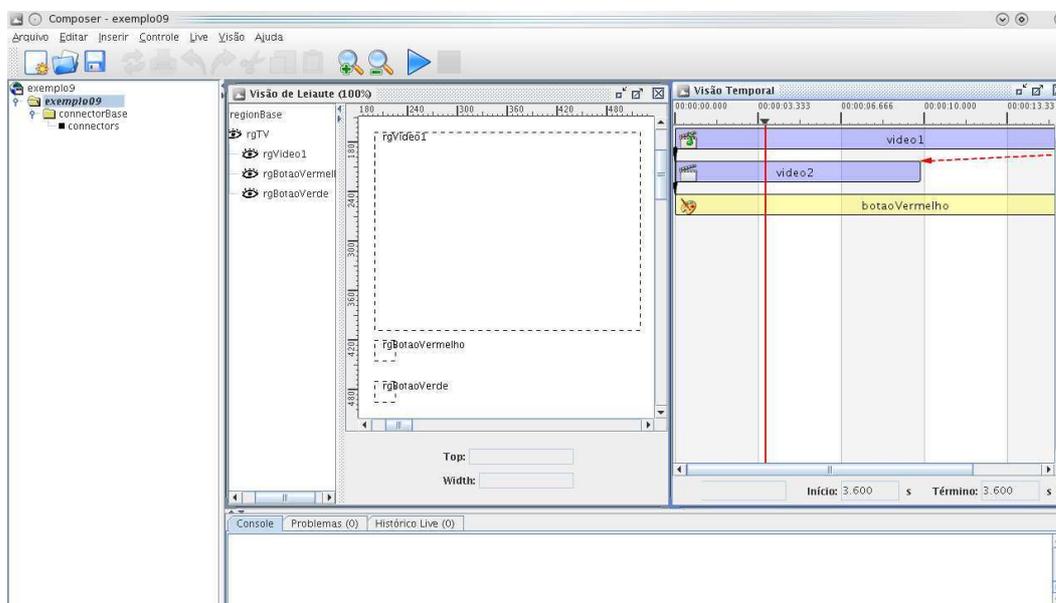


Figura A.1: Visões de Layout e Temporal

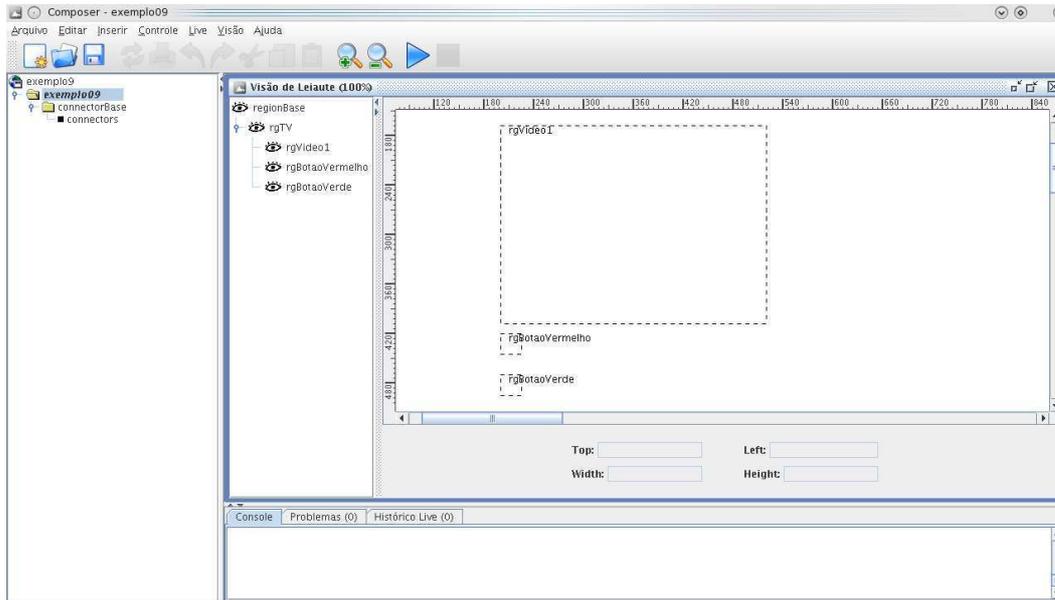


Figura A.2: Visão de Layout

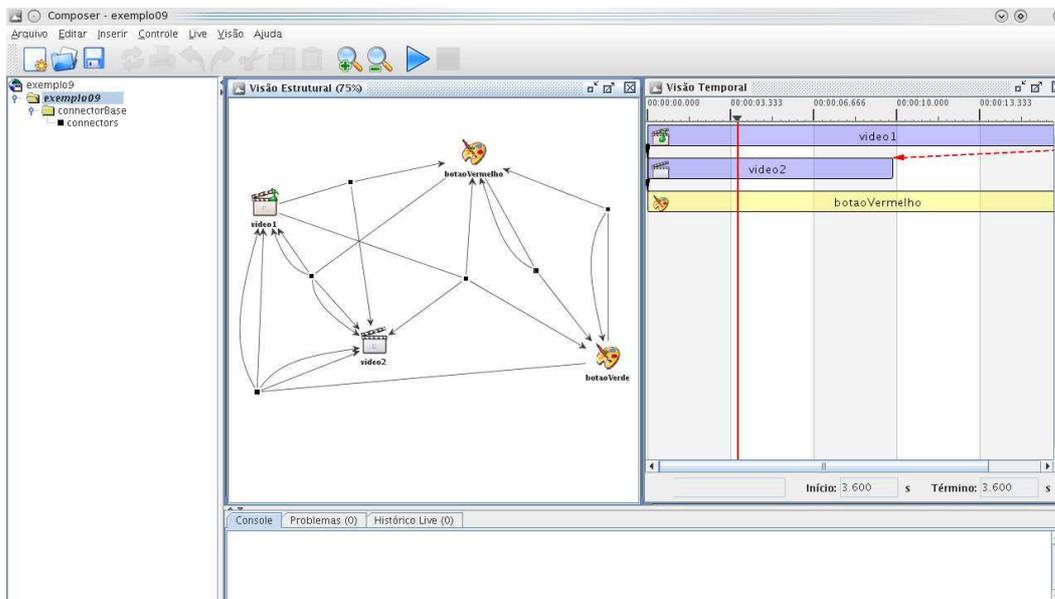


Figura A.3: Visões Estrutural e Temporal

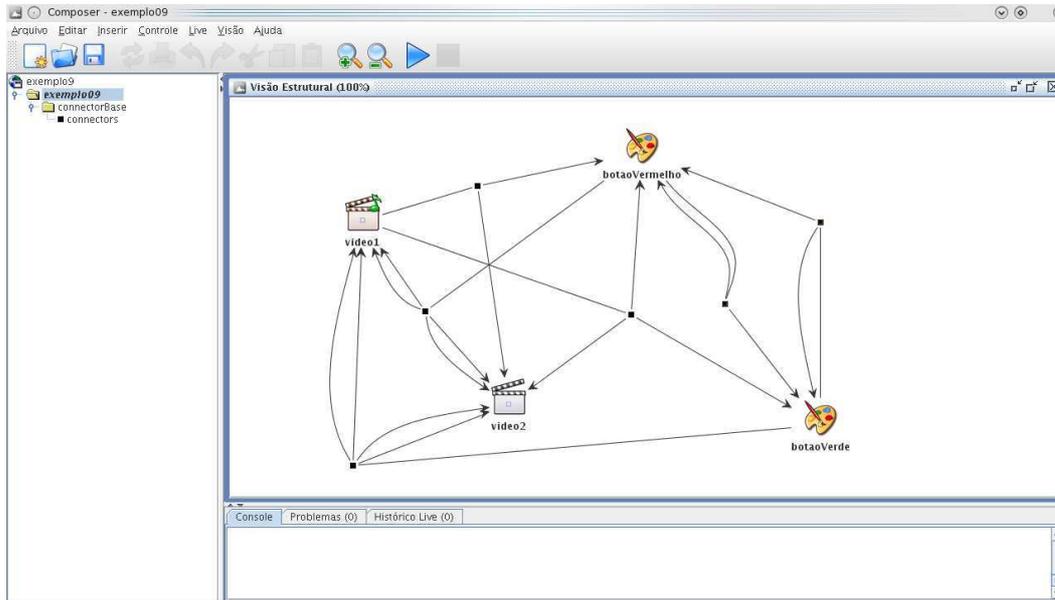


Figura A.4: Visão Estrutural

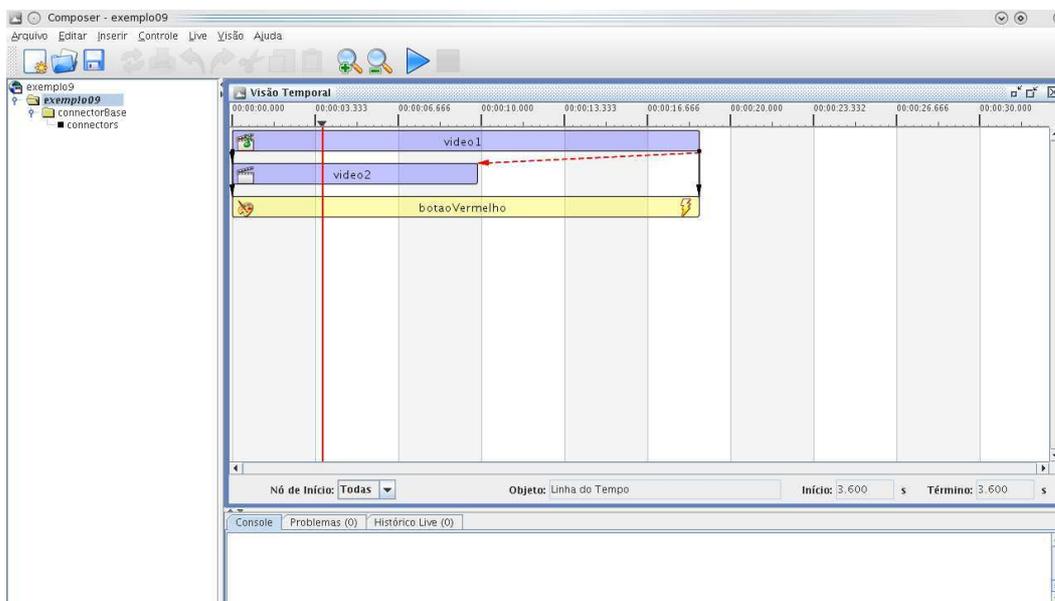


Figura A.5: Visão Temporal

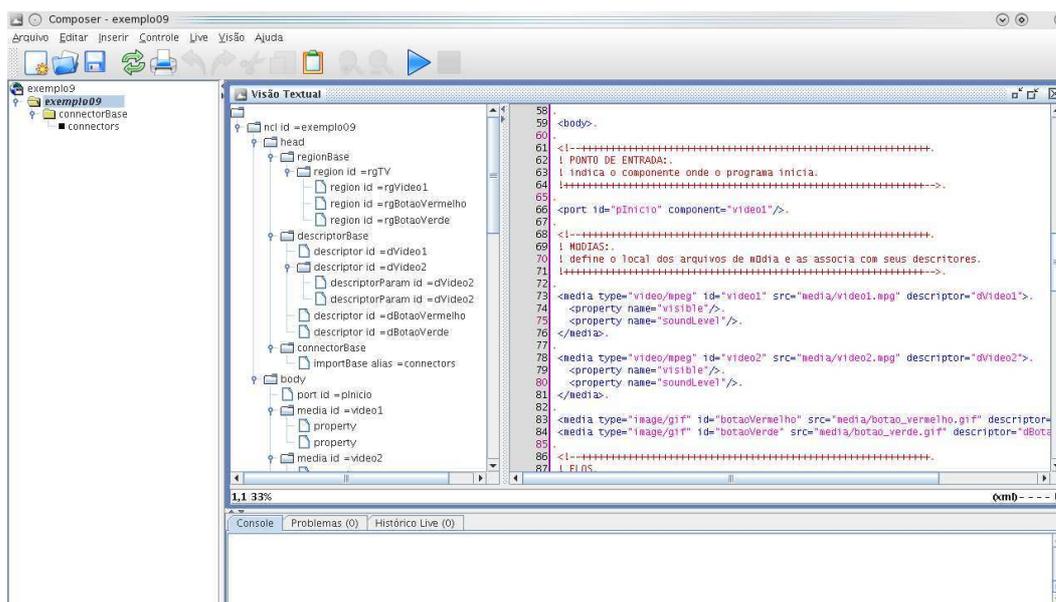


Figura A.6: Visão Textual

PUC-Rio - Certificação Digital Nº 08125662/CA

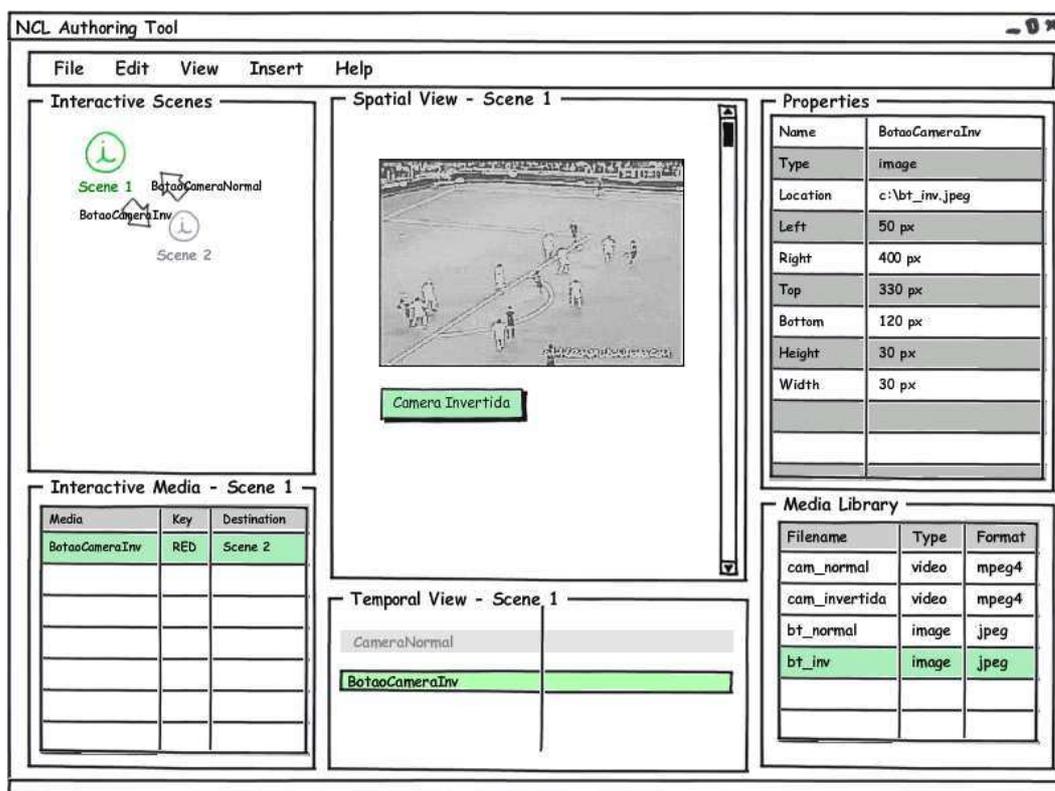


Figura A.7: Parte 1 do exemplo da NCLite



Figura A.8: Parte 2 do exemplo da NCLite

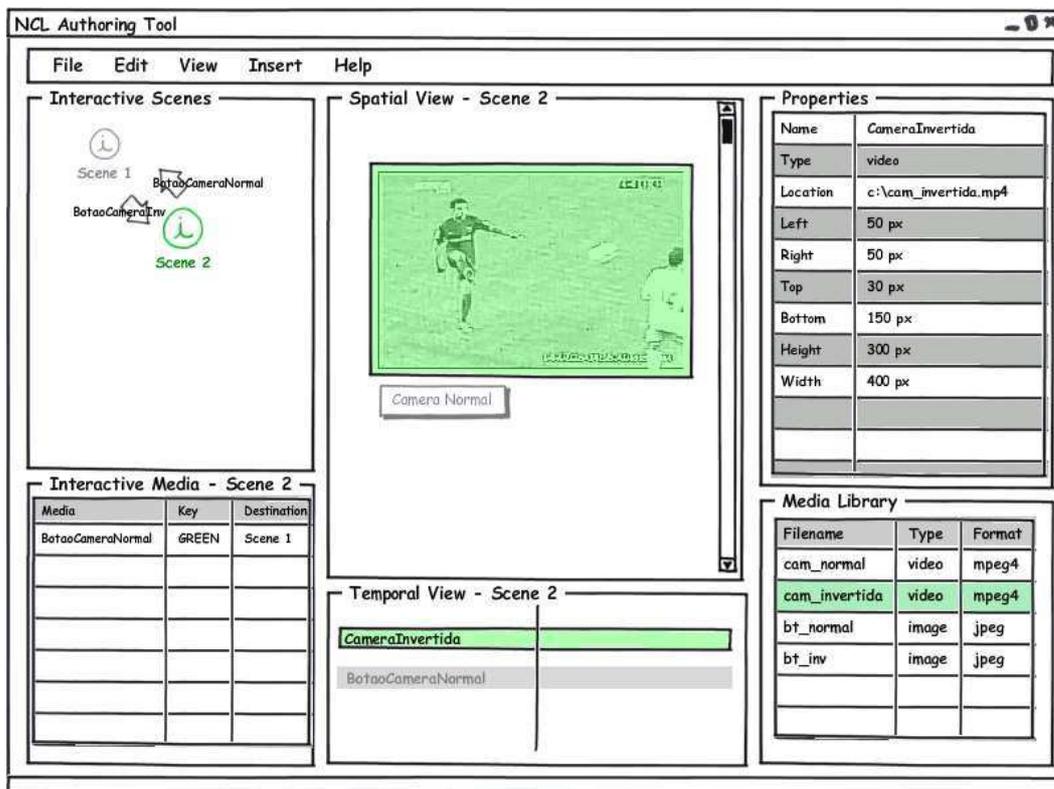


Figura A.9: Parte 3 do exemplo da NCLite

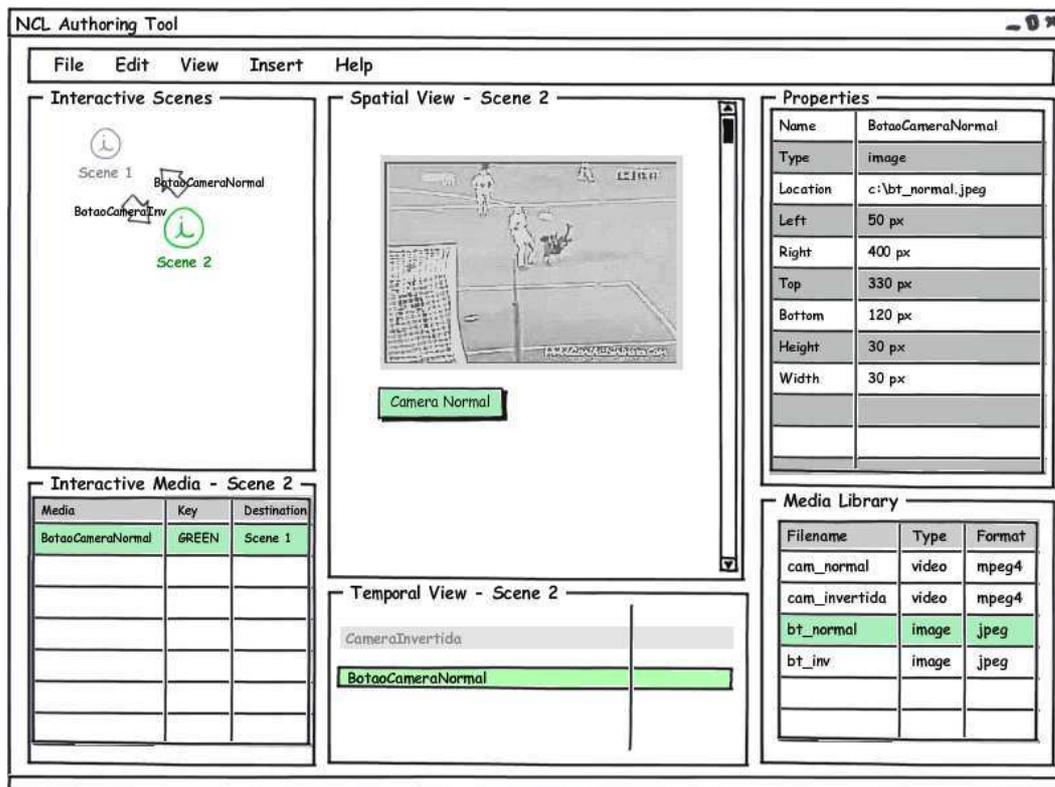


Figura A.10: Parte 4 do exemplo da NCLite

A.5 Resumo das sessões dos participantes

Nesta seção apresentamos e destacamos os pontos mais interessantes da interpretação e da entrevista com cada um dos participantes. Para uma melhor análise, eles foram separados em dois grupos, o primeiro contém os participantes que começaram a interpretação pelo Composer e o segundo os que iniciaram pela NCLite. Sendo assim, os integrantes do primeiro grupo são Alexandrita, Esmeralda e Rubi e o segundo é composto por Ágata, Diamante, Opala e Safira. Essa é a ordem em que os participantes são apresentados a seguir.

Alexandrita

Este participante começou pelo Composer. Não conseguiu escrever nada olhando somente a tela da visão estrutural e, por isso, olhou todas para tentar entender. Compreendeu que a visão estrutural era onde armazenávamos os *takes* (ou tomadas, em português), algo como uma biblioteca de *takes*, e a visão temporal era onde deveria ser feita a edição propriamente dita. Sendo assim, quando apresentamos essas duas visões em conjunto, o participante entendeu que os *takes* vídeo 1 e botão vermelho eram levados da visão estrutural para a temporal para serem editados. Ainda na visão temporal, o participante também entendeu que a seta que leva o final do vídeo 1 para o final do vídeo 2 era uma edição com

o mouse que fazia com que o *take* vídeo 1 fosse diminuído e se transformasse no vídeo 2.

A visão textual foi entendida como sendo um gerenciador de comandos e o participante também afirmou que essa visão não é nada prática. Já a visão de leiaute ele entendeu como sendo um componente diferente para visualizar os *takes* disponíveis. Isso foi percebido quando as visões de leiaute e temporal foram visualizadas em conjunto e o participante escreveu que “a visualização do programa é parecida com o *Windows Movie Maker*¹. Lado direito (visão temporal) ficam as imagens, lado esquerdo (visão de leiaute) o tamanho dos *takes*”.

A NCLite foi entendida como uma ferramenta de efeitos, algo parecido com o *After Effects* só que bem mais simples. O participante escreveu que o exemplo apresentado era um efeito de câmera invertida. A visão temporal foi corretamente compreendida.

Com a entrevista foi possível analisar melhor as dificuldades do participante. “Tive dificuldade no Composer, mas como ele parecia com *After Effects* e *Vegas Pro*, acho que entendi algo da ferramenta. Para mim a NCLite é uma ferramenta de efeitos do Composer. Para editar usaria o Composer, se quisesse inverter posição do vídeo ou aumentar/diminuir usaria a NCLite”. Inicialmente, teve dificuldade em entender a visão estrutural. Depois percebeu que deveria pegar os *takes* da estrutural e jogar para a temporal onde editaria.

Quando perguntado sobre a NCLite, afirmou que não sabia como trocava da cena 1 para a 2 e que percebeu com clareza mudanças na visão temporal.

Quanto à escolha da ferramenta, afirmou: “Escolheria o Composer pelo fato de saber como criar a timeline. Com a Composer eu tenho como editar, mas com a NCLite eu não edito. Com ela eu crio efeitos. Se fosse só para criar efeitos simples, ficaria com a NCLite”.

Esmeralda

Logo que começou, o participante abriu todas as telas do Composer para poder entender a ferramenta. Escreveu na tela que apresentava as visões estrutural e temporal (Figura A.3) que o exemplo era semelhante a uma construção de menu do DVD Studio Pro². Os conectores da estrutural seriam o mapeamento das mídias para formar o menu. O participante afirmou ainda que a visão temporal “seria uma espécie de zoom da visão estrutural. Por exemplo, a visão temporal exibida é uma redução da estrutural na conexão do vídeo 1 com o vídeo 2”. Ficou claro o objetivo da visão de leiaute, mas não o da visão textual. Considerou que essa visão era somente para visualizar e não editar.

¹<http://www.moviemakerpreview.com/>

²<http://www.apple.com/finalcutstudio/dvdstudiopro/>

Assim como no Composer, o participante precisou abrir todas as telas da NCLite para escrever o seu entendimento. Os componentes Media Library, Properties e Spatial View foram compreendidos sem problemas. Entretanto, não entendeu os componentes cenas interativas e mídias interativas. Continuou com a ideia de menu de DVD e achou que as cenas seriam o mapeamento das mídias para formar o menu.

Considerou ainda que as mídias presentes no Media Library com formato MPEG4 seriam os vídeos visualizados na execução e as com extensão JPEG seriam imagens que exemplificariam de forma estática esses vídeos. Com essa afirmação, observamos que o participante mudou o seu entendimento do componente.

Na entrevista, algumas questões percebidas na interpretação ficaram mais claras. O participante novamente descreve o entendimento de que são “ferramentas para criar menu. No caso da NCLite, teria menos opção de criação. O Composer tem várias formas de visualizar a mesma coisa, você tem vários caminhos para chegar até o mesmo objetivo. Ele dá um pouco de medo, mas parece ser mais exato de trabalhar”. Além disso, afirmou que analisou mais a estrutura das ferramentas e não o conteúdo específico de cada componente gráfico.

Considerando somente o Composer, também falou que achou “a visão textual e estrutural muito similares, só alterando a forma que apresentam as informações”. Também reafirma o que escreveu sobre a visão temporal ser um zoom da estrutural. Já sobre a NCLite, a considerou bem simples e que o destaque em verde (a indicação de seleção do elemento) representaria a execução do vídeo.

Caso precisasse escolher alguma ferramenta, escolheria o Composer porque tem várias maneiras de chegar ao mesmo objetivo e, dessa forma, é possível usar a mais adequada ao seu método de trabalho.

Rubi

Iniciando pelo Composer, o participante interpretou a visão estrutural como sendo um fluxograma do projeto sendo editado. Ele entendeu também que as setas dessa visão apontavam para o elemento que acionou a execução do elemento presente na origem da seta. Ainda na estrutural, ele entendeu, pela nomenclatura, que as imagens funcionavam como botões.

Passando para a visão temporal, o participante só não compreendeu a seta pontilhada vermelha e as setas iniciais passaram despercebidas. Considerou a visão textual como equivalente à temporal só que com edição direta no código fonte. Já a visão de leiaute foi corretamente entendida.

Na NCLite compreendeu corretamente os componentes Temporal View, Spatial View, Properties e Media Library. O conceito de cenas parece que não foi entendido. Percebemos isso por ele afirmar que as telas um, dois e três

(respectivamente, Figuras A.7, A.8 e A.9) tratam da mesma cena mas que a terceira está em outro momento do tempo com outro vídeo e outro botão. Só para lembrar, a cena da NCLite exibida muda na terceira tela.

Na entrevista confirmamos algumas observações da interpretação. Perguntado sobre a dificuldade das ferramentas, primeiro falou do Composer: “Tive muita dificuldade em entender a visão estrutural no Composer. Ele parece ser mais confuso do que a NCLite”. Sobre a NCLite afirmou ainda: “A NCLite me pareceu ser melhor e mais intuitiva. Mas não consegui saber até quando seria a cena 1 e o momento que mudava para a cena 2”.

Sobre a escolha de uma ferramenta, ficou com a que considerou mais fácil de usar. “Escolheria a NCLite por ser mais simples. O Composer pareceu ser mais completo, mas também confuso”.

Ágata

Iniciando pela NCLite, o participante conseguiu logo na primeira tela (Figura A.7) interpretar corretamente os componentes Spatial View, Temporal View, Properties e Media Library. Já os componentes Interactive Scenes e Interactive Media só entendeu quando fez uma comparação entre a segunda e a terceira telas (Figuras A.8 e A.9). O participante não compreendeu se as propriedades referentes ao tamanho do vídeo correspondiam ao tamanho real do vídeo exibido na visão espacial ou ao tamanho estabelecido para a aplicação final. Ficou em dúvida se a visão temporal funcionava em camadas como as ferramentas que utiliza. As camadas fazem com que as mídias das camadas superiores fiquem à frente das mídias das inferiores na visão espacial.

Passando para o Composer, começou analisando a visão estrutural e entendeu que existia uma interação entre os elementos apresentados, mas não sabia dizer ao certo como se daria essa interação. Na visão temporal interpretou que ao fim do vídeo 1 deveria apertar o botão vermelho para mudar para o vídeo 2. Não soube ao certo “se as setas anteriores aos vídeos também permitiriam acesso ao botão vermelho para ver o vídeo 2”. Interpretou corretamente a visão de leiaute, só sentiu falta do bottom e right. Quanto à visão textual, escreveu: “Isso é de assustar!”. Mesmo assim achou fácil entender os atributos dos elementos media e property. Preferiu a tela que apresentava as visões estrutural e temporal em paralelo por parecer que tinha maior controle. Quando observou as visões de leiaute e temporal em paralelo ficou na dúvida como deveria inserir elementos na temporal.

Quanto à entrevista, começamos perguntando o que achou da experiência. O participante afirmou que “é estranho olhar tela no papel. Você tem vontade de testar/fuçar para ver se deu certo ou errado”. Ainda assim achou interessante porque “quando você lê a tela, você presta atenção em algo que passaria desper-

cebido e te levaria para um caminho errado”. Durante a observação, percebemos que o participante abriu todas as telas para entender o processo antes de chegar a uma conclusão. “Se olhar somente a primeira tela pode paralisar e desistir”. Ele afirmou ainda que no Composer sentiu maior necessidade de olhar todas as telas para poder entender.

Caso precisasse escolher alguma ferramenta, como sugerido no cenário, escolheria a NCLite porque está mais próxima das ferramentas conhecidas. “Tenho mais controle de tudo. Quando você me apresentou a primeira tela do Composer, tinha somente a visão estrutural. Eu não tinha mais nada ali. Isso me assustou. A NCLite eu de cara já conseguia identificar elementos parecidos com outras ferramentas de edição. Como a mídia vai aparecer na tela, as informações de cada elemento, quais os botões de interatividade”.

Ainda sobre o Composer, o participante afirmou que precisaria testá-lo em um computador para saber se é complicado como aparenta ser. Considerou também que a “visão textual é assustadora para não programadores”. Além disso, sentiu falta da visão espacial e também de uma maior relação entre as visões. Afirmou ainda que não conseguiu descobrir o objetivo dos relacionamentos da visão estrutural, além de considerar essa visão confusa.

Quanto ao exemplo do Composer e do NCLite, os achou parecidos porque ambos trabalham com vídeos e botões e existe uma estrutura de ir e vir onde os botões levam a vídeos diferentes.

Diamante

Ao iniciar a interpretação, o participante escreveu direto nas primeiras telas da NCLite. Quando chegou na terceira (Figura A.9), ficou em dúvida, olhou a seguinte e voltou a analisar as telas anteriores. Continuou a descrever as telas até chegar na última (Figura A.10). Nesse momento, abriu todas as telas novamente e passou a escrever todo o seu entendimento de novo.

Analisando as duas partes do entendimento do participante, percebemos que na primeira ele se fixou nos componentes Interactive Scenes, Interactive Media e Properties. Entretanto, não compreendeu completamente como as cenas se relacionavam. Na segunda parte, conseguiu entender o funcionamento das cenas e do componente Media Library, mas não percebeu que a visão temporal era uma linha do tempo. Considerou que este componente somente exibia os elementos presentes na cena em questão.

Dando continuidade ao estudo, o participante analisou o Composer. Começou pela visão temporal e teve dúvidas em descrever o que essa visão estava fazendo. “Visão temporal: eu tenho ou dois vídeos mais um botão ou vídeo 2 sendo chamado do vídeo 1. Parece haver algum tipo de ação associada ao botão”. Interpretou ainda

que o ícone usado pelo Composer para indicar que o vídeo 1 é a mídia inicial a ser executada durante a execução representava a existência de áudio no vídeo 1 e a ausência no vídeo 2.

Na visão estrutural escreveu: “Não faço ideia do que representam os conectores. A visão estrutural não faz sentido”. Quando analisou as visões de leiaute e temporal juntas considerou que elas não possuíam os mesmos elementos.

Quanto à entrevista, começamos perguntando o que achou da experiência devido à dificuldade encontrada na NCLite. O participante afirmou que “é mais fácil quando vc está mexendo no programa propriamente dito do que pelas telas”. Passamos para o Composer e o participante nos contou que “ela lembra um pouco a interface do Flash, mas várias coisas não fazem sentido na maneira como é mostrada. Principalmente a visão estrutural, para mim não faz o menor sentido. Os conectores deixam quem está acrescentando elementos completamente perdido. Eu vejo que existe uma conexão, que diz algo para o programador, mas para mim não diz nada”.

Em relação à NCLite, disse que é “muito mais fácil mexer nela do que no Composer”. Explicou ainda o seu entendimento sobre as cenas, mas fez uma ressalva: “Na NCLite não vejo como criar/inserir novos elementos. Só vejo informações da cena montada. Não vejo como montar, como colocar aquele botão ali, como inserir a interatividade. Parece um programa pronto, onde você vê a estrutura de uma aplicação criada”.

Perguntado sobre qual das duas ferramentas escolheria, afirmou que usaria as duas em conjunto porque o Composer não exhibe um retorno do que está sendo criado, mas possibilita criar/inserir conteúdo. Já a NCLite é mais fácil de usar e muito boa para visualizar mas parece não oferecer a possibilidade de criar.

Opala

Analisando a NCLite, o participante conseguiu entender corretamente os componentes Spatial View, Temporal View e Properties. Entretanto, os componentes referentes às cenas não foram compreendidos. O participante interpretou que as imagens (botões) que alteravam as cenas eram parte da interface gráfica e não do conteúdo interativo sendo criado. Percebemos isso pela afirmação: “A ideia das visões de câmera é boa, mas poderia ser mais objetiva”. Além disso, verificamos que ele levou em consideração a parte visual da interface.

No Composer, o participante analisou mais a parte visual das diferentes visões. Quanto ao entendimento, a visão de leiaute considerou como sendo um painel para criação de botões e a visão temporal conseguiu interpretar como uma linha do tempo mas não entendeu os relacionamentos que as setas proporcionavam. Percebeu que a visão textual servia para edição do código fonte. Finalmente,

considerou a visão estrutural confusa e não conseguiu entender os relacionamentos presentes.

Pela entrevista percebemos que o participante conhece uma gama considerável de ferramentas. Ele achou tanto o Composer quanto a NCLite parecidos com Flash, After Effects, Final Cut e Adobe Premiere. Confirmamos que o participante ficou muito preso ao visual gráfico das ferramentas, ele afirmou que não gostou do visual da NCLite e justificou: “Acho que o visual deixa a pessoa mais confortável para trabalhar. Ela pode achar que a interface está mal feita se tiver um visual ruim”³.

Quanto às cenas da NCLite, ele não conseguiu descobrir como acontecia o seu funcionamento. “As cenas ficaram confusas, não entendi se podia interagir com as duas ao mesmo tempo. Se clicasse em uma ou outra iria para aquela cena. Isso não ficou claro”. Interpretou ainda que os elementos da media library eram visões de câmera e que através deles seria possível escolher a câmera da esquerda, do topo ou alguma outra posição. Dessa forma, percebemos que interpretou os botões de interatividade que alteravam as cenas como componentes da interface gráfica para a escolha da câmera.

Perguntado sobre qual das duas ferramentas escolheria, afirmou: “Ficaria na dúvida se tivesse que escolher. A NCLite me apresenta mais informação para desenvolver o projeto, mas o Composer apresenta a linguagem que pode facilitar para quem entende de programação. Só utilizando no computador para saber”.

Safira

Assim que olhou a primeira tela (Figura A.7) da NCLite começou a escrever, mas sempre que passava para uma seguinte retornava à anterior e as comparava. Percebeu semelhança com o *After Effects* em relação à disposição das informações na interface gráfica. Entendeu corretamente os componentes Spatial View, Properties e Media Library logo na primeira tela. Já os componentes Interactive Scenes e Interactive Media só foram entendidos na terceira tela (Figura A.9) após comparações com as anteriores.

Por fim, percebemos também que o participante não entendeu o componente Temporal View como uma linha do tempo. As primeiras linhas escritas na análise do Composer nos confirmam essa percepção: “Reparei logo na existência de uma timeline, que se assemelha ao After Effects, mas que não havia no outro teste”.

No Composer, o participante percebeu a linha do tempo da visão temporal facilmente, mas na visão estrutural não escreveu nada sobre os elementos presentes

³O avaliador responsável pelo estudo explicou posteriormente que o objetivo da NCLite era justamente transmitir a imagem de uma interface gráfica não finalizada e desenhada a mão.

nela. Pareceu que ele não conseguiu compreender o conteúdo dessa visão. Entendeu de maneira correta a visão textual. Por fim, na visão de leiaute escreveu que o vídeo 1 parecia habilitado e que não entendeu o propósito dos botões verde e vermelho.

Na entrevista, algumas questões percebidas na interpretação ficaram mais claras. Em relação ao ir e vir entre as telas do NCLite, o participante afirmou: “Voltei para analisar a dinâmica da seleção e onde leva cada comando. São quatro telas, cada hora com um botão ou câmera selecionado, eu voltei para entender a lógica”. Falou também que não sentiu dificuldade em entender essa ferramenta.

Perguntado sobre o Composer, disse que o achou “mais complexo. Se assemelha mais ao Flash, com uma parte só para programação”. Conseguiu entender alguns componentes, mas afirmou que “o problema foi a visão estrutural e a de leiaute. Não cheguei a um propósito para essas visões e não entendi a relação entre elas. Voltei às telas para tentar fazer uma relação, mas não consegui”.

Caso precisasse escolher, disse que “escolheria a NCLite porque achei mais fácil de usar e entendi o propósito dos componentes”.

B

Estendendo a implementação da NCLite

Neste capítulo detalhamos como deve ser feita a recuperação e a montagem do ambiente para estender a implementação da NCLite.

B.1

Recuperando o código fonte

Até o momento da finalização desta dissertação nenhuma versão estável da NCLite foi liberada. Porém uma versão usável pode ser baixada diretamente da linha principal de desenvolvimento. As informações aqui demonstradas tanto servem para sistemas operacionais Windows quanto para Unix.

Para obter o código fonte da NCLite, são necessários:

1. Cliente SVN de linha de comando (pode ser outro equivalente, como o Tortoise, porém será demonstrado aqui usando apenas linha de comando). Disponível em <http://subversion.tigris.org>;
2. Usuário cadastrado nos repositórios. Vale ressaltar que a localização do repositório pode vir a ser alterado por questões técnicas. Caso não consiga recuperar o código fonte através do passo-a-passo apresentado, entre em contato com os autores;
3. Conexão com a internet disponível.

A NCLite depende do componente NCL-Parser, que foi desenvolvido em conjunto com outro aluno de mestrado (Gustavo Honorato), que também não possui uma versão estável. Portanto, antes de recuperar o código da NCLite, precisaremos fazer o mesmo para o NCL-Parser. Crie um diretório para armazenar as cópias de trabalho do código fonte da NCLite e do NCL-Parser. Iremos chamar essa pasta de workspace. Na linha de comando, entre dentro da pasta criada e chame o comando: **svn checkout <http://ncl-parser.googlecode.com/svn/trunk/ncl-parser>**. No final da execução do comando é criada uma pasta chamada `ncl-parser` com o código fonte completo do NCL-Parser.

Agora devemos recuperar o código fonte da NCLite. Ainda no diretório workspace execute o seguinte comando: **svn checkout <https://ideias.inf.puc-rio.br:8443/svn/htrannin/workspace/nclite/trunk/source> nclite**. A pasta

nclite é criada com todos os projetos da ferramenta. Atualmente, esses projetos são os seguintes:

- br.pucrio.inf.serg.nclite
- br.pucrio.inf.serg.nclite.auxiliary-components
- br.pucrio.inf.serg.nclite.edit
- br.pucrio.inf.serg.nclite.gui-rcp
- br.pucrio.inf.serg.nclite.persistence
- br.pucrio.inf.serg.nclite.scenes
- br.pucrio.inf.serg.nclite.spatial
- br.pucrio.inf.serg.nclite.timeline

A NCLite depende ainda das bibliotecas *Jaret Timebars v1.41* e *Xuggler*. Para realizar o *download* da *Jaret Timebars* é preciso ir à página <http://www.jaret.de/timebars/index.html> e clicar no *link* *timebars-1.41-bin.zip*. O componente *Timebars* depende do componente *Utils v0.32*, por isso precisamos ir à página <http://www.jaret.de/jaretutil/index.html> e fazer o *download* do arquivo *jaretutil-0.32-bin.zip*. Terminada essa parte, precisamos agora descompactar os arquivos baixados e copiar os arquivos *.jar* para o diretório *lib/jaret* dentro da raiz do projeto *br.pucrio.inf.serg.nclite*. Caso esse diretório não exista, é preciso criá-lo.

Finalmente, precisamos recuperar a biblioteca *Xuggler v3.4*. Vá até o *site* <http://www.xuggler.com/xuggler/downloads/> e faça o *download* da versão correspondente ao seu sistema operacional. Após a recuperação do arquivo, é preciso instalar a biblioteca. Na própria página onde foi feito o *download* temos uma explicação de como instalá-la. Além disso, na página <http://blog.xuggler.com/2009/01/23/how-to-install-xuggler-on-microsoft-windows/> encontramos um exemplo com vídeos da instalação no *Windows*. Terminada a instalação, copie os arquivos *.jar* da pasta *DIRETORIO_DA_INSTALACAO/xuggler/share/java/jars/* para o diretório *lib/xuggler* dentro da raiz do projeto *br.pucrio.inf.serg.nclite*. Novamente, se esse diretório não existir, é preciso criá-lo.

B.2

Preparando o ambiente

Para continuar o desenvolvimento da NCLite é preciso configurar o ambiente completo do *Eclipse-RCP*. Para isso precisamos dos seguintes componentes:

1. Eclipse for RCP/Plug-in Developers 3.5 (Galileo)
2. subclipse (plugin do Subversion para o Eclipse)
3. m2eclipse (plugin do Maven para o Eclipse)
4. *Graphical Editing Framework* (GEF) plugin
5. *Eclipse Modeling Framework* (EMF) plugin

Para realizar o *download* do *Eclipse* vá ao site <http://www.eclipse.org/downloads/> e selecione a versão do *Eclipse* de acordo com a Figura B.1. Terminado o *download*, descompacte o arquivo. Isso resultará na criação da pasta *eclipse*. Dentro desta pasta, execute a aplicação através do arquivo *eclipse.exe* no *Windows* ou *eclipse* no *GNU/Linux*.

The screenshot shows the Eclipse Downloads page with the following content:

- Navigation: Home, Downloads, Users, Members, Committers, Resources, Projects, About Us
- Search: Google Custom Search
- Section: Eclipse Downloads
- Sub-sections: Eclipse Packages, Projects, Development Builds
- Galileo Packages (based on Eclipse 3.5 SR2) - Compare Packages
- Looking for Older Versions or Source Code?
- Table of packages:

Package Name	Size	Downloads	Operating Systems
Eclipse IDE for Java EE Developers	189 MB	1,577,343	Windows 32bit, Mac Carbon 32bit, Mac Cocoa 32bit 64bit, Linux 32bit 64bit
Eclipse IDE for Java Developers	92 MB	712,409	Windows 32bit, Mac Carbon 32bit, Mac Cocoa 32bit 64bit, Linux 32bit 64bit
Eclipse IDE for C/C++ Developers	79 MB	314,986	Windows 32bit, Mac Carbon 32bit, Mac Cocoa 32bit 64bit, Linux 32bit 64bit
Eclipse for PHP Developers	138 MB	265,313	Windows 32bit, Mac Carbon 32bit, Mac Cocoa 32bit 64bit, Linux 32bit 64bit
Eclipse IDE for Java and Report Developers	220 MB	58,006	Windows 32bit, Mac Carbon 32bit, Mac Cocoa 32bit 64bit, Linux 32bit 64bit
Eclipse Modeling Tools (includes Incubating components)	367 MB	50,414	Windows 32bit, Mac Carbon 32bit, Mac Cocoa 32bit 64bit, Linux 32bit 64bit
Eclipse for RCP/Plug-in Developers	183 MB	46,634	Windows 32bit, Mac Carbon 32bit, Mac Cocoa 32bit 64bit, Linux 32bit 64bit
Pulsar for Mobile Java Developers	112 MB	36,248	Windows 32bit, Mac Carbon 32bit, Mac Cocoa 32bit 64bit, Linux 32bit 64bit
Eclipse SOA Platform for Java and SOA Developers	137 MB		Windows 32bit, Mac Carbon 32bit
- Right sidebar: Yoxos SecureSource, EclipseSource, Eclipse Training Series (April 12 - May 28 RCP • OSGi • Modeling), Popular projects (May 31 0)

Figura B.1: Página de download do Eclipse-RCP

Ao iniciar, o Eclipse pedirá para selecionar uma pasta para armazenar seu *workspace*. Selecione a pasta onde recuperamos o código fonte da NCLite e do NCL-Parser e clique em OK.

O próximo passo, será a instalação do plugin do Eclipse para integração com o SVN. Para prosseguir, clique na opção de menu

Help > Install new software... Irá aparecer a janela apresentada na Figura B.2 e então clique no botão Add. Na nova janela digite no campo Location a URL http://subclipse.tigris.org/update_1.6.x e clique em OK.

Selecione todos os plugins, clique em Next. Prossiga pelo assistente e aceite os termos da licença e clique em Finish. É possível que o *Eclipse* apresente uma mensagem pedindo a confirmação para instalar um conteúdo não-assinado. Se isto ocorrer, permita que o conteúdo seja instalado. Será solicitado o reinício do *Eclipse* quando a instalação terminar.

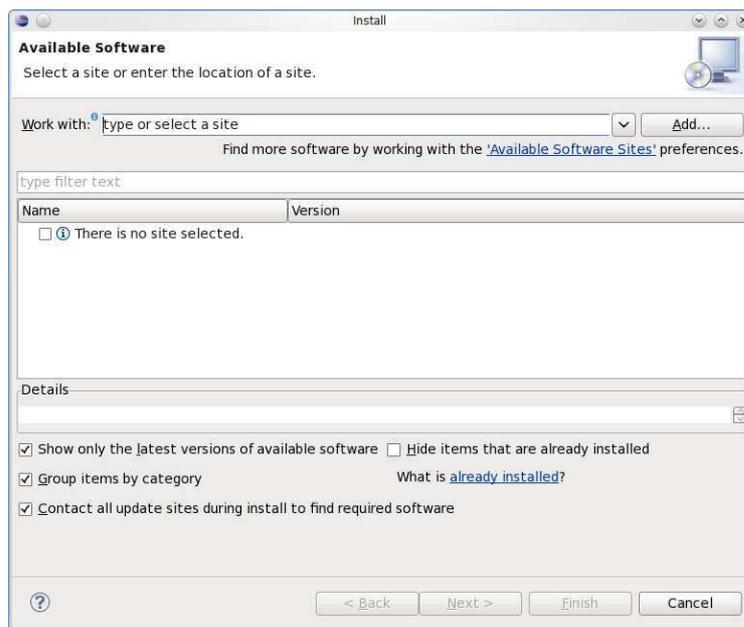


Figura B.2: Janela de instalação de plugins do Eclipse

A seguir instalaremos o plugin do Eclipse para integração com o Maven. Clique novamente na opção Help > Install new software... para aparecer a janela da Figura B.2 outra vez. Clique no botão Add, digite no campo Location da janela aberta a URL <http://m2eclipse.sonatype.org/update/> e clique em OK.

Selecione todos os plugins e clique em Next. Prossiga pelo assistente e aceite os termos da licença e clique em Finish. Novamente, permita a instalação de conteúdo não-assinado e reinicie o *Eclipse* após a instalação.

Continuando, é preciso instalar também o *plugin* do GEF. Vá outra vez para a janela de instalação de novos plugins e clique no botão Add. Agora insira no campo Location a URL <http://download.eclipse.org/tools/gef/updates/releases/> e clique em OK. Selecione todos os *plugins* de *GEF SDK 3.5.2* e prossiga pelo assistente como descrito acima.

Finalmente, a última instalação é o *plugin* do EMF. Vá para a janela de instalação de novos *plugins* de novo e clique no botão Add. Agora insira no campo Location a URL <http://download.eclipse.org/modeling/emf/updates/releases/> e clique em OK. Selecione todos os *plugins* da opção *EMF SDK 2.5.0 (EMF + XSD)* e prossiga pelo assistente como descrito anteriormente.

B.3 Configurando os projetos

Terminada a preparação do ambiente, precisamos agora configurar os projetos no *Eclipse*. Para isso, clique na opção de menu File > Import. Na janela que abrir, escolha a opção Existing Projects into Workspace como exibido na Figura B.3 e clique no botão Next. Na tela seguinte, clique em Browse da opção Select root directory e selecione a raiz do *workspace*. Todos os projetos descritos acima estarão marcados, incluindo o NCL-Parser. Clique em Finish para terminar.

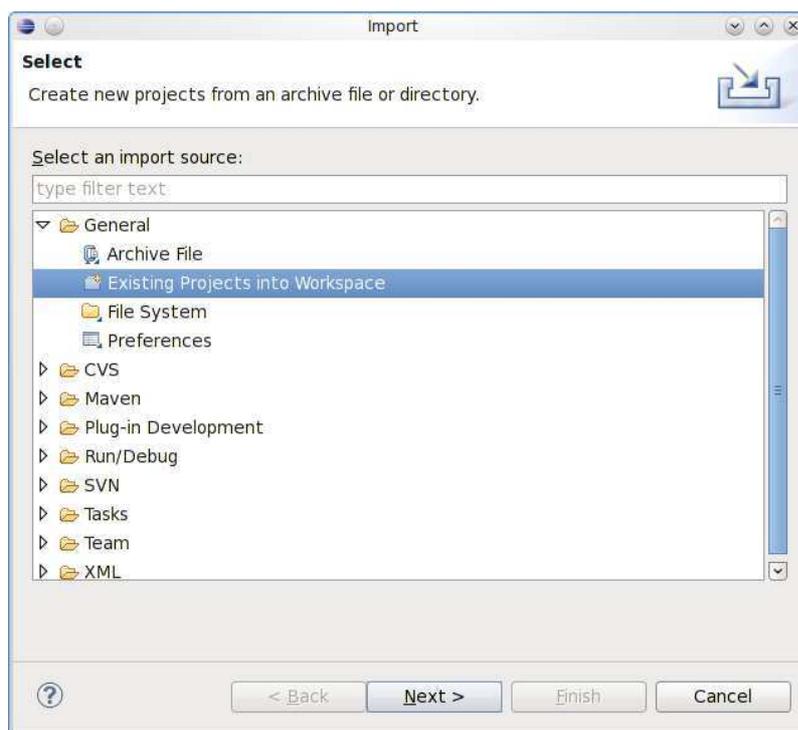


Figura B.3: Janela para importação de projetos existentes

Os projetos irão aparecer com alguns erros porque falta recuperar algumas bibliotecas. Essa recuperação é feita pelo *plugin* do *Maven* que instalamos anteriormente. Clique com o botão direito do mouse no projeto *ncl-parser* e escolha a opção Run as > Maven install. Espere um tempo para poder baixar as dependências e compilar o projeto. Quando terminar, faça o mesmo para o projeto

`br.pucRio.inf.serg.nclite`. Ao final de tudo, não teremos mais nenhum erro de compilação e já será possível estender a implementação da NCLite.

É importante lembrar que o projeto a ser executado é o `br.pucRio.inf.serg.nclite.gui-rcp` porque, como detalhado no Capítulo 5, este projeto é o núcleo que agrupa todos os módulos e gera uma aplicação separada do *Eclipse*.