

Referências Bibliográficas

ANNETT, J.; & DUNCAN, K. D. **Task analysis and training design**, In: Occupational Psychology, 1967. v. 14, p. 211 – 21.

BARBOSA, S. D. J. **Programação via Interface**. 1999. Tese de doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil. Maio de 1999.

BARBOSA, S. D. J. **Expanding Software through Metaphors and Metonymies**. In: Proceedings of 5th International Conference on Intelligent User Interfaces, 2000, New Orleans, USA, 9 – 12 Janeiro. p. 13 – 20.

BECKER, V.; FORNARI, A.; HERWEG F., GÜNTHER H.; MONTEZ, C. **Recomendações de Usabilidade para TV Digital Interativa**. In: II WTVD, 2006, Curitiba. Anais do WTVD 2006 - Workshop de TV Digital. p. 27 – 38.

BLACKWELL, A. F. Psychological issues in end-user programming. In: **End User Development**. New York: Springer, 2006. p. 9 - 30.

BURNETT, M.; ATWOOD, J.; DJANG, R.; GOTTFRIED, H.; RECHWEIN, J.; YANG, S. Forms/3: A first-order visual language to explore the boundaries of the spreadsheet paradigm. In: Journal of Functional Programming, 2001. v. 11, p. 155 - 206.

BURNETT, M.; ROTHERMEL, G.; COOK, C. An Integrated Software Engineering Approach for End-User Programmers. In: **End User Development**. New York: Springer, 2006. p. 87 – 113.

CARROL, J. M. **Introduction to the special issue on “Scenario-Based Systems Development.”** In: Interacting with Computers, 2000. v. 13(1), p. 41 - 42.

COBLENZ, M. J.; KO, A. J.; MYERS, B. A. **JASPER: An Eclipse Plug-In to Facilitate Software Maintenance Tasks**. In: Proceedings of the 2006 OOPSLA workshop on eclipse technology eXchange, Portland, Oregon, 2006. p. 65 – 69.

COELHO, R. M.; SAORES, L. F. G., **Integração de Ferramentas Gráficas e Declarativas na Autoria de Arquiteturas Modeladas através de Grafos Compostos**. 2004. Dissertação de mestrado -- Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil. 2004.

COSTABILE, M. F.; FOGLI, D.; LETONDAL, C.; MUSSIO, P.; PICCINNO, A. **Domain-Expert Users and their Needs of Software Development**. In: Special Session on EUD, UAHCI Conference, Crete, Junho de 2003. p. 232 - 236.

COSTABILE M.; FOGLI D.; LANZILOTTI R.; MUSSIO P.; PICCINNO A. **End-User Development Supporting Work Practice**. In: Journal of Organizational and End User Computing, 2006. v. 18, n. 4, p. 43 – 65.

CUNHA, C. K. V. **Um Modelo Semiótico dos Processos de Comunicação Relacionados à Atividade de Extensão à Aplicação por Usuários Finais**. 2001.

Tese de doutorado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil. Agosto de 2001.

CYPHER, A., **EAGER**: programming repetitive tasks by example, In: Proceedings of CHI '91, New Orleans, Abril de 1991. New York: ACM. p. 33 - 39.

CYPHER, A. **Watch what I do**: Programming by Demonstration. Cambridge, MA: The MIT Press, 1993. 662 p.

DA SILVA, B. S.; BUENO, A. M.; BARBOSA, S. D. J.; In: **___Análise dos mecanismos de adaptação do Mozilla Thunderbird** Departamento de Informática, PUC-Rio, Rio de Janeiro. Março de 2008.

DA SILVA, S. R. P. **Um Modelo Semiótico para Programação por Usuários Finais**. 2001. Tese de doutorado - Pontifícia Universidade Católica do Rio de Janeiro, Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil. 2001.

DE SOUZA, C. S. Semiotic engineering principles for evaluating end-user programming environments. In: **___Monografias em Ciência da Computação**, Departamento de Informática, PUC-Rio, Rio de Janeiro. Outubro de 1999.

DE SOUZA, C. S. **The Semiotic Engineering of Human-Computer Interaction**. Cambridge, Mass.: The MIT Press, 2005. 307p.

DE SOUZA, C. S.; BARBOSA, S. D. J. A semiotic framing for end-user development. In: **___End User Development**. New York: Springer, 2006. p. 401-426.

DE SOUZA, C. S.; BARBOSA, S. D. J.; SILVA, S. R. P. **Semiotic engineering principles for evaluating end-user programming environments**. In: Interacting with Computers, Elsevier, Amsterdam, 2001. v. 13, p. 467 - 495.

DE SOUZA, C. S.; LEITE, J. C.; PRATES, R. O.; BARBOSA, S. D. J. **Projeto de Interfaces de Usuário** – Perspectivas Cognitivas e Semióticas, Anais da Jornada de Atualização em Informática, XIX Congresso da Sociedade Brasileira de Computação, Rio de Janeiro, julho de 1999. Disponível em <http://www.dimap.ufrn.br/~jair/piu/JAI_Apostila.pdf>. Acesso em 12 ago. 2007.

DIETERICH, H.; MALINOWSKI, U.; KÜHME, T.; SCHNEIDER-HUFSCHMIDT, M. State of the Art in Adaptive User Interfaces. In: **___Adaptive User Interfaces: Principle and Practice**. Amsterdam, North Holland: Elsevier Science & Technology Books, Junho de 1993. p. 13 – 48.

DOUCE, C. **Metaphors we program by**. In 16th Annual Workshop of the Psychology of Programming Interest Group, Carlowe, Ireland, Abril de 2004.

FERNANDEZ, J.; LEMOS, G.; SILVEIRA, G. **Introdução à Televisão Digital Interativa**: Arquitetura, Protocolos, Padrões e Práticas. In: JAI-SBC, Salvador, 2004.

EISENBERG, M.; FISCHER, G.; **Programmable Design Environments**: Integrating End-User Programming with Domain-Oriented Assistance. In Proceedings of CHI'94 (Human Factors in Computing Systems). Boston, MA. 1994. p.431 - 437.

FISCHER, G.; GIRGENSOHN, A. **End-User Modifiability in Design Environments**. In: Human Factors in Computing Systems, CHI'90 Conference Proceedings, New York: ACM, 1990. p. 183 - 191.

FISCHER, G. **Beyond 'couch potatoes'**: From consumers to designers In: Proceedings of APCHI 98, IEEE Computer Society, Washington, DC, USA, 1998. p. 2 – 9.

FISCHER, M.; CAO, M.; ROTHERMEL, G.; COOK, C.; BURNETT, M. **Automated test generation for spreadsheets**. In: Proceedings of International Conference of Software Engineering, Orlando, Maio de 2000. p. 141 - 151.

GOULART, R. **Personalização e adaptação de conteúdo baseadas em contexto para TV interativa**. 2003. Tese de doutorado - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos.

GUIMARÃES, R. L. **Composer**: um ambiente de autoria de documentos NCL para TV digital interativa. 2007. Dissertação de mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.

GUZDIAL, M.; WEINGRAD, P.; BOYLE, R.; SOLOWAY, E. Design Support Environments for End Users. In: **___Languages for Developing User Interfaces**. Boston, Mass.: Jones & Barlett Publishers, 1992. p. 57 – 78.

HACKOS, J. T.; REDISH, J. C. **User and task analysis for interface design** New York, NY: John Wiley & Sons, Inc., 1998. 512p.

HARTSON, H. R. **Human-computer interaction**: Interdisciplinary roots and trends. In: The Journal of System and Software, 1998. v. 43, p. 103 - 118.

HENRICKSEN, K.; INDULSKA, J. **Adapting the Web Interface**: An Adaptive Web Browser. In: Proceedings of the Second Australasian User Interface Conference, Los Alamitos, CA, USA: IEEE Computer Society, 2001. v. 23, p. 21 – 28.

HÖÖK, K. **Evaluating the utility and usability of an adaptive hypermedia system**. In: Proceedings of the 2nd international conference on Intelligent user interfaces - International Conference on Intelligent User Interfaces, 1997. New York, NY, USA: ACM, 1987. p. 179 – 186.

HÖÖK, K.; KARLGREN, J; WAERN, A. **A Glass Box Approach to Intelligent Help**. In: First workshop on Intelligent MultiModal Interaction - IMMI 1, 1995. Edinburgh, U.K.

HYPERCARD. Apple© HyperCard® Script Language Guide(1993). Apple Computer, Inc.

KAHN, K.; **ToonTalk™** - An Animated Programming Environment for Children. In: The Journal of Visual Languages and Computing, Academic Press, Junho de 1996. v. 7, n. 2, p. 197-217(21).

KO, A. J.; MYERS, B. A. **Designing the Whyline**: A Debugging Interface for Asking Questions About Program Failures. In: CHI 2004. Vienna, Austria. 24-29 de Abril de 2004. p. 151 - 158.

KO, A. J. AND MYERS, B. A. **Barista**: An Implementation Framework for Enabling New Tools. In: Proceedings of the SIGCHI Conference on Human

Factors in Computing Systems, CHI 2006, Montréal, Québec, Canada, 2006. p. 387 – 396.

KOCH, N. P. **Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process**. 2000. Tese de Doutorado - Institut für Informatik, Ludwig-Maximilians, Universität München, 2000.

LAKOFF, J.; JOHNSON, M. **Metaphors we live by**. Chiacgo, USA: University Of Chicago Press, Abril de 1980. 256p.

LIEBERMAN, H.; Tinker: A Programming by Demonstration System for Beginning Programmers, In: **___Watch What I Do: Programming by Demonstration**. Cambridge, MA: MIT Press, 1993. p. 49 – 64.

LIEBERMAN, H.; LIU, H. Feasibility Studies for Programming in Natural Language. In: **___End User Development**. New York: Springer, 2006. p. 459 - 473.

LIEBERMAN, H.; PATERNÒ, F.; KLAN, M.; WULF, V. End-User Development: An Emerging Paradigm. In: **___End User Development**. New York: Springer, 2006. p. 1 - 8.

LIEBERMAN, H.; PATERNÒ, F.; WULF, V. **End User Development**. Kluwer/Springer, Human-Computer Interaction Series, Junho de 2006. v. 9, 492p.

MACKAY, W. E. **Triggers and barriers to customizing software**. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Reaching through technology, 27 de Abril a 02 de Maio de 1991. New York, NY, USA: ACM, 1991. p. 153 - 160.

MØRCH, A. Three Levels of End-User Tailoring: Customization, Integration, and Extension. In: **___Computers and Design in Context**. Cambridge, MA: The MIT Press, 1997. p. 51 - 76.

MUCHALUAT-SAADE, D. C.; SILVA, H. V. O; SOARES, L. F. G. **Linguagem NCL versão 2.0 para Autoria Declarativa de Documentos Hipermedia**. In: IX Simpósio Brasileiro de Sistemas Multimídia e Web - WebMídia2003, 2003, Salvador. Anais do IX WebMídia, 2003.

MYERS, B. A. **Introduction to visual programming and program visualization**. In: Proceedings of Computer/Human Interaction, 1989 – CHI '89. Austin, Texas. Workshop on Visual Computing Environments.

MYERS, B. A. **Evaluation of visual programming and program visualization**. In: Proceedings of Computer/Human Interaction, 1989 – CHI '89. Austin, Texas. Workshop on Visual Computing Environments.

MYERS, B. A. **Languages for Developing User Interfaces**. Boston: Jones and Bartlett, 1992. 480p.

MYERS, B. A.; PANE, J. F.; KO, A.; **Natural programming languages and environments**. In: Communications of the ACM, special issue on End-User Development. Setembro de 2004. v. 47, p 47 - 52.

NARDI, A. B. **A Small Matter of Programming: Perspectives on End User Computing**. Cambridge, Mass.: MIT Press, Julho de 1993. 178p.

NIELSEN, J. **Usability Engineering**. Boston, MA: Academic Press, 1993. 362p.

NORMAN, D. Cognitive engineering. In ____ **User centered system design: New perspectives on human-computer interaction**, Hillsdale, NJ: Lawrence Erlbaum Associates, 1986. p. 32 - 65.

NORMAN, D. A. **The design of future things**. NY: Basic Books, Novembro de 2007 (originalmente publicado como *The Psychology of everyday Things*. NY: Basic Books, Abril de 1988). 272p.

PANE, J. **A programming system for children that is designed for usability**. 2002. Tese de doutorado - Carnegie Mellon University, Pittsburgh, PA. Disponível em <http://www.cs.cmu.edu/~pane/thesis>. Acesso em 10 de ago. 2007.

PANE, J.F.; RATANAMAHATANA, C. A.; MYERS, B. A. **Studying the Language and Structure in Non-Programmers' Solutions to Programming Problems**. In: *International Journal of Human-Computer Studies*, Fevereiro de 2001. v. 4, p. 237 - 264.

PAUSCH, R.; BURNETTE, T.; CAPEHEART, A.C.; CONWAY, M.; COSGROVE, D.; DELINE, R.; DURBIN, J.; GOSSWEILER, R.; KOGA, S.; WHITE, J. **A Brief Architectural Overview of Alice, a Rapid Prototyping System for Virtual Reality**. In: *IEEE Computer Graphics and Applications*, Maio de 1995. p. 195 – 203.

PEIRCE, C. S. 1931 – 1958. “**Collected Papers**”. Cambridge, MA: Harvard University Press, 1960. v.6, p. 452 – 493.

PRATES, R. O, A. **Engenharia Semiótica para o Domínio Educacional**. Atas da Oficina "Design e Avaliação de Interfaces para Ambientes Educacionais" organizado no IHC 2004. Disponível em <<http://www.ime.uerj.br/~raquel/wied/ihc2004/>>. Acesso em 10 de nov. de 2007.

PRATES, R. O.; BARBOSA, S. D. J.; DE SOUZA, C. S. **A Case Study for Evaluating Interface Design through Communicability**. In: *Proceedings of the ACM Designing Interactive Systems: processes, practices, methods, and techniques - DIS'2000*. New York, NY, USA: ACM, Agosto de 2000. p. 308 – 316.

PREECE, J.; ROGERS, Y.; SHARP, H. **Design de Interação**. Porto Alegre: Bookman, 2005. 348p.

REPENNING, A.; **Bending Icons**: Syntactic and Semantic Transformations of Icons. In *VL'94: IEEE VL Symposium*. St. Louis, Missouri, 1994. p. 296 – 303.

REPENNING, A.; IOANNIDOU, A. What Makes End-User Development Tick? 13 Design Guidelines In: ____ **End User Development**. New York: Springer, 2006. p. 51 – 85.

RODRIGUES, R. F.; SOARES, L. F. G. **Formatação e controle de apresentações hipermídia com mecanismos de adaptação temporal**. 2003. Tese de Doutorado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro. Disponível em <ftp://ftp.telemidia.puc-rio.br/pub/docs/theses/2003_03_rodrigues.pdf> Acesso em 05 de ago. de 2007>.

RODRIGUES, R. F. ; SOARES, L. F. G. **Produção de Conteúdo Declarativo para TV Digital**. In: *Seminário Integrado de Software e Hardware*, 2006, Campo Grande. XXXIII SEMISH - Seminário Integrado de Software e Hardware (aceito

para publicação). Porto Alegre: Sociedade Brasileira de Computação, 2006. p. 286 - 300.

SCAFFIDI, C.; MYERS, B.; SHAW, M. **The Topes Format Editor and Parser**. Pittsburgh, PA: Carnegie Mellon University, School of Computer Science, 2007, 19 p. Relatório Técnico. .

SHU, N. **A visual programming language designed for automatic programming**. In: Proceedings of 21st Annual Hawaii International Conference on System Sciences, 1988. v.2, p. 662 - 671.

SOARES, L. F. G.; RODRIGUES, R. F.; MUCHALUAT-SAADE, D. C. **Modelo de Contextos Aninhados** – versão 3.0, Relatório Técnico - Laboratório TeleMídia, Departamento de Informática, PUC-Rio, 2003.

SUTCLIFFE, A.G.; LEES D.; MEHANDJIEV, N., **Contributions, Costs and Prospects for End-User Development**. In: Proceedings of HCII'03. Crete, Greece, 24 - 27 de junho de 2003. v. 2,

TIRESIAS. **Inclusive Design**. Disponível em <http://www.tiresias.org/guidelines/> Acesso em 12 de ago. de 2007.

TRIGG, R. H.; MORAN, T. P.; HALASZ, F. G. **Adaptability and tailorability in note cards**. In: Proceedings of Interact'87 2nd IFIP Conference on Human-Computer Interaction, Stuttgart, North Holland, September 1987. p. 723 – 728.

WALDIE, B. **Automator for Mac OS X 10.5 Leopard**: Visual QuickStart Guide. Peachpit Press, 2008. 288 p.

Apêndices

Apêndice A Estudo Preliminar – Conhecendo os Usuários

Preparação para os testes

Como as entrevistas se enquadram na categoria **pesquisas envolvendo seres humanos**, foi desenvolvido um protocolo de ética para ser submetido ao Comitê de Ética em Pesquisa da PUC-Rio. Esta prática está de acordo com o Manual Operacional de CEPs – CONEP: 2002, p. 47¹. O protocolo ainda está sendo revisado.

Assim como qualquer pesquisa que envolva usuários, foi desenvolvido um termo de consentimento, apresentado em duas vias no momento das entrevistas. Ambas devem ser assinadas pelo usuário e pelo pesquisador (e observador, se houver), confirmando o consentimento da realização da entrevista. Uma via deve ser entregue ao usuário e a outra fica com o pesquisador. Neste documento há uma breve explicação sobre o trabalho e algumas informações de interesse do usuário como a explicação sobre o anonimato dos dados, a possibilidade de interrupção do teste caso seja do interesse do usuário e necessidade de gravação das atividades para futuras consultas. O termo de consentimento (Termo de consentimento para entrevista) encontra-se no final deste anexo.

Convocação

Os profissionais foram convidados via e-mail. A carta-convite encontra-se no final deste anexo. As entrevistas foram agendadas de acordo com o horário e local escolhido pelos convidados.

¹ <http://conselho.saude.gov.br/>

Elaboração de material

Foram elaborados dois exemplos de programas para TV Digital. O primeiro programa criado foi desenvolvido usando a linguagem NCL, de maneira que o mesmo pudesse ser interpretado pelo formatador¹. O programa permite que o usuário obtenha informações sobre um filme, no momento que ele está sendo exibido. As informações são o elenco (e dados relevantes sobre os atores), dados sobre o filme e cenas que não foram ao ar.

O segundo programa permite que o usuário monte o seu próprio telejornal. Apenas isto foi informado ao convidado. Foram tirados *dumps* de tela, em vários tamanhos, dos vídeos para que pudessem ser impressos e manuseados pelo convidado. Foram impressos também textos e imagens dos botões do controle remoto. Um vídeo similar de telejornal foi exibido repetidamente enquanto o participante criava seu programa. Isto foi feito apenas para que ele entendesse o contexto do material que recebeu.



Figura 110: Material usado para a criação do programa de jornal

¹ Formatador é a ferramenta de apresentação de documentos NCL (Rodrigues et al., 2003).

Realização

Alguns profissionais dos Departamentos de Artes e Comunicação da PUC-Rio, assim como outros da TV PUC, foram convidados a participar de uma entrevista individual realizada no laboratório de Engenharia Semiótica da PUC-Rio – SERG.

A entrevista foi realizada em três etapas. A primeira consistiu em uma conversa com o participante para conhecer melhor como ele realiza seu trabalho e suas experiências. Foi elaborado um roteiro para que o entrevistador se focasse nas partes interessantes ao estudo. Este roteiro encontra-se no final deste anexo.

A segunda e terceira etapas ocorreram com atividades práticas. Na segunda, o convidado assistiu a um programa (quantas vezes fosse necessário) desenvolvido em NCL, interpretado pelo formatador – o simulador da TV. Foi pedido a ele que explicasse o funcionamento deste programa, assim como todos os detalhes que fossem relevantes. O objetivo desta etapa foi capturar como o profissional se expressa com relação aos elementos que compõem o programa. Desta forma, o profissional fornece indícios sobre como ocorre a leitura e a interpretação do programa.

Na terceira etapa, o pesquisador mostrou ao convidado alguns materiais como sequência de fotos, textos, papéis coloridos no formato de botões, papéis em branco etc. Ele, então, pediu que o convidado construísse um determinado programa utilizando o material fornecido, narrando o que ele estava pensando e fazendo no momento. A importância desta etapa é analisar como ocorre o processo de concepção e montagem, a partir de uma idéia inicial e de mídias já produzidas.

A primeira etapa da entrevista foi gravada utilizando um gravador de áudio. Na segunda e terceira etapas, utilizou-se uma câmera para filmar as movimentações do convidado. As gravações foram permitidas pelos convidados, no momento de leitura e assinatura do termo de consentimento.

Resultado dos Testes

Perfil

O teste foi realizado com três profissionais: um designer, um jornalista e um profissional de comunicação social. Com relação à experiência dos participantes em programação, um deles já havia programado utilizando linguagens de marcação, como HTML, mas sabe apenas o básico da linguagem, e sempre usou programas de autoria para auxiliá-lo nesta tarefa (P2). O outro, apesar de ter noções de programação, só aprendeu Cobol e Pascal, mas nunca trabalhou especificamente com desenvolvimento (P1). Além disto, ele conhece algumas tags HTML, mas só trabalha com o código para descobrir problemas. Apesar da experiência prévia em programação, os participantes continuam dentro do perfil da pesquisa uma vez que não são desenvolvedores, e o pouco que conheceram de programação não envolveu a linguagem em questão, a NCL.

Todos os participantes mencionaram ter alguma experiência com programas de edição de vídeo, e os três citaram o Adobe Premiere como *software* para esta finalidade. Os softwares utilizados para outros fins tais como autoria de multimídia, foram o ToolBook, Authorware, Director, DVD Studio e Photoshop.

Nenhum participante conhecia a NCL. Dois deles já ouviram falar, mas desconheciam a fundamentação e estrutura da linguagem.

Experiência de trabalho

O participante P1 trabalha com soluções para a área de vídeo e para a área de comunicação para agências de publicidade, editoras, escritórios de design, produtoras de vídeo e televisão. O participante P2 é designer e autor e trabalha com narrativas e histórias para crianças, transformando todo esse conteúdo em mídias como CD-ROM. O participante P3 é jornalista, trabalha com edição, concepção, direção e produção de produtos para a televisão.

Produção de programas audiovisuais

Todos os participantes já participaram de projetos para a concepção de programas audiovisuais, porém, não passaram da etapa de prototipação determinada por cada empresa ou setor onde trabalham. Os programas não chegaram a ser desenvolvidos na prática.

P1 produz DVDs lineares com os programas. A interatividade é apenas simulada, para mostrar as finalidades dos botões e mídias.

“A gente só mostra que pode ter interatividade, a gente simula a interatividade, na hora da edição, a gente finge que o cara apertou o botão e mostra o que acontece.”

P2 também participou da concepção de programas criando um CD-Rom para demonstrar seu funcionamento. Ele usava o Authorware, o Dreamweaver e às vezes o Flash para auxiliá-lo nesta tarefa.

P3 possui como meta de trabalho o conteúdo e produz roteiros que relatam as propriedades dos programas, a interatividade e o conteúdo propriamente dito.

Metodologia para a produção de programas

O participante P1 começa a trabalhar a partir de uma idéia.

“A gente tem uma necessidade qualquer e a gente precisa fazer um programa de esporte com interatividade.”

A partir da idéia, a equipe se reúne para uma etapa de brainstorming, para discutir quais interatividades podem ser atribuídas ao programa, sobre qual esporte será este programa etc. Todas as características do programa são discutidas e definidas nesta etapa.

“Que esporte vai ser? Vai ser... escalada. Que interatividade a gente vai colocar em um programa de escalada? (...)

Isso vai no ar, um programa que vai ter duração de 10 minutos. E o que vamos poder colocar ali? (...)

Vamos ter patrocínio. Patrocínio de quem? De uma loja de esportes. Patrocínio de um mercado, Zona Sul. E uma escalada na Pedra da Gávea que tem um Zona Sul ali embaixo. O Zona Sul pode vender ali. Você pode comprar uma água, uma barra de cereal.”

Neste caso, eles começam a pensar no que pode fazer sentido, sem pensar na interação ainda. Note como todo o programa vai sendo elaborado de acordo com as idéias que eles vão tendo. A partir daí, cada participante da equipe começa a pensar em idéias de interatividade com os dados de programa fornecidos anteriormente. Depois, a equipe se reúne para discutir as idéias e definir a interatividade. Nesse exemplo específico de programa, eles filmaram uma escalada, editaram as imagens usando o Photoshop e construíram o programa

audiovisual usando um software de edição de vídeos, onde toda a interatividade pode ser simulada.

A equipe do participante 2 é mais diversificada. Ela é formada por jornalistas, designer, pessoas de comunicação social e programadores. Eles formavam reuniões de trabalho e construíam uma documentação, um briefing do projeto. Neste documento havia características de projeto, demanda de tecnologia, organograma e um fluxograma. No fluxograma, havia um roteiro das telas e do conteúdo, e marcações de momentos de inserção de vídeo, banco de dados etc.

Cada profissional responsável por uma determinada área possuía toda a informação necessária para o desenvolvimento do projeto na documentação. O fluxograma era feito em programas como o PowerPoint, que também era usado para a documentação, quando esta não era feita no Word.

O terceiro participante cria os programas a partir de uma história. Juntamente com outra jornalista, elas constroem o programa de acordo com as cenas da história. Depois, de pensar em todo o conteúdo, elas conversam com um profissional técnico, para saber o que poderia realmente ser feito e o que deveria ser alterado.

Toda a etapa de decisão sobre as características do programa era feita usando papel.

[O participante desenhando em um papel dizia:] "Não, aí imagina, tem a cena e tem um garotinho sentado com a televisão grandona aqui. Aí aqui podiam surgir ícones, que não atrapalham..."

Da mesma forma que o participante anterior, este cria roteiros para descrever as funcionalidades do programa.

Conteúdo apresentado na tela	Comportamento do programa ¹
CRÉDITO A menina Direção: Maria Bonita/ Música: Charles Chaplin Autorização: Continental Filmes Tel: (11) 3333-3333 E-mail: valeria@terra.com.br	SOBE SOM "A MENINA"
"A gata andaluz" - Luisa Becker (1928/30) Autorização: Versátil Vídeo Tel:	OFF 1 ???CINEMA MUDO??? NEM TANTO.// MESMO QUANDO OS FILMES NÃO ERAM FALADOS, O CLIMA DA NARRATIVA JÁ ERA CRIADO COM MÚSICAS.//

Tabela 18: Exemplo de um roteiro usado pelo participante 3 para descrever o programa

Sobre softwares de autoria

Os três participantes já usaram ou ainda usam softwares de autoria para auxiliar o desenvolvimento dos programas audiovisuais. Um software comum aos três é o Adobe Premiere, que auxilia a produção de vídeos.

Sobre as facilidades, apenas o participante 2 mostrou-se satisfeito com o software Authorware, pois além de ser muito mais acessível, pois sua estrutura poderia ser usada para a produção, ele possui scripts prontos e facilita a edição deles.

[Falando sobre o Authorware:] “E eu me lembro que ele era muito didático, do tipo, se você tinha que escrever uma função e trocar algum campo, ele dizia para você que se você tirar um, tira isso, já abria uma própria caixinha de função. Ele dava realmente aquela opção para o usuário final. O outro não, você tinha que ter um manual do lado para saber que aquele campo...”

Além disto, ele citou a manipulação direta como ponto forte de softwares de autoria.

Sobre as dificuldades, o primeiro participante apontou a discrepância do que deveria ser WYSIWYG. Como ele teve experiências ruins com softwares que produziam uma simulação do produto diferente da real, ele se preocupa muito com isto. Então, ele acha que um software de autoria, deve simular corretamente o produto, o que não deve ser diferente para programas de TV Digital.

“Será que eu, produtor, tenho como ter certeza que o que eu estou produzindo ali na ferramenta, no Composer, será que vai rodar no set-top-box da Gradiente, da Sony, de não sei mais quem?”

“Se eu tiver que fazer alguma coisa para a TV e eu puder visualizar na TV, eu não confio hoje na visualização que essas ferramentas oferecem.”

O segundo participante mencionou o software Director como problemático devido a sua exigência por scripts.

“O Director você tinha que lembrar esse script, não dava pronto, gravado em algum lugar. Você às vezes tinha que escrever e eu percebia que o usuário às vezes ficava... era muita coisa, não sabia revisar aquilo e dava um problema, a coisa não funcionava.”

¹ Este cabeçalho foi introduzido pela autora para facilitar o entendimento do leitor.

O terceiro participante possui experiência apenas com o Premiere. Ele aprendeu a usar o software com um aluno. Sua experiência de uso é ainda muito recente e ele não soube dizer os pontos de facilidade e dificuldade, porque como era tudo novo, era tudo difícil, principalmente porque ele tinha medo de ousar.

“A gente tem um pouco de receio por duas coisas: primeiro porque o equipamento é caro e seguindo porque o material que você vai colocar ali é precioso. Então imagina se você destrói... Eu já perdi um vídeo todo editado!”

Sobre a aprendizagem e ajuda

Apenas o participante 1 afirmou que quando possui algum problema, procura a ajuda do software. Apenas quando não encontra a solução para o seu problema, ou não a entende, ele procura ajuda de outras pessoas. O mesmo ocorre quando ele deseja aprender a usar algum programa.

Os outros dois participantes sempre pedem auxílio a uma outra pessoa para aprender a usar um determinado programa ou resolver um problema.

Sobre a explicação do programa pronto

O participante 1 mostrou-se preocupado em mostrar ao usuário (referindo-se à pessoa que irá interagir com o programa) quando uma dada interatividade é alterada.

“Você ser informado de alguma forma que apareceu uma interação. Ainda que fosse mudar a cor do botão. Então, apareceu uma interatividade nova... ao invés de você começar com o botão vermelho que é o padrão, começa com o azul, começa com o verde. Aí o usuário vai dizer: ‘Opa! Mudou alguma coisa’.”

Ele também informou que ficou sem saber quando o “botão da interatividade apareceu”. A explicação sobre a importância desta informação foi que o usuário poderia ficar na dúvida se ele havia perdido alguma interatividade por não prestar atenção ao botão ou não.

“Eu fiquei sem saber se na hora que eu vi, se ele já estava ali. Eu achei que talvez ele estivesse discreto demais.”

Ele citou a utilização de um padrão, por exemplo, um ícone sempre no mesmo lugar. Outra preocupação foi com a área útil da tela. Ele informou que mesmo com o redimensionamento do vídeo para o aparecimento de um texto explicativo, a tela ficou com uma área preta vazia. Nesse caso, ele sugeriu que fosse re-diagramada.

“O mais importante disso aqui é o vídeo. Por mais que eu vá ler alguma coisa, eu quero continuar vendo o vídeo (...). Eu tentaria diagramar isto de uma maneira que, ao invés do texto ficar na vertical, ele ficaria na horizontal.”

O participante não notou a presença de um botão de interação, e, quando o entrevistador perguntou sobre isto, ele afirmou que não havia percebido o botão, talvez porque havia muito azul na tela também.

Falando mais sobre os programas de forma geral, ele exemplificou a idéia de um programa documentário. Neste tipo de programa, ele acha válida a presença de um glossário e uso de link para a visualização do mesmo.

“Evitaria que a pessoa saísse da televisão para ir a internet pesquisar, que é o que acontece hoje. Uma forma de enriquecer o programa interativo é disponibilizar todo o conteúdo ali.”

Além disto, ele simpatiza muito com a interatividade, principalmente quando ela envolve outras pessoas, como é o exemplo da enquete.

“A interatividade não deveria se limitar somente ao usuário e ao programa”.

Material

Termo de consentimento para entrevista

Fazemos parte da equipe de pesquisadores do Departamento de Informática da PUC-Rio, em particular do laboratório de engenharia semiótica, o SERG (Semiotic Engineering Research Group). Estamos realizando esta pesquisa em parceria com o Laboratório de Redes de Telecomunicações e Sistemas Multimídia - Telemídia.

Esta entrevista é parte de pesquisas de Mestrado e Doutorado de alunos pertencentes a esses laboratórios. O tema é construção de programas audiovisuais interativos por não-programadores. Para apoiar os profissionais ligados à elaboração de mídias interativas, gostaríamos de conhecer a sua experiência, entender como os programas audiovisuais são construídos atualmente e explorar como eles podem vir a ser construídos no futuro. A idéia é capturar como ocorre a comunicação e a interpretação sobre o funcionamento desses programas, para entender a linguagem usada pelos profissionais envolvidos nessas atividades. E você será um deles!

Por estas razões, solicito seu consentimento para a realização de uma entrevista e dois exercícios, bem como para a sua gravação em áudio e vídeo. Para tanto, é importante que você tenha algumas informações adicionais:

- Os dados coletados durante as atividades destinam-se estritamente a atividades de pesquisa e desenvolvimento.
- A divulgação destes resultados pauta-se no respeito a sua privacidade e o anonimato dos mesmos é preservado em quaisquer documentos que elaboramos.
- O consentimento para essas atividades é uma escolha livre, feita mediante a prestação de todos os esclarecimentos necessários sobre a pesquisa.
- As atividades podem ser interrompidas a qualquer momento, segundo a disponibilidade do participante.
- Nossa equipe encontra-se disponível para contato através do telefone 3527-1500, ramal 3323, ou pelo e-mail serg@inf.puc-rio.br.

De posse das informações acima, gostaria que você se pronunciasse acerca das atividades propostas.

() Dou meu consentimento para a sua realização.

() Não dou meu consentimento para a sua realização.

Rio de Janeiro, _____.

Participante	Observador
Nome: _____	Nome: _____
Assinatura: _____	Assinatura: _____

Roteiro da entrevista¹

Dados de identificação
<ul style="list-style-type: none"> ▪ Nome ▪ Formação / Profissão ▪ Local de trabalho ▪ Atividades que realiza no trabalho
Programadores
<p>Trabalha (ou já trabalhou) com quais linguagens de programação?</p> <p>Trabalho atual envolve que linguagens?</p> <p>Alguma linguagem declarativa? Qual?</p> <p>Quais são as principais características dessas linguagens?</p> <p>Já trabalhou com HTML? XML? Quanta experiência?</p> <p>Já fez algum programa audiovisual?</p> <p>Como ele era?</p> <p>A quem era destinado?</p> <p>Como foi concebido o projeto?</p> <p>Usou algum software para a construção?</p> <p>Qual linguagem usava? Usava a NCL?</p> <p>Já usou alguma ferramenta de autoria?</p> <p>Qual?</p> <p>Como era?</p> <p>Quais dificuldades/facilidades você encontrou?</p> <p>Qual a preferência? (código ou ferramenta?)</p> <p>Conhece a NLC?</p> <p>Como aprendeu? (tutorial, cursos)</p>
Comunicação / jornalismo / artes
<p>Você participa de projetos (concepção/execução) de programas audiovisuais interativos?</p> <p>Em que atividades?</p> <p>Como é concebido um programa?</p>

¹ Este roteiro serve como guia para que o avaliador não esqueça de fazer as principais perguntas relacionada ao estudo. Eventualmente outras perguntas foram realizadas de acordo com as respostas dos participantes.

Como se decide sobre mídias, interatividade?

Usa algum software para apoiar suas atividades?

Qual?

Como aprendeu?

Costuma usar “sozinho”, consulta o manual, colega? (Precisa de ajuda para usá-lo?)

Já trabalhou com algum tipo de programação, script, macro, customização do software que usa?

O que era?

Como era?

Como foi feito?

Por quem?

Sobre o trabalho com os programas audiovisuais

Elaboração dos programas

Temas

Roteiro

Recursos (vídeos, imagens)

Pessoas envolvidas na elaboração [descobrir mais sobre cada ‘quem’: formação, experiência, softwares utilizados, representações utilizadas]

Quem elabora?

Quem participa da criação?

Quem revisa?

Quem colabora com os recursos?

Quem faz a montagem final?

Criação dos programas

Etapas

Como realmente é feito?(Papel, PowerPoint, outro software)

Há algum teste? Protótipos?

Pedir para exemplificar a criação de um programa específico criado por ele, ou que ele tenha participado.

Carta Convite

Oi, (nome do convidado).

Sou aluna de mestrado da Professora Simone D. J. Barbosa, do Departamento de Informática. O tema da minha dissertação é construção de programas audiovisuais interativos por não-programadores. Parte do meu trabalho de pesquisa é realizar entrevistas com profissionais ligados à elaboração de mídias interativas para conhecer o trabalho realizado atualmente na construção dos programas e explorar como eles poderão ser construídos futuramente.

As entrevistas serão individuais e realizadas em duas etapas. Primeiro, irei conversar com o participante para colher dados sobre a sua formação, experiência e perfil profissional. A segunda etapa consiste na realização de duas tarefas. Mostrarei ao participante um programa pronto e pedirei que ele "narre" a execução do mesmo. Depois, pedirei que ele construa um determinado programa, usando material fornecido por mim como fotos, cartolina, papel etc.

A idéia é capturar como ocorre a comunicação e a interpretação sobre o funcionamento desses programas, para entender a linguagem usada pelos profissionais. A sessão inteira, incluindo entrevista e atividades, deve durar aproximadamente 2 horas. Ela poderá ser realizada na própria PUC. Caso o participante não tenha preferência de local, sugerimos fazer no nosso laboratório (SERG, no prédio ITS).

Desta forma, gostaria de saber se você gostaria de participar dessa entrevista ou se pode me indicar outras pessoas do (departamento do convidado) que atualmente estejam envolvidas em projetos sobre TV Digital e que poderiam participar desta pesquisa.

Desde já agradeço.

Ariane Moraes Bueno

Apêndice B**Questionário sobre o Composer – Utilizado nos Cursos Ginga/NCL****Perguntas**

1. Você já tinha usado o Composer anteriormente?
 - a. sim
 - b. não

Se sim, como ouviu falar dele?
2. Em que pontos Composer ajuda ou atrapalha a entender a NCL?
3. Em que pontos a NCL ajuda ou atrapalha a entender o funcionamento do Composer?
4. O Composer ajuda:
 - a. quem sabe mais da NCL
 - b. quem sabe menos da NCL

Se possível, elabore sua resposta em mais detalhes.
5. Você prefere:
 - a. usar o Composer do que criar um código de programa
 - b. escrever o código em um editor e importá-lo para o Composer, para analisar a estrutura e detectar problemas
 - c. escrever o código no Composer apenas
 - d. escrever o código em outro editor
6. Sobre o apoio do Composer para a criação de programas: (marque apenas uma opção por linha)

	discordo totalmente	discordo parcialmente	indiferente	concordo parcialmente	concordo totalmente
A interface do Composer facilita o entendimento sobre o que o sistema se destina a fazer.					
A estrutura de menus é de fácil entendimento e auxilia iniciantes.					
A divisão por visões facilita					

o entendimento da estrutura do programa.					
As mensagens de erro ajudaram a descobrir e recuperar erros.					
Consegui criar e usar todos os elementos da NCL para os quais o Composer fornece auxílio de criação.					

Por favor, comente os itens com os quais discorda total ou parcialmente

7. Você acha que as funcionalidades do Composer correspondem bem aos elementos da linguagem NCL?
 - a. sim
 - b. não

Por quê?









8. Por favor utilize esse espaço para fazer comentários sobre a experiência que teve com o Composer e/ou sugerir mudanças.

Resultado da análise quantitativa

A primeira parte do questionário diz respeito ao perfil dos participantes, enquanto a segunda se refere ao uso do Composer e aprendizado da NCL. Nem todas as perguntas da parte 1 foram transcritas para este documento para preservar o anonimato dos usuários, assim como algumas respostas. As respostas estão apresentadas exatamente da mesma forma que foram escritas.

Parte 1 - perfil

1. Formação acadêmica:

1o. grau incompleto		4.8%
1o. grau completo		10.8%
2o. grau completo		38.9%
curso técnico		12.6%
ensino superior / graduação		28.7%
mestrado profissionalizante (pós-graduação lato sensu)		1.2%
mestrado acadêmico		1.8%
doutorado		1.2%
pós-doutorado		0.0%

2. Experiência com linguagens de programação e de marcação:

	nenhuma	pouca	média	muita	especialista
ASP	93.4% (156)	4.2% (7)	0.6% (1)	1.8% (3)	0.0% (0)
ASP.NET	94.6% (158)	4.2% (7)	0.6% (1)	0.6% (1)	0.0% (0)
C	70.7% (118)	9.6% (16)	14.4% (24)	5.4% (9)	0.0% (0)
C++	76.0% (127)	14.4% (24)	6.6% (11)	3.0% (5)	0.0% (0)
C#	91.0% (152)	5.4% (9)	3.6% (6)	0.0% (0)	0.0% (0)
Delphi/Pascal	79.0% (132)	8.4% (14)	9.6% (16)	3.0% (5)	0.0% (0)
ECMAScript	95.2% (159)	3.6% (6)	0.6% (1)	0.0% (0)	0.6% (1)
HTML	32.3% (54)	29.9% (50)	29.3% (49)	7.2% (12)	1.2% (2)
Javascript	73.7% (123)	16.8% (28)	8.4% (14)	1.2% (2)	0.0% (0)
Java	68.9% (115)	15.6% (26)	9.6% (16)	5.4% (9)	0.6% (1)
Lua	99.4% (166)	0.6% (1)	0.0% (0)	0.0% (0)	0.0% (0)
NCL	65.3% (109)	19.8% (33)	13.2% (22)	1.2% (2)	0.6% (1)
PERL	98.2% (164)	1.2% (2)	0.6% (1)	0.0% (0)	0.0% (0)
PHP	74.3% (124)	14.4% (24)	10.2% (17)	1.2% (2)	0.0% (0)
Python	95.2% (159)	4.2% (7)	0.6% (1)	0.0% (0)	0.0% (0)
Ruby	98.8% (165)	0.6% (1)	0.6% (1)	0.0% (0)	0.0% (0)
Visual Basic	86.8% (145)	7.8% (13)	3.6% (6)	1.2% (2)	0.6% (1)
XML	68.9% (115)	19.8% (33)	9.6% (16)	1.2% (2)	0.6% (1)

Algumas pessoas responderam que possuem conhecimentos de Actionscript (para Adobe® Flash®) e banco de dados.

3. Experiência com ferramentas de autoria e programas de edição multimídia:

	nenhuma	pouca	média	muita	especialista
Adobe Premiere	68.3% (114)	10.8% (18)	12.0% (20)	6.6% (11)	2.4% (4)
Adobe Photoshop	29.3% (49)	31.7% (53)	26.3% (44)	10.8% (18)	1.8% (3)
Composer	69.5% (116)	21.0% (35)	9.0% (15)	0.6% (1)	0.0% (0)
CorelDraw	34.1% (57)	27.5% (46)	22.2% (37)	13.2% (22)	3.0% (5)
Flash	64.1% (107)	19.8% (33)	12.6% (21)	3.0% (5)	0.6% (1)
Final Cut	82.6% (138)	5.4% (9)	6.0% (10)	4.2% (7)	1.8% (3)
GRiNS	94.6% (158)	2.4% (4)	2.4% (4)	0.0% (0)	0.6% (1)
Macromedia Director	91.6% (153)	6.6% (11)	1.2% (2)	0.0% (0)	0.6% (1)
Microsoft Digital Image	82.6% (138)	7.8% (13)	7.2% (12)	2.4% (4)	0.0% (0)
Windows Movie Maker	56.9% (95)	15.0% (25)	7.8% (13)	15.6% (26)	4.8% (8)

Algumas pessoas responderam que possuem conhecimentos sobre o Adobe® Fireworks® e o GIMP, um editor gratuito de imagens. Outras repostas incluíram o Adobe® PageMaker® e o Adobe® InDesign®.

4. Qual sua área de atuação?

Academia		6.6%
Autônomo		9.6%
Comunidades		18.0%
Governo		6.0%
Indústria de Software		5.4%
Indústria de Eletrônicos		0.0%
Rádiodifusão		4.8%
Terceiro Setor		13.8%
Outra área		35.9%

5. Qual é sua experiência em programação de computadores?

Nenhuma. Sou usuário final.		53.7%
Programador C/C++		20.7%
Programador Java		16.5%
Programador em outras linguagens		17.7%
Programador Web (Javascript/AJAX/PHP...)		16.5%
Webdesigner (HTML/CSS/Flash)		31.7%

Parte 2










1. Você já tinha usado o Composer anteriormente?

sim	<input type="checkbox"/>	2.6%
não	<input checked="" type="checkbox"/>	97.4%

Algumas repostas incluíram o conhecimento através do Portal Ginga NCL, com professores e no próprio curso.

2. Em que pontos Composer ajuda ou atrapalha a entender a NCL?





Em geral, os pontos apresentados referiam-se à ajuda que o Composer fornece no entendimento da linguagem. A seguir são apresentadas algumas respostas.

	<i>“Ele atrapalha no momento em que se tem o primeiro contato com NCL. é muito mais interessante que este seja em documento xml para melhor absorção do que está sendo feito.”</i>
	<i>“(…) Contudo, depois de já conseguir olhar pro xml e perceber o funcionamento daquela estrutura com clareza, o composer torna-se um grande aliado no que diz respeito à facilidade com que se faz as aplicações e mesmo o tempo.”</i>
	<i>“O validador é muito importante para checar se o código está corretamente escrito. A visão de layout e estrutural também é muito importante.”</i>
	<i>“É a maneira mais prática de entender a linguagem.”</i>
	<i>“O composer é uma boa ferramenta de aprendizado NCL”</i>
	<i>“para mim em nenhum, pois tudo o que fazemos vidualmente, pode ser visto no modo texto.”</i>
	<i>“Ainda é uma ferramenta incompleta.”</i>
	<i>“Para quem é de uma áre legada a informática, engenharia da computação e ciências da computação é mais fácil fazer manulamente pq já vivenciam isso em suas vidas no curso. “</i>
	<i>“(…) Já quem está ligado a áreas ligada a artes, cinema, produção de video, ou comunicação social poderão gostar mais de trabalhar no composer, é o meu caso, mas a estruturação em si dada pelo programa poderá ser um pouco complicada logo de cara para</i>

	<i>aprender.””</i>
--	--------------------

3. Em que pontos a NCL ajuda ou atrapalha a entender o funcionamento do Composer?

Alguns alunos não entenderam o intuito desta pergunta e responderam coisas semelhantes à pergunta anterior. Os outros alunos, quando não eram indiferentes ou não responderam nada, responderam que a NCL é fundamental para o entendimento da ferramenta.

	<i>“A NCL ajuda no entendimento dos conceitos que a ferramenta expõe. Auxilia também na própria depuração do programa.”</i>
	<i>“Entender a linguagem NCL é essencial para o entendimento do software Composer.”</i>
	<i>“o NCL e a base da compressão do composer.”</i>
	<i>“Não vejo uma relação direta, é possivelmente capaz de criar uma ferramenta no qual o usuário não precise conhecer NCL. Talvez a evolução do Composer possa chegar a esse ponto. Mas atualmente é necessário saber NCL para entender como funciona o composer, pois hoje não tem como haver transparencia entre a construção gráfica e o código da NCL.”</i>

4. O Composer ajuda:

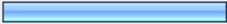



quem sabe mais da NCL		47.9%
quem sabe menos da NCL		52.1%

A maioria decidiu que o Composer ajuda quem sabe menos da NCL, porém a diferença é muito pequena. De forma geral, as pessoas que optaram por quem sabe mais da NCL associaram o conhecimento da NCL à prática e rapidez de uso do Composer.

- NCL	<i>“A associação das figuras (ou dos nós, elos, etc) e a interação entre elas, através do Composer, me deu um pouco mais de clareza do assunto. Creio que, para iniciantes o Composer auxilia na assimilação.”</i>
- NCL	<i>“Por ser uma ferramenta gráfica, ajuda a quem tem dificuldade em programação ou nunca programou. Também para quem sabe NCL,</i>

	<i>facilita o acesso, pois permite que a criação de elementos (regiões, nós, links) seja feita de maneira automática e detalhes possam ser posteriormente inseridos.”</i>
+ NCL	<i>“Creio que quem estudou NCL tem mais pratica.”</i>
a mbos	<i>“O composer ajuda na estruturação de quem sabe pouco NCL e na rapidez de quem sabe muito.”</i>
+ NCL	<i>“A pessoa conhecendo a linguagem ela é capaz de entender com mais clareza os layouts e os conceitos embutidos nesses layouts. Além do mais, ela consegue dominar com mais facilidade a ferramenta e utilizar as funcionalidades dela.”</i>
- NCL	<i>“Pra mim, o composer ajuda nas duas opções, mas, tende a facilitar o entendimento de pessoas que não tem conhecimento em outra linguagem.”</i>

5. Você prefere:

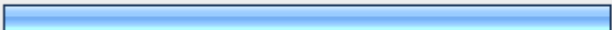

usar o Composer do que criar um código de programa		36.8%
escrever o código em um editor e importá-lo para o Composer, para analisar a estrutura e detectar problemas		44.7%
escrever o código no Composer apenas		5.3%
escrever o código em outro editor		13.2%





Sobre o apoio do Composer para a criação de programas:



	discordo totalmente	discordo parcialmente	indiferente	concordo parcialmente	concordo totalmente
A interface do Composer facilita o entendimento sobre o que o sistema se destina a fazer.	1.4% (1)	2.7% (2)	13.7% (10)	41.1% (30)	41.1% (30)
A estrutura de menus é de fácil entendimento e auxilia iniciantes.	0.0% (0)	4.1% (3)	5.5% (4)	52.1% (38)	38.4% (28)
A divisão por visões facilita o entendimento da estrutura do programa.	0.0% (0)	2.8% (2)	6.9% (5)	34.7% (25)	55.6% (40)
As mensagens de erro ajudaram a descobrir e recuperar erros.	6.8% (5)	5.5% (4)	11.0% (8)	45.2% (33)	31.5% (23)
Conseguir criar e usar todos os elementos da NCL para os quais o Composer fornece auxílio de criação.	1.4% (1)	8.2% (6)	15.1% (11)	47.9% (35)	27.4% (20)

Sobre a estrutura de menus	<i>“os menus nao são 100% faceis de entender...”</i>
Sobre as mensagens de erro	<i>“As mensagens de erro não ajudaram em nada. Alguns elementos tiveram que ser criados diretamente no código”</i>
Sobre o uso dos elementos da NCL	<i>“quase todos consegui, mas tem toda uma estrutura que ajuda”</i>

6. Você acha que as funcionalidades do Composer correspondem bem aos elementos da linguagem NCL? Por favor utilize esse espaço para fazer comentários sobre a experiência que teve com o Composer e/ou sugerir mudanças.

sim		75.3%
não		24.7%

	<i>“De acordo com o que foi visto no curso o composer tem tem uma ótima correspondencia em todos os aspectos”</i>
	<i>“Apesar de não conhecer bem a ferramenta, em geral o Composer tem atendido às necessidades.”</i>
	<i>“Por que as vezes precisamos detalhar melhor nosso video e o composer não permite isso.”</i>
	<i>“pois tudo que se pode fazer no composer é feito na linguagem</i>

	<i>NCL, só que de uma maneira mais prática e simples.”</i>
	<i>“Acho que a ferramenta composer ainda tem de ser melhor trabalhada para que a linguagem NCL seja melhor elaborada e melhor aproveitada...”</i>
	<i>“O Composer precisa o quanto antes agregar mais funções da linguagem NCL.”</i>

A versão do Composer utilizada apresentou alguns problemas de travamento e, em algumas ocasiões, a perda de código. Este problema, apesar de muito grave, não impediu que os usuários pudessem testá-lo e usá-lo. De maneira geral, os usuários fizeram pedidos para consertar esses problemas de travamento e sugeriram outras como por exemplo, uma melhor especificação dos erros.

“Ele é bastante útil mas possui muitos bugs. Acredito que o passar do tempo e melhoramento do moesmo no que diz respeito a isso deve torná - lo uma ferramenta "perfeita".”

“ele no auxilia muito na visao layout, mas ele trava muito o espaço onde fica os erros poderia ser em português pois quem esta ultilezando e brasileiro. e nao dar para apaga por isso o programa fica muito pesado e trava muito.”

“uma avaliação dos erros do programa com relação ao sistema operacional, correção e atualização para novas funções.”

“O Composer se mostrou uma ótima ferramenta para a compreensão da linguagem NCL apesar dos vários travamentos. O design da interface é interessante, mas pode melhorar. Sugiro que as próximas versões Linux, tanto do Composer, quanto do Ganga NCL Player, sejam melhor desenvolvidas e que a instalação seja simplificada através de pacotes para as principais distribuições Linux disponíveis.”