

ESTUDO E TESTES DE USABILIDADE EM SISTEMAS DE AUTORIA DE SOFTWARE: Scratch e Alice

Allan Kássio Beckman Soares da Cruz
Universidade Federal do Maranhão
allankassio@gmail.com

Leandro Cavalcante Mendonça Lima
Universidade Federal do Maranhão
leandro.slz@gmail.com

Resumo: Com a grande quantidade de softwares disponíveis no mercado, os testes dos mesmos são cada vez mais importantes para o sucesso de uma aplicação, visto que os usuários estão cada vez mais exigentes em relação a questões anteriormente não tão importantes como, nesse caso, a qualidade das interações com a interface do sistema. Neste trabalho, são apresentados os diversos conceitos que envolvem interação humano-computador, usabilidade e softwares de autoria, dando ênfase aos testes de usabilidade, sua preparação, procedimentos, desenvolvimento e apresentação de resultados. Com o objetivo de possibilitar a preparação de um modelo padrão mais efetivo de interface e interações de softwares de autoria automatizada de conteúdo.

Palavras-chave: usabilidade, autoria, softwares de autoria, teste de usabilidade, ihc.

1 INTRODUÇÃO

O Curso de Ciência da Computação da Universidade Federal do Maranhão (UFMA) na sua formação básica possui um sub-eixo formativo cujo objetivo visa o domínio dos fundamentos e das técnicas básicas da computação, o desenvolvimento do raciocínio lógico e da habilidade de resolução de problemas, da organização e manipulação de informações; da organização e arquitetura de computadores e da utilização de técnicas e ferramentas básicas (PAIVA et al., 2007). Esse sub-eixo proporciona ao aluno apreender o conceito de programa computacional. Um programa pode ser descrito como um conjunto estruturado de instruções que capacitam uma máquina aplicar sucessivamente certas operações básicas e testes em uma parte determinada dos dados iniciais fornecidos, até que esses dados tenham se transformado numa forma desejável (MENEZES e DIVERIO, 2000). Apesar de parecer simples e objetivo, esse conceito abrange uma série infinita de combinações e de

resultados finais. Por esse motivo é tão difícil criar um programa que supra as necessidades das pessoas, conhecidas como usuários, para as quais foram projetados. Mesmo seguindo padrões, modelos, normas, metodologias das mais diversas, criar programas, ou seja, programar é uma atividade abstrata (PRESSMAN, 1995).

O desenvolvimento de programas de computador, também conhecidos como aplicativos, é uma atividade que já existe há algum tempo. Turing (1936) foi o primeiro a descrever o que viriam a ser os programas utilizados hoje em dia, contudo, no início, essa era uma atividade restrita a um pequeno grupo de profissionais e isso acontecia por diversos motivos: inexistência de interfaces, difícil acesso aos dispositivos capazes de processar instruções, conhecimento muito específico e novo etc. Entretanto, atualmente, o cenário é muito diferente. Devido à popularização dos computadores pessoais (PCs) e o surgimento e posterior massificação da internet, praticamente todas as atividades intelectuais são realizadas utilizando algum tipo de dispositivo computacional como ferramenta de auxílio, seja um computador, um tablete, ou até mesmo um dispositivo embarcado (BARBOSA e DA SILVA, 2010).

Desse modo, muitas pessoas, mesmo aqueles que não são programadores ou estudiosos do ramo de tecnologia da informação (TI), têm sentido a necessidade de criar seus próprios aplicativos para facilitar o seu trabalho e até sua vida como um todo. Direcionados para as necessidades desse público, surgiram os aplicativos chamados de programas de autoria ou softwares de autoria. Eles são empregados a fim de abstrair do autor toda, ou pelo menos parte da complexidade de se utilizar uma linguagem de programação na criação de aplicações (JUNIOR, 2011). Ou seja, são basicamente programas que são utilizados como ferramentas para auxiliar no desenvolvimento de outros programas. O objetivo principal é criar maneiras alternativas, sem a atividade de programação em si, para criar os aplicativos.

Este trabalho abrange duas visões que surgiram devido a diferentes necessidades. Uma delas é a dos usuários que precisam criar seus próprios softwares, jogos ou animações, e a outra é a dificuldade dos programadores em fornecer ferramentas que atendam esses anseios dos usuários, mas de forma simples, completa, efetiva e prática. A motivação do mesmo surgiu em função da dificuldade de mensurar a qualidade das interações dos usuários com os diferentes softwares produzidos atualmente, pois a diversidade de softwares é muito grande. Por exemplo, apenas a Apple, na sua loja de aplicativos para dispositivos móveis, possui atualmente mais de 500.000 aplicativos em suas versões finais disponíveis para os seus usuários (APPLE INC., 2012), e como são objetos abstratos, não existe uma maneira objetiva e exata que consiga mensurar se a interatividade dos mesmos é adequada ou não. Existem diversas classes ou categorias de aplicativos. A classe escolhida para ser o objeto de estudo desse trabalho foi a dos softwares de autoria, pois seu foco é justamente usuários que não têm grandes conhecimentos sobre desenvolvimento de aplicações.

O objetivo desse trabalho é realizar um estudo da interação dos usuários com diversos softwares de autoria por intermédio de testes de usabilidade. Estudou-se, desse modo, a interação dos usuários com os sistemas computacionais, as diferentes formas de aplicação dos testes de usabilidade, tal como identificar a mais efetiva para cada sistema de autoria. Apresenta-se, também, uma comparação entre os resultados dos testes.

2 INTERAÇÃO HUMANO-COMPUTADOR E USABILIDADE

Uma interface é o meio físico ou lógico através do qual um ou mais dispositivos ou sistemas incompatíveis conseguem comunicar-se entre si (FERREIRA, 2009). A interface é considerada a embalagem do software de computador, portanto se ela for fácil de aprender, simples de usar, direta e amigável, o usuário estará inclinado a fazer bom uso da mesma (PRESSMAN, 1995) e, além disso, a grande maioria dos usuários acredita que o sistema é a interface com a qual entram em contato (HIX e HARTSON, 1993). As áreas de IHC e de usabilidade se concentram no estudo dessas interfaces.

2.1 Interação Humano-Computador

O comportamento humano não é sempre igual, mas cheio de surpresas, o que dificulta o estabelecimento de simples verdades sobre o que esperar das pessoas em determinadas situações (LINDGAARD, 1994). Esse foi um dos motivos que levaram as pessoas ao estudo de IHC.

A Interação Humano-Computador (IHC) é a área da computação que investiga o design, avalia e implementa interfaces para que seres humanos possam interagir com sistemas computacionais de maneira eficiente e intuitiva (SANTOS e TEIXEIRA, 2010).

A interação dos seres humanos com os dispositivos computacionais acontece há muito tempo. O primeiro mouse, como exemplo de um dos primeiros dispositivos construídos para facilitar essa interação, foi construído em 1963 por Douglas Engelbart (EDWARDS, 2008). Mas, sem dúvida, essas interações evoluíram muito. Os tablets podem ser citados como uma dessas evoluções.

Sabe-se que os computadores pessoais atuais apenas conseguem computar a linguagem binária, então cada estímulo, cada interação com os mesmos não será nada além de uma forma de transmitir zeros e uns de modo que possam ser processados (PRESSMAN, 1995). Uma das primeiras formas de interação foi o uso de cartões perfurados, em que os programas eram escritos em cartões e inseridos no mainframe para processamento. Com o tempo surgiram novas formas de interação: teclado e mouse; toque, através de telas sensíveis; voz, através de microfones que captam e reconhecem comandos em fala natural; imagem, através de câmeras e sensores que captam movimentos (GADELHA, 2009).

Todas essas inovações e evoluções visam melhorar as formas de interações, facilitar as mesmas, tentar simular interações do mundo real através do uso de computadores e máquinas. Como os estudos em usabilidade e IHC são, de certa forma, recentes se comparados com o surgimento dos primeiros PCs, estas áreas do conhecimento ainda têm muitos desafios e o desenvolvedor deve ter consciência que o resultado do seu trabalho vai influenciar e modificar a vida de um grande número de pessoas das mais diversas formas.

2.2 Usabilidade

A usabilidade de um sistema está relacionada com a experiência que os usuários têm com o mesmo, sua interação com a interface (FERREIRA, 2002).

Deve ser um processo interdisciplinar e colaborativo (BOTELLA et al., 2011). Sendo assim, é importante que todos os envolvidos no processo de criação das soluções tenham consciência dos seus princípios básicos e estejam atentos para o mesmo durante toda a fase de desenvolvimento.

Segundo a norma ISO/IEC 9126 (1991), que define critérios de qualidade de software, usabilidade é “um conjunto de atributos relacionados com o esforço necessário para o uso de um sistema interativo, e relacionado com a avaliação individual de tal uso, por um conjunto específico de usuários”.

Já a norma ISO 9241-11 (1998), que trata de requisitos de ergonomia, define usabilidade como “o grau em que um produto é usado por usuários específicos para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto de uso específico”. Essas duas definições são bem amplas, mas pode-se, contudo, já ter uma noção da abordagem orientada ao produto e ao usuário do mesmo, em comparação com a abordagem tradicional focada no desenvolvimento. Para Nielsen (1993), o critério de usabilidade é um conjunto de fatores que qualificam quão bem uma pessoa pode interagir com um sistema interativo.

Esses critérios estão relacionados com a facilidade e o esforço necessários para os usuários aprenderem e utilizarem um sistema. Desse modo, a usabilidade endereça principalmente a capacidade cognitiva, perceptiva e motora dos usuários empregada durante a interação. Os fatores de usabilidade por ele considerados são:

Facilidade de aprendizado – está relacionado com o tempo e esforço que o usuário necessita para poder ter determinado nível de performance e desempenho no uso do sistema. *Facilidade de recordação (no caso de uso intermitente)* – está relacionado com quantidade de esforço que o usuário precisa ter para lembrar de determinada ação já previamente utilizada. *Eficiência ou produtividade* – está relacionada com a performance, ou seja, o tempo que o usuário necessita para realizar determinada tarefa com o uso do sistema. *Segurança de uso* – está relacionada com a eficácia do sistema em proteger o usuário de situações que possam prejudicá-lo acidentalmente. *Satisfação do usuário* – está relacionada com uma avaliação subjetiva dos usuários em relação à utilização do sistema.

Caso os critérios de usabilidade não sejam satisfeitos, há grandes chances de haver problemas de usabilidade. Eles ocorrem quando um usuário ou um grupo de usuários encontra dificuldades para realizar uma tarefa com uma interface. Tais dificuldades podem ter origens variadas e ocasionar perda de dados, diminuição da produtividade e podem chegar até mesmo à rejeição total do software por parte dos usuários (FERREIRA, 2002).

Desenvolver sistemas com boa usabilidade significa focar o desenvolvimento nos usuários. Para tanto, deve-se observar suas características, seu contexto de utilização e o seu objetivo com a utilização do sistema.

3 SOFTWARES DE AUTORIA

A forma convencional de desenvolvimento de aplicativos é baseada no uso de linguagens de programação tradicionais, a grande vantagem dessa abordagem é a liberdade dos autores para criar o software. Dessa forma, é possível, por exemplo, organizar o código conforme sua vontade, agrupar determinados trechos, especificar os tipos abstratos de dados e os tipos de interações a serem utilizados (RODDY, 1996). Como desvantagens, podem ser citadas a maior curva de aprendizado necessária para conseguir um bom resultado final, as preocupações com erros sintáticos, com as particularidades da linguagem utilizada e a dificuldade em criar abstrações que representem objetos do mundo real.

Os softwares de autoria surgiram para tentar diminuir as desvantagens do método tradicional de desenvolvimento. Segundo Junior (2011), softwares de autoria são aqueles que visam desenvolver a criatividade do usuário que trabalha como autor de uma obra. A produção pode ser tanto a exposição de dados quanto a construção do conhecimento, dependendo das orientações do usuário. Com esse tipo de software os próprios usuários podem desenvolver suas aplicações sem que para isso precisem entender de programação de computadores.

Basicamente, um software de autoria é um programa que possibilita às pessoas que possuem pouco ou nenhum conhecimento em linguagens de programação e desenvolvimento de aplicativos poder criar aplicativos, animações, e-books ou apresentações. Esse tipo de software normalmente encapsula uma linguagem de programação (é importante que mesmo que não seja utilizada pelos usuários, a linguagem de programação tradicional esteja presente no mesmo, pois, dessa forma, permite que os usuários avançados possam fazer ajustes finos na implementação) e um compilador (que transforma um programa escrito numa linguagem fonte para um programa escrito em uma linguagem objeto) (AHO et al., 1995), permitindo que ao final do processo o usuário tenha um arquivo executável que realize as tarefas para as quais o software foi criado, de acordo com a vontade e a necessidade do usuário.

Devido ao grande número de softwares dessa categoria, esse trabalho tem foco em apenas dois. São eles: Scratch e Alice.

3.1 Scratch

Scratch é um ambiente de desenvolvimento que utiliza uma linguagem de programação baseada em blocos, projetada para ser intuitiva e fácil de ser utilizada por desenvolvedores iniciantes (CONWA et al., 1999). Scratch adiciona programação nas atividades de mídia rica e baseada em rede que são populares entre os jovens nos centros de ensino. Scratch tira proveito do extraordinário poder de processamento dos computadores atuais, suporta novos paradigmas de programação e atividades que antes eram inviáveis, tornando ele uma melhor opção que as tentativas anteriores de introduzir a programação aos usuários iniciantes (MALONEY et al., 2004).

Os desenvolvedores do Scratch tiveram a iniciativa de desenvolvê-lo devido à dificuldade de introduzir a programação para leigos ou jovens iniciantes, utilizando as linguagens de programação tradicionais. Essa foi a motivação para tentar expandir o acesso a esse tipo de atividade a grupos muito maiores de pessoas. Um fator que possibilitou essa ideia foi o fato de os computadores atuais terem grande poder de processamento, fato que inviabilizava o projeto anteriormente.

Scratch é baseado em uma estrutura de programação utilizando construção de blocos através do recurso de arrastar e soltar, que gera scripts ao colocar blocos gráficos unidos, como se o usuário estivesse montando um quebra cabeça (MALONEY et al., 2004). Essa abordagem tende a eliminar erros de sintaxe, um dos maiores empecilhos nas linguagens de programação baseadas em texto, permitindo que o usuário foque no problema que ele deseja resolver. Existem diversas classes diferentes de blocos, cada classe reúne blocos com temática relacionada, como blocos de movimento, som, aparência etc.

3.2 Alice

Alice é um ambiente de programação 3D que torna fácil criar uma animação para contar uma história, jogar um jogo interativo ou compartilhar um vídeo na web. Ela permite que os usuários aprendam conceitos fundamentais da programação tradicional no contexto de criar vídeos animados ou jogos simples. Em Alice, os objetos 3D fazem parte de um mundo virtual no qual os usuários criam programas para animar esses objetos (ALICE, 2012). Segundo Conwa et al. (1999), o Projeto Alice foi iniciado com o objetivo de criar novas ferramentas que possibilitassem criar gráficos 3D de maneira mais acessível, de forma que um público maior pudesse ter acesso aos mesmos. Os princípios básicos foram: escolher um público alvo e manter as suas necessidades em foco; no caso de Alice, estudantes de graduação de áreas não técnicas; evitar notações matemáticas e enigmáticas na API (como, por exemplo, vetores e matrizes) sempre que possível e introduzir novas terminologias somente quando necessário; fazer testes interativos do design com usuários reais, melhorando ambos - aprendizado e usabilidade do processo do sistema.

Assim como o Scratch, o Alice também é um software de autoria que se baseia em um formato de montar blocos e pilhas de códigos, mostrando sempre uma resposta visual do que determinado bloco faz, permitindo aos usuários relacionar a programação com os seus resultados. Os seus desenvolvedores afirmam que esse tipo de resposta rápida e eficiente tem uma grande porcentagem na atração de interesse pelo sistema. Seus testes mostram uma diminuição considerável na evasão de cursos de programação, além de melhorar o desempenho dos estudantes.

4 AVALIAÇÃO DE IHC E TESTES DE USABILIDADE

Conhecer, e seguir os critérios de qualidade e os diferentes processos de fabricação de um produto não garante que o resultado possua qualidade. Diversos problemas podem ocorrer, por exemplo alguma falha humana no processo, a baixa qualidade da matéria prima, etc. De certa forma essa linha de raciocínio também é válida para o desenvolvimento de softwares, onde os problemas costumam ocorrer nas etapas de coleta dos dados, interpretação ou até mesmo na implementação (BARBOSA e DA SILVA, 2010).

O objetivo de uma avaliação de IHC é avaliar a solução do ponto de vista dos usuários, a melhor forma de fazer isso é justamente envolvê-los nesse processo. Outra vantagem do teste de usabilidade é a possibilidade de fornecer mais dados quantitativos em relação aos demais métodos, fator importante para a comparação com os demais softwares de autoria.

Para conseguir resultados mais específicos: ao agrupar os tipos de usuários semelhantes, é mais fácil analisar e tirar conclusões dos testes; e imparciais: ao utilizar representantes de usuários reais as interferências no processo diminuem (COOPER, 1999). Nesse estudo, foram feitos testes com 3 grupos de usuários. Cada grupo será representado por uma Persona. Onde a Persona do tipo 1 é um estudante do 7º período de Ciência da Computação, já trabalhou como técnico de suporte e agora é estagiário no setor de desenvolvimento de softwares de uma empresa multinacional. Tem plenos conhecimentos em programação, utiliza e domina diversas linguagens, além disso, tem conhecimentos aprofundados nas áreas de redes e banco de dados. A Persona do tipo 2 é estudante do 8º período de Desenho Industrial, já participou de um projeto de criação de websites para diversos eventos que acontecem na sua cidade. Atualmente trabalha em uma empresa de publicidade, é responsável por

coordenar a equipe de design. Tem conhecimento razoável na área de computação, no entanto não sabe programar. Utiliza o computador no trabalho para checar e-mails, fazer pesquisas, agendar reuniões, planejar ações, etc. Já em casa e na universidade, seu uso é para lazer e para realizar as atividades acadêmicas. Utiliza, em média, o computador por 3h/dia. E a Persona do tipo 3 é formado em administração, é funcionário público de uma secretaria da sua cidade. Exerce a função de auxiliar administrativo há 10 anos. Tem pouco ou nenhum conhecimento na área de computação, utiliza apenas funções básicas do seu computador, mesmo assim, tenta evitar o máximo possível, utiliza o computador, em média, 6h/semana apenas para questões relacionadas ao trabalho, normalmente responder e-mails ou apenas para imprimir algum documento. Os testes foram realizados por 6 personas de cada tipo, cada uma em determinada ordem diferente de utilização dos 2 aplicativos. Essa foi a forma utilizada para garantir que mesmo que a utilização de um aplicativo pudesse influenciar na experiência de uso do seguinte, haveriam pelo menos 2 pessoas que testaram cada aplicativo sem influência de testes anteriores, além disso, dessa forma foi possível realizar uma quantidade razoável de testes sem a necessidade de muitas pessoas diferentes envolvidas.

A coleta de dados consiste no questionário pré-teste, a sessão de observação e a entrevista pós-teste. Essa coleta pode ser considerada o teste de usabilidade em si, e acontece da seguinte forma: primeiramente é feita uma apresentação do teste. Em seguida, os usuários respondem um questionário a fim de serem identificados e agrupados conforme suas características. Posteriormente, o mesmo utiliza o sistema por um período de tempo, normalmente entre 15 e 20 minutos. Durante essa utilização, o avaliador faz a coleta de dados principais. E, por fim, é feito um questionário final para saber a opinião dos usuários sobre o sistema testado.

Na atividade de interpretação ou consolidação, os dados são organizados de forma a evidenciar a relação entre os mesmos, visando chegar às conclusões que objetivaram os testes. Esses dados serão apresentados na forma de gráficos, tabelas, médias e quaisquer outros indicadores relevantes para esse estudo. Nessa fase, segundo Kuniavsky (2003), os testes de usabilidade também fornecem dados qualitativos, em que é possível identificar a origem, o motivo dos problemas de usabilidade, de interação de cada usuário. Cabe ao observador e analisador verificar todos os dados coletados para conseguir chegar a explicações para os problemas que tenham surgido.

4.1 Objetivo dos testes

O objetivo dos testes foi avaliar as seguintes metas de usabilidade: Usos da ajuda do aplicativo e a qualidade da mesma; Tempo necessário para obter o primeiro resultado final; Chegar ao resultado com menos tentativas fracassadas, sem ajustes; Quantidade de funções do aplicativo utilizadas; Quantidade de erros encontrados; Uso da ferramenta desfazer, voltar; Tempo de realização de determinada tarefa. Esses objetivos foram detalhados em perguntas de forma que ficassem operacionais conforme indica Sharp et al. (2007).

5 RESULTADOS

Durante os testes, ocorreram 5 desistências, elas aconteceram por volta de 5 minutos do início dos mesmos e se deram conforme o Quadro 1.

Quadro 1 – Desistências conforme o tipo das Personas.

Sistema	Persona 1	Persona 2	Persona 3	Total
Scratch	0	0	0	0
Alice	0	2	3	5
Total	0	2	3	

Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Tais resultados se devem ao fato da maior dificuldade das Personas do tipo 3 em utilizar aplicativos diferentes dos quais estavam habituados e à maior complexidade de utilização do Alice.

Quadro 2 – Quantidade média de erros por avaliado.

Sistema	Persona 1	Persona 2	Persona 3	Total
Scratch	0,33	0,50	0,50	1,33
Alice	0,17	0,00	0,33	0,50
Total	0,50	0,50	0,83	

Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Foram considerados erros, sempre que um usuário tentou realizar uma função não permitida pelo software ou quando o mesmo conseguiu finalizar a interação mas a mesma não produzia o efeito desejado, a quantidade de erros pode ser observada no Quadro 2, que mostra a média de erros por avaliado para cada grupo de personas e para cada software. Os principais erros foram no Alice, a tentativa de voltar para a tela do aplicativo sem antes fechar a janela de saída e no Scratch a tentativa de conectar bloco incompatíveis. Os motivos de as Personas 1 terem mais erros é reflexo da utilização e exploração de mais e mais complexas funcionalidades em comparação com as demais. Em relação ao uso de ajuda dos aplicativos, foram coletadas duas informações, o tempo aproximado médio de utilização de ajuda Quadro 3 e a quantidade de avaliados que utilizou esse recurso Quadro 4.

Quadro 3 – Tempo médio de utilização de ajuda (em minutos).

Sistema	Persona 1	Persona 2	Persona 3	Total
Scratch	0	5	1,7	3,4
Alice	0	3,3	3,3	3,3
Total	0	4,2	2,5	

Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Quadro 4 – Quantidade de avaliados que utilizou os menus de ajuda.

Sistema	Persona 1	Persona 2	Persona 3	Total
Scratch	0	4	1	5
Alice	0	3	2	5
Total	0	7	3	

Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Essas informações mostram que as personas 1 não tem o costume de utilizar a ajuda de aplicativos, não importando qual seja. Já as personas do tipo 2 foram as que mais utilizaram e por mais tempo. Porém os menus de ajuda dos aplicativos não foram muito efetivos, provavelmente este foi o motivo de terem sido pouco utilizados, o Alice por exemplo, sua ajuda é constituída de diversos tutoriais práticos. O Scratch é o único que mostra as ferramentas e suas funcionalidades uma a uma. Uma crítica em relação à ajuda do Scratch é que apesar de o software ser o único com menus em português, sua ajuda está disponível somente em inglês.

A utilização do comando desfazer, que pode ser observada no Quadro 5, foi maior com as personas 1, isso se deu ao fato de as mesmas terem explorado mais os aplicativos, utilizando funcionalidade mais complexas. A quantidade maior de utilizações no Alice em relação ao Scratch é justificável por diversos fatores: o Alice possui um botão na interface principal com essa funcionalidade e os comandos “Ctrl+z” funcionam nele; já no Scratch, quando os usuários sentiam a necessidade de desfazer o software apenas possibilita desfazer o último comando de apagar, mas mesmo se fosse diferente os usuários ainda usariam pouco, pois no Scratch é bem simples apagar o que você fez e fazer novamente.

Quadro 5 – Quantidade de utilizações da função "desfazer".

Sistema	Persona 1	Persona 2	Persona 3	Total
Scratch	2	1	0	3
Alice	14	7	4	25
Total	16	8	4	28

Fonte: Elaborado pelo autor, com base na pesquisa realizada.

A métrica seguinte, ilustrada no Quadro 6, mostra o tempo que os usuários necessitaram para criar o primeiro resultado, foi considerado resultado qualquer saída do aplicativo com algum objeto ou interação. Com esses dados fica claro que as personas do tipo 1 tiveram mais facilidades para criar resultados nos 2 aplicativos testados. No caso das personas 3, os avaliados não conseguiram nenhum resultado, por isso a representação com média de 20 minutos, que foi o tempo total de cada testes. Essas informações estão relacionadas com a eficiência e efetividade na utilização do software. Ou seja, pode-se dizer que as personas do tipo 1 foram as mais eficientes e o software mais efetivo foi o Scratch.

Quadro 6 – Tempo médio em minutos para cada avaliado conseguir o primeiro resultado.

Sistema	Persona 1	Persona 2	Persona 3	Total
Scratch	2	4	4	3,3
Alice	4	5	8	5,6
Total	3	4,5	6	

Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Em relação à nota que os avaliados atribuíram aos softwares por suas funcionalidades, as personas 1 foram as que atribuíram as maiores notas, e o Scratch foi o software que mais agradou os usuários. Essas informações, mostradas no Quadro 7, estão

relacionadas com a satisfação dos usuários ao utilizar os softwares, ou seja, se os mesmos se sentiram à vontade, e a utilização foi agradável.

Quadro 7 – Nota média atribuída pelos usuários aos softwares.

Sistema	Persona 1	Persona 2	Persona 3	Total
Scratch	8,4	7,9	7,5	7,9
Alice	7,9	4,2	4,2	5,4
Total	8,2	6,0	5,8	

Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Para a nota de usabilidade ou facilidade de uso, conforme o Quadro 8, o comportamento foi parecido com o anterior, contudo nesse quesito as personas do tipo 2 foram as que atribuíram as menores notas no geral. Esse comportamento provavelmente se deve ao costume em utilizar softwares mais intuitivos que os referidos neste trabalho.

Quadro 8 – Nota média atribuída pelos usuários aos softwares.

Sistema	Persona 1	Persona 2	Persona 3	Total
Scratch	8,8	6,0	8,5	7,8
Alice	7,3	2,7	2,2	4,1
Total	8,0	4,4	5,4	

Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Ao serem questionados acerca do sentimento em relação à utilização dos softwares os avaliados responderam conforme o Quadro 9 que mostra a quantidade de usuários que elogiaram ou criticaram os sistemas. Curiosamente mesmo após dar notas baixas para o software e para sua usabilidade alguns usuários responderam como satisfeitos nesse quesito.

Quadro 9 – Classificação dos usuários em satisfeitos ou frustrados.

Scratch	Persona 1	Persona 2	Persona 3
Satisfeito	6	4	5
Frustrado	0	2	1
Alice	Persona 1	Persona 2	Persona 3
Satisfeito	4	2	0
Frustrado	2	4	6

Fonte: Elaborado pelo autor, com base na pesquisa realizada.

Além dos diversos dados quantitativos, os testes de usabilidade também proporcionaram diversos dados qualitativos da utilização dos sistemas, esses dados foram coletados juntamente com os demais, porém não existe uma maneira precisa de medi-los. Eles foram observados durante os testes e registrados pelo avaliador, para exemplificar alguns serão usadas ilustrações e frases ou gestos que os avaliados apresentaram durante os testes. Esses dados serão expostos a seguir.

Devido ao fato de os sistemas serem de certa forma simples, com poucas telas, praticamente todos os usuários utilizaram os mesmos da forma esperada, as personas

do tipo 1, sempre buscaram utilizar alguma interação a mais, através do uso de laços para repetição de comandos por exemplo. Já as personas do tipo 2 se preocuparam com a estética do resultado final, editando manualmente os objetos pré-definidos nos programas por exemplo. As personas do tipo 3 ficaram localizadas em um meio termo entre os dois outros tipos.

Os maioria dos usuários não teve dificuldades em explorar a maior parte das funcionalidades dos softwares, o que apresentou mais problemas nesse sentido foi o Alice devido à grande quantidade de informações e configurações disponíveis simultaneamente. Durante os testes, não ocorreu falhas de comunicação entre os sistemas e os usuários, ou seja, os usuários conseguiram executar as tarefas sem se perder, e conscientes dos passos seguintes. Os usuários conseguiram compreender a grande maioria dos elementos da interface do Scratch, já no que diz respeito ao Alice houve dificuldades em alguns pontos. No Alice, alguns usuários tiveram dificuldades para entender deveriam arrastar os itens dos menus para a área de trabalho.

6 CONCLUSÃO

O estudo das interações humano-computador e da usabilidade dos softwares são de grande importância para o desenvolvimento de aplicações de sucesso em um mercado que está cada vez mais concorrido.

Para avaliar a qualidade dessas interações os testes de usabilidade cumprem seu papel possibilitando aos desenvolvedores e designers tirar grandes conclusões a respeito dos sistemas avaliados, além de proporcionar ao mesmo tempo ideias e soluções para os possíveis problemas detectados.

A aplicação desses testes nos três diferentes softwares de autoria permitiu fazer uma comparação eficiente entre os mesmos, na qual foram identificados diversos problemas de usabilidade em cada um como por exemplo quantidade de erros durante a realização de determinada tarefa, e também seus pontos fortes, partes intuitivas das interfaces. Todos esses dados foram coletados com a utilização de entrevistas, formulários e dos testes de usabilidade em si.

A organização dessas informações permitiu que fosse criada uma base de dados que poderá ser utilizada para a construção de um modelo para os softwares de autoria automatizada de conteúdo. O resultado deste trabalho foi positivo e serve para mostrar que as questões de usabilidade devem ser levadas em consideração no desenvolvimento de qualquer sistema. A satisfação e a produtividade dos usuários aumenta e, dessa forma, o software e seus desenvolvedores passam a ser vistos como provedores de um serviço de maior qualidade.

Como trabalhos futuros pode-se citar o desenvolvimento de um modelo de interface, construção de uma ferramenta de autoria baseada nesse modelo e também a realização de testes de usabilidade em outras classes de softwares, como softwares de edição de imagem, edição de vídeo, IDEs, etc.

REFERÊNCIAS

AHO, A.V.; SETHI, R.; ULLMAN, J.D. **Compiladores: princípios, técnicas e ferramentas**. Rio de Janeiro/RJ: LTC, 1995.

ALICE. **What is Alice?** Disponível em http://alice.org/index.php?page=what_is_alice/what_is_alice, 2012. Acesso em 17/11/2013.

APPLE INC. **The App Store**. Disponível em <http://www.apple.com/iphone/built-in-apps/app-store.html>. 2012. Acesso em 17/11/2013.

BARBOSA, Simone D. J.; DA SILVA, Bruno S. **Interação Humano-Computador**. Rio de Janeiro: Elsevier, 2010.

BOTELLA, Federico; GALLUD, Jose A.; TESOREIRO, Ricardo. **Using Interaction Patterns in Heuristic Evaluation**. 2011.

CONWA, Matthew. et al. **Alice: Lessons Learned from Building a 3D System For Novices**. CHI. 1999.

COOPER, A. **The Inmates Are Running the Asylum: why High-Tech Products Drive Us Crazy and How to Restore Sanity**. Sams Publishing, 1999.

EDWARDS, Benj. **The Computer Mouse Turns 40**. 2008. Disponível em <http://www.macworld.com/article/1137400/mouse40.html> Acesso em 21/10/2013.

FERREIRA, Aurélio B. de H., **Novo dicionário Aurélio da língua portuguesa**. São Paulo: Editora Positivo, 2009.

FERREIRA, Kátia G. **Teste de Usabilidade**, Monografia de Final de Curso: Especialização em Informática, Universidade de Minas Gerais. Belo Horizonte, Brasil, 2002.

GADELHA, Julia. **A Evolução dos Computadores**. 2009. Disponível em <http://www.ic.uff.br/~aconci/evolucao.html> Acesso em 21/10/2013.

HIX, G.; HARTSON, H. **Developing User Interfaces: Ensuring Usability Through Product and Process**. New York, NY: John Wiley & Sons, 1993.

ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VTDs) Part 11: Guidance on Usability. ISO, 1998.

ISO/IEC 9126: Software Engineering – Product Quality. ISO, 1991.

JUNIOR, Vital P. D. S. **Software educacional de autoria e a metodologia Webquest**. 2011.

KUNIAVSKY, M. **Observing the User Experience: A Practitioner's Guide to User Research**. San Francisco, CA: Morgan Kaufmann Publishers, 2003.

LINDGAARD, Gitte. **Usability Testing and System Evaluation. A guide for designing useful computer systems**. New York: Chapman & Hall, 1994.

MALONEY, J. et al., **Scratch: A Sneak Preview**. Second International Conference on Creating, Connecting, and Collaborating through Computing. Kyoto, Japan. 2004.

MENEZES, P. B.; DIVERIO, T. A. **Teoria da Computação**. Sagra-Luzatto, 2000.

NIELSEN, Jakob B. **Interactive User-Interface Design**. Computer, IEEE. 1993.

PAIVA, Anselmo C. De, et al., **Projeto Pedagógico do Curso De Ciência Da Computação**. Disponível em http://www.deinf.ufma.br/cocom/files/arquivos/ProjPedagogico_CP_2007_final.pdf Acesso em 21/10/2013.

PRESSMAN, Roger S. **Engenharia de Software**. 3 ed. São Paulo: Makron Books, 1995.

RODDY, Brian; MARKOWITZ, Sidney; WANG-EPELMAN, Hernán. **User Interfaces for Authoring Systems with Object Stores**. IEEE Proceedings of COMPCON. 1996.

SANTOS, Sérgio L.; TEIXEIRA, F.G. **Design de uma Interface de Interação Tridimensional com Foco na Usabilidade e no Desempenho Gráfico**. PgDesign. Porto Alegre, 2010.

SHARP, H.; ROGERS, Y.; PREECE, J. **Interaction design: beyond human-computer interaction**, 2ª edição. New York, NY: John Wiley & Sons, 2007.

TURING, A. M., **On Computable Numbers, With An Application To The Entscheidungs problem**, 1936. Disponível em http://www.cs.virginia.edu/~robins/Turing_Paper_1936.pdf Acesso em 03/01/2014.