
Proposta de metodologia para o desenvolvimento de objetos de aprendizagem de autoria própria com *scratch*

Proposal of methodology for the development of self-learning learning objects with *scratch*

Luiz Carlos Aires Macedo

Universidade Federal Rural do Semi-Árido
luizcarlos@ufersa.edu.br

.....

Maria Ivete Basniak

Universidade Estadual do Paraná
basniak2000@yahoo.com.br

.....

Marcos Cesar Danhoni Neves

Universidade Estadual de Maringá
macedane@yahoo.com

.....

Sani de Carvalho Rutz da Silva

Universidade Tecnológica Federal do Paraná
sanirutz@gmail.com

.....

Débora Amaral Taveira Mello

Universidade Tecnológica Federal do Paraná
datmello@gmail.com

Resumo

Este artigo apresenta uma proposta de metodologia voltada à produção de Objetos de Aprendizagem (OA) de autoria própria para educadores sem conhecimentos aprofundados em modelagem e programação de softwares. Esta metodologia utiliza *storyboards* como estratégia de modelagem e programação adaptada à linguagem *Scratch*, se configurando como fácil e eficiente de ser utilizada. Para discuti-la é apresentado um exemplo que salienta as características desta metodologia em sua aplicação prática. Conclui-se que a metodologia pode ser interessante para educadores que desejam construir seus próprios OA utilizando a linguagem *Scratch* sem que isso signifique o não uso de um processo de desenvolvimento de software.

Palavras-chave: Software educativo. Multimídia interativo. Tecnologia e didática. Material autodidático. Tecnologia e educação.

Abstract

This article presents a proposal for a methodology aimed at the production of Learning Objects (LO) of its own authorship for educators with no depth knowledge in software modelling and programming. This methodology uses storyboards as a modelling and programming strategy adapted to the Scratch language, setting itself up as easy and efficient to use. To discuss it, an example is presented that highlights the characteristics of this methodology in its practical application. It is concluded that the methodology may be interesting for educators who wish to build their own LO using the Scratch language, without implying the non-use of a software development process.

Key words: Educational software. Interactive multimedia. Technology and didactics. Self-taught material. Technology and education.

Introdução

Atualmente as Tecnologias Digitais (TD) estão presentes em escolas de todo país, possibilitando que os professores as integrem em sua prática de sala de aula.

Sobre possíveis usos das TD em sala de aula, Valente (1997) aponta que o computador pode ser utilizado de forma positiva no reforço do processo de ensino instrucionista, quando há a transmissão da informação, e por outro lado, de forma construcionista, quando se constitui ambiente propício para o aluno construir seu conhecimento. Quanto à forma instrucionista, podemos citar a apresentação de *slides* com conceitos a serem expostos em uma aula dialogada, contrapondo-se a construcionista, que pressupõe interações entre os alunos e o objeto a conhecer, como por exemplo, em um jogo ou simulação.

Contudo, o processo de incorporar as tecnologias em sala de aula não é um processo simples, pois apesar da grande variedade de softwares hoje disponíveis, há situações em que nenhum destes contempla o objetivo que o professor quer atingir. Para Braga et al. (2012), além do uso das TD como ferramentas de auxílio nas aulas, estas poderiam também ser apropriadas pelos educadores para elaborar conteúdos didáticos com estas tecnologias.

Muitos professores ainda relutam em usar as tecnologias em sala de aula ou ainda que as utilizem o fazem apenas para reproduzir suas antigas práticas. É o caso do professor que deixa de usar a lousa e o pincel para usar uma apresentação em *slides* de determinado assunto sem agregar as possíveis vantagens do uso das TD, impossibilitando uma aprendizagem mais significativa (RODRIGUES; STANO, 2016).

No sentido da perspectiva de construção de materiais didáticos, destacamos o papel dos Objetos de Aprendizagem (OA) como alternativas computacionais para o ensino que podem se originar de qualquer situação em que o professor

deseje trabalhar com recursos das TD com seus alunos. E para que os professores possam construir seus próprios materiais didáticos de apoio ao ensino com o uso das TD é necessário que eles conheçam um pouco sobre softwares e programação. Dada a complexidade da programação para os leigos no assunto, alguns softwares apresentam opções que facilitam esta tarefa. Entre os quais podemos citar o *Scratch*, um programa que permite criar OA com recursos de multimídia que favorecem interações com os usuários por meio de uma linguagem de programação simples com vários recursos computacionais de fácil compreensão. Então, neste artigo apresentamos uma metodologia de desenvolvimento de OA possível de ser utilizada inclusive por aqueles que não possuem conhecimentos específicos em desenvolvimento e programação de softwares.

Iniciamos apresentando os conceitos e características dos OA e da linguagem *Scratch* e suas características. Na sequência discutimos modelos de desenvolvimento de OA e, em seguida apresentamos um modelo de desenvolvimento que pode ser utilizado para a criação de OA de autoria própria validamos o modelo por meio de um breve relato de estudo de caso sobre o desenvolvimento da metodologia proposta que suportam nossas conclusões.

Objetos de Aprendizagem (OA)

Um objeto de aprendizagem pode ser entendido como qualquer recurso, como por exemplo, textos, imagens, filmes, jogos, sons dentre outros objetos computacionais, de forma isolada ou em conjunto, que podem ser usados tendo como fim a aprendizagem (BEHAR et al., 2009).

Para Falkembach (2005, p.02) “as expressões: software educacional; *courseware*; aplicativos educacionais; material educativo; material educacional; ferramentas instrucionais; material didático na forma eletrônica e material didático digital são sinônimos”. A autora destaca que a expressão objetos educacionais também significa material educativo digital e que estes devem ter como característica a possibilidade de reutilização e compartilhamento do mesmo.

Um OA pode ser do tipo interativo ou estático. No estático, não há interação do aluno de forma direta, como em textos digitais estáticos com ou sem recursos de multimídia. Já no interativo, a interação entre o objeto e o aprendiz é fundamental. Como exemplos de OA interativos, temos os questionários, simuladores e jogos com fins educativos. Uma classificação mais detalhada dos OA quanto aos tipos pode ser encontrada em Battistella et al. (2009).

Embora a criação de softwares para educação tenha se iniciado já no século passado, conforme aponta o trabalho de Valente (1993), a produção dos OA é um fenômeno recente que surgiu com a expansão das TD nas escolas e com os investimentos do Ministério da Educação (MEC) no incentivo aos usos das TD em sala de aula. Também podemos relatar a recente expansão dos cursos de Educação superior a Distância a partir das Instituições de Ensino Superior tradicionais, com a criação dos núcleos de ensino a distância, que utilizam as TD

como meio de ensino e aprendizagem. À medida que problemas com o ensino foram sendo percebidos estes núcleos de ensino passaram a investir em equipes para a criação de OA, visando diminuir as dificuldades características ao ensino a distância por meio das TD. O trabalho de Nike, Carneiro e Passos (2011) destaca o papel dos núcleos pedagógicos para produção de materiais digitais para Educação a Distância (EaD) e apresenta um pouco da complexidade que a criação destes objetos apresenta.

A multimídia é uma realidade nas TD e quando esta é associada à educação, temos a possibilidade de criação de materiais digitais de apoio didático ao ensino e aprendizado, permitindo ao professor uma melhor forma de expor os conteúdos, facilitando a construção do conhecimento, unindo a tecnologia com a educação (FALKEMBACH, 2005). Ávila et al. (2014) destacam que os recursos de multimídias como sons, combinando imagens e interatividade têm sido pouco a pouco incorporados ao ensino.

A multimídia em um OA é importante especialmente pelo potencial de prender a atenção do aluno e proporcionar maior riqueza de detalhes para o ensino. Um conjunto de recomendações sobre o visual de um OA pode ser encontrado em Grandó, Konrath e Tarouco (2003). Uma simulação envolve a criação de modelos dinâmicos e simplificados do mundo real, podendo conter elementos multimídia e ser entendido como uma simplificação de uma situação real. Ao propor um modelo, o professor poderá explorar situações fictícias como experiências diversas por meio de simulação de uma situação desejada. O aluno pode interagir, desenvolver hipóteses, analisar resultados e observar conceitos a serem apreendidos (VALENTE, 1993).

Um OA é uma aplicação computacional que equilibra aspectos técnicos e pedagógicos e pode ser reutilizado contribuindo de maneira efetiva para o aprendizado, como apontam Braga et al. (2012). Os autores destacam que um OA com deficiência pedagógica pode comprometer o aprendizado do conteúdo e no caso de um OA com deficiência técnica de produção, este pode desmotivar o aluno quanto ao seu uso.

É comum encontrarmos OA prontos para uso em repositórios, que são páginas da internet que armazenam e distribuem estes objetos. Existem diversos repositórios para os OA como o *Latin American Federation of Learning Object Repositories* (LAFLO - <<http://laflor.laclo.org>>) com mais de 50 mil OA, e o repositório brasileiro Banco Internacional de Objetos Educacionais (BIOE - <objetoseducacionais2.mec.gov.br>) com quase 20 mil OA disponíveis. Nestes espaços é possível qualquer educador pesquisar e usar OA para fins didáticos. Apesar desta disponibilidade de OA, existem situações específicas em que o educador pode não encontrar um material adequado disponível por não atender às necessidades pedagógicas desejadas.

Neste sentido, Braga et al. (2012) apontam algumas dificuldades enfrentadas quando buscamos um OA em repositórios, como: de recuperação, que se refere a dificuldade de encontrar um OA específico nos repositórios; de instalação, alguns OA não são compatíveis com os dispositivos desejados; usabilidade, OA que são de difícil manipulação por parte dos professores e dos alunos; avaliação pedagógica, quando não foram avaliados e não permitem a avaliação;

imprecisão, que se refere ao fato de muitos OA serem imprecisos quanto aos resultados; e baixa confiabilidade, quando os OA possuem algum defeito ou abordam algum conceito de forma errada.

Linguagem *Scratch*

O *Scratch* é um projeto do *Lifelong Kindergarten Group* do *MIT Media Lab* que consiste em um ambiente gratuito disponibilizado no qual é possível programar por meio de combinações de blocos lógicos de comandos. Permite criar aplicações de multimídia envolvendo textos, sons e imagens para compor narrativas interativas, jogos, animações dentre outros. Além da criação, possibilita também o compartilhamento das construções com outras pessoas, assim como buscar construções desenvolvidas por outros usuários por meio dos sites do *Scratch* - <<https://scratch.mit.edu>> e *Scratch* Brasil - <<http://www.scratchbrasil.net.br>>. Estes sites oferecem tutoriais da linguagem, fóruns de discussão, atualizações e notícias, sendo um ambiente de interação para pessoas que usam o *Scratch*. O programa pode ser baixado gratuitamente ou utilizado na plataforma on-line do próprio site. É uma ferramenta de autoria multimídia que permite a criação de apresentações multimídias interativas, jogos, vídeos, simulações dentre várias outras possibilidades. O *Scratch* pode ser usado como uma ferramenta para o aprendizado construcionista e segundo Bastos, Borges e D'Abreu (2010) e Passos (2015) é uma ferramenta capaz de alterar a forma de trabalho do professor ao proporcionar opções para trabalhar com os alunos de maneira interativa e criativa.

O *Scratch* proporciona também interação com outros autores, possibilitando a troca de experiências sobre o uso da linguagem. Por isso, o *Scratch* pode ser também a porta de entrada do educador no mundo da programação, pois opera a partir da lógica comum a todas as linguagens de programação. Para Sant'anna e Neves (2012), iniciativas como o *Scratch Day*, que acontecem em nosso país, proporciona a pessoas interessadas a oportunidade de conhecer e iniciar no uso dessa linguagem.

Modelo de desenvolvimento de OA

Programar não é uma tarefa simples, pois envolve além dos detalhes de conhecer a linguagem de programação, o planejamento do software. E quanto ao planejamento de um OA, se faz necessário também preocupar-se com a estratégia de aprendizagem envolvida. Dessa forma, a construção do OA exige uma estratégia didática aliada à estratégia de programação para o objeto. Por tal característica especial, a participação do educador no processo de desenvolvimento de um OA se faz necessário (AUDINO; NASCIMENTO, 2012).

Uma metodologia de desenvolvimento de software é uma abordagem organizada para atingir um objetivo por meio de passos a serem seguidos usando um ou vários métodos, técnicas e ferramentas que auxiliarão o desenvolvimento do software (REZENDE, 2005).

Para Sommerville (2000), uma metodologia de desenvolvimento de software deve possuir, ao menos, quatro atividades básicas:

- i) Especificação do software: Levantamento das funcionalidades e restrições que o software deve possuir;
- ii) Projeto e implementação de software: O processo de produção do software segundo as especificações;
- iii) Validação do software: Garantir que o software foi desenvolvido de acordo com as especificações dos clientes; e
- iv) Evolução do software: o software deve ser propenso a evoluir para atender as necessidades mutáveis do cliente.

Porém, um OA é um software com a especificidade de ter um componente educacional que deve ser levado em consideração. Esta especificidade requer atenção especial em seu desenvolvimento, que leva a necessidade de uma metodologia que atente para tal característica. Existem propostas de metodologias específicas para este fim, algumas das quais apresentamos no quadro 1.

Quadro 1 - Metodologias propostas por autores para a produção de OA

Descrição da metodologia proposta	Fases da metodologia	Referência
Metodologia baseada em fases, papéis que as pessoas devem desempenhar e artefatos - artefatos podem ser entendidos como qualquer documento que descreva uma parte em um ou mais processos na construção de um software (SUMMERVILLE, 2000) - associados às etapas, estas por sua vez caracterizadas como um conjunto de atividades para a elaboração do OA.	Três fases: inicial, intermediária e transição.	Braga, Pimentel e Dotta (2013).
Metodologia que detalha as fases e especifica três modelos para representar o software: modelo conceitual, navegacional e de interface.	Quatro fases: Planejamento, Modelagem, Desenvolvimento e Encerramento.	Geller e Sobrinho (2011)
Metodologia que se baseia no <i>design</i> instrucional - <i>design</i> instrucional detalha o planejamento do ensino aprendizado, incluindo atividades, estratégias, sistemas de avaliação, métodos e materiais instrucionais servem para estruturar o planejamento do ensino-aprendizagem (FILATRO; PICONEZ, 2004). Usa ferramentas como mapa conceitual - mapa conceitual pode ser definido como uma ferramenta gráfica que apresenta um conjunto de significados e relações significativas a ser apreendida (MONTEIRO et al., 2006; apud MOREIRA, 1988) - para aprendizagem, <i>storyboards</i> - <i>storyboards</i> são representações de momentos de um OA a ser implementado (OLIVEIRA; AMARAL; BARTHOLO, 2010) - para detalhar a implementação dos objetos e sugere vários documentos para descrever o processo e manuais para melhor uso do objeto.	Quatro fases: Análise, Projeto, Implementação e Distribuição.	Oliveira, Amaral e Bartholo (2010).
Metodologia que detalha a concepção de artefatos para descrever a função metodológica do OA com um plano pedagógico e um plano de desenvolvimento. Não detalha a implementação do objeto.	Três fases: Projeto, Desenvolvimento e Distribuição.	Pessoa e Benitti (2008).

Fonte: Próprios autores (2018).

Um ponto em comum que as metodologias citadas no quadro acima possuem é o fato de suportarem o trabalho em equipes para o desenvolvimento de um OA, servindo bem ao propósito de construção do objeto como enfatizam a parte pedagógica, conforme as especificidades de cada uma delas. Em outros trabalhos, como em: Carneiro e Silveira (2012); Nitzke, Carneiro e Passos (2011); e Falkenbach (2005) podemos observar, comparações entre propostas metodológicas para construção de OA.

Ao se trabalhar com um grupo de profissionais sem o uso de uma metodologia para a produção de um OA, corre-se o risco de perda de qualidade, erros de implementação, falhas dentre outros problemas que podem acontecer quando há muitas pessoas envolvidas sem uma metodologia que conduza o trabalho de desenvolvimento do OA, conforme pode ser visto no trabalho de Carneiro e Silveira (2012) e Braga et al. (2012).

Normalmente, quando temos a fase de planejamento pedagógico do OA e a fase de projeto e programação nas propostas, estas tendem ser realizadas por pessoas distintas, com saberes específicos, pois muitas vezes tarefas referentes ao planejamento do software e os detalhes ligados aos conhecimentos em computação, são de domínios de pessoas desta área, assim como os detalhes pedagógicos do OA são de domínio de profissionais da área de educação, podendo ocasionar um distanciamento entre as partes envolvidas no processo de produção de um OA.

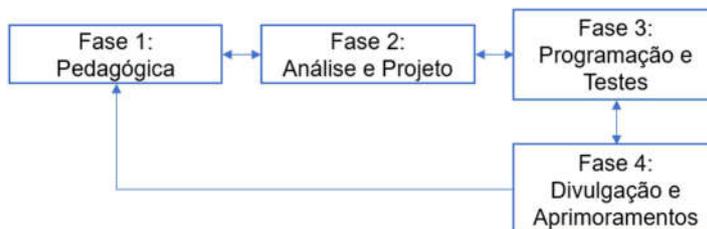
Outro ponto observado é o de que em nenhuma das propostas destacadas temos direcionamentos específicos para a programação com o *Scratch*.

É fato que nem sempre o educador pode dispor de uma equipe para a produção de um OA, como também pode não ter conhecimentos avançados em projeto e programação de software. Entendemos que pode haver o desejo do educador em produzir seu próprio OA para adquirir novos conhecimentos e viver uma experiência computacional em sua vivência docente. Também entendemos que o *Scratch* pode ser um bom começo para que este educador possa aprender conceitos de programação e se motive a produzir OA, e que, para isso, ele necessita de um processo para se orientar inicialmente. Para estes casos citados aqui, defendemos uma proposta de metodologia específica que atenda a tais características.

Metodologia proposta

Respeitando as especificidades da produção de um OA quanto a sua elaboração com duas especificidades, em que uma se refere às suas características pedagógicas e outra a ser um software, apresentamos nossa proposta de metodologia na Figura 1:

Figura 1 - Fases da metodologia proposta



Fonte: Próprios autores (2018).

A Figura 1 representa as fases da metodologia proposta para a criação de OA. Notemos que a metodologia não induz uma sequência obrigatória, pois entendemos que esta deva ser flexível quanto a sua aplicação e deve permitir adaptações que podem ser necessárias durante o processo, e assim, podem ocasionar alterações em fases anteriores. Na fase 4, por exemplo, pode haver a necessidade de uma reconfiguração completa no objeto. Neste caso é necessário voltar e rever todas as etapas anteriores a fim de ajustes e modificações a serem implementados. A seguir, descrevemos as fases especificando as ações e artefatos que podem ser produzidos em cada fase:

i) Fase Pedagógica: Neste momento, o educador ciente da necessidade de confeccionar um OA, deverá planejar aspectos relacionados à sua proposta pedagógica, que envolve: a) definir o componente curricular relacionado à série e assunto a ser trabalhado com o OA; b) definir a quem se destinará este OA; c) determinar o tipo do OA, ou seja, se o objeto será uma apresentação, animação, simulação, dentre outros; d) definir os objetivos educacionais do OA; e) descrever como o objeto cumprirá seus objetivos; f) definir o título do OA; g) prever os benefícios que o uso do OA poderá trazer; e h) prever as limitações pedagógicas que o OA poderá apresentar.

Após a definição destes elementos do OA, o educador poderá consultar se já não existe um objeto que atenda aos critérios estabelecidos nos repositórios de OA, fazendo uma avaliação sobre a real necessidade de construção de um novo OA. Caso não encontre nenhum objeto que satisfaça sua necessidade, ou opte por desenvolvê-lo, o educador passará para a segunda fase:

ii) Análise e Projeto: nesta fase nos deparamos com a necessidade de projetar o OA adaptando as ideias pensadas na fase anterior para a programação. É a fase de ligação entre as ideias e o mundo computacional. Sugerimos os seguintes passos para esta fase: requisitos: descrever como podemos atingir os objetivos propostos na análise em tarefas a serem realizadas com o computador; diagrama de *Storyboard*: criar diagramas que ajudarão no processo de programação; e diagrama de navegação e instrução: construir os diagramas que detalharão com se dará a navegação em telas do objeto.

Acreditamos que é neste ponto que muitos educadores, sem maiores conhecimentos em modelagem de sistemas, sentirão maiores dificuldades. Dessa forma, detalharemos adiante um pouco mais sobre como podemos realizar esse processo.

iii) Programação e Testes: esta é a fase prática, na qual a partir dos insumos criados na análise e projeto, começamos a criar nosso OA. O uso do *Scratch* nos permite programar e testar ao mesmo tempo, um processo dinâmico e ágil.

Nesta fase, a necessidade de um conhecimento mínimo da linguagem se faz necessário. Tais conhecimentos podem ser adquiridos por meio de diversas fontes de informação, como as já citadas, e minicursos fornecidos em plataformas de vídeo como o *YouTube*.

iv) Divulgação e Aprimoramentos: depois de programado e testado é hora de aplicar o OA com o público desejado e sentir a reação quanto ao uso do seu objeto. É importante fazer uma pesquisa sobre a eficácia do objeto com o público alvo para que se possa colher sugestões para possíveis aprimoramentos e evolução do mesmo, assim como fazer uma reflexão sobre o atendimento aos objetivos pedagógicos e técnicos explicitado em fases anteriores. Isso pode ser realizado por meio de observações, pesquisa e testes com o público alvo.

Ao finalizar estas fases, o OA já deve estar pronto e deve expressar a qualidade desejada com a aplicação da metodologia, bem como ter um bom material explicativo sobre o projeto realizado. O próximo passo pode ser o compartilhamento do projeto com toda a comunidade - comunidade *scratch* para divulgação de projeto: <<https://scratch.mit.edu/projects>> se faz necessária a criação de uma conta para divulgação dos trabalhos - que utiliza o *scratch*. Isso é importante, pois permite colher informações de outras pessoas sobre o objeto e trocar informações e experiências diversas que ajudarão no aprimoramento do educador com relação ao uso da linguagem e de experiências pedagógicas com o OA.

Detalhamento da fase de Análise e Projeto

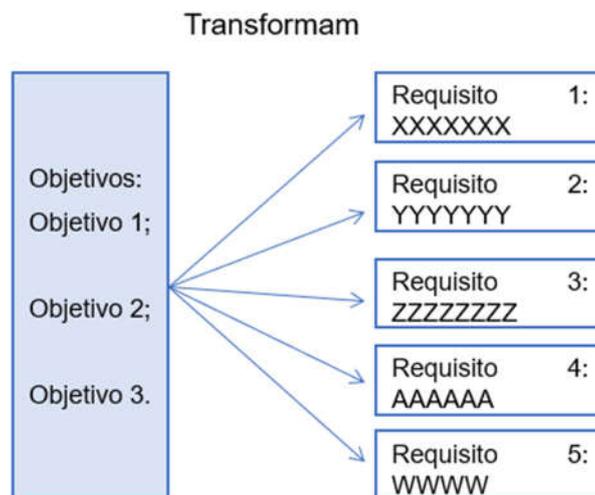
Conforme já explicamos, esta é uma fase que merece destaque, pois é aqui que podem surgir dúvidas para os educadores que não possuem maiores conhecimentos sobre projetos computacionais. Tendo em vista que o objetivo deste trabalho é o de facilitar o processo de construção de um OA e incentivar a formalização de um processo de desenvolvimento deste por pessoas que não tenham maiores conhecimentos na área de projetos de softwares. Propomos a seguir alguns métodos empíricos em que discutimos sua implementação na prática em uma seção posterior.

Requisitos

Requisitos são as descrições das funcionalidades que um software deve atender, devendo ser cuidadosamente trabalhados antes do desenvolvimento de um software (SOMMERVILLE, 2000). Aqui, trabalharemos os requisitos como as funcionalidades a serem implementadas no nosso objeto e estas funcionalidades serão determinadas a partir dos objetivos descritos na fase do projeto, sendo esta etapa a inicial no processo de analisar e projetar o OA.

A figura 2 ilustra uma técnica de especificar requisitos a partir dos objetivos, em que estes podem dar origens a um ou mais requisitos. Os requisitos devem ser descritos como ações e podem ser caminho para a implementação na programação.

Figura 2 - Transformação de objetivos em requisitos



Fonte: Próprios autores (2018).

O processo de transformação de objetivos em requisitos deve obedecer a algumas regras para a criação de requisitos, como citadas em Paula Filho (2003): expressar através de um verbo que represente uma ação que deverá conter no OA; descrevê-lo com clareza e sem ambiguidades; escrever de forma a ser completo e consistente; dentre outras características relevantes nesta etapa. Os requisitos devem ser cuidadosamente escritos e representar ações computacionais.

Os requisitos devem ser numerados para que possamos associa-los aos *storyboards*, em que um ou mais requisitos podem ser representados em um ou mais *storyboards*. O trabalho de Carvalho, Tavares e Castro (2001) detalha um processo de análise de requisitos específico para aplicações educacionais que pode ser usado também para os OA.

Storyboard Proposto

Para criar uma aproximação entre o projeto do sistema e a programação do *Scratch*, propomos um modelo de *storyboard* que irá conter elementos pertencentes à linguagem e ao jeito de programar do *Scratch*, seguindo a lógica de que um programa é uma sequência de tarefas que envolvem entradas, processamento e saída de dados.

Figura 3 - Sequência de processamento de dados



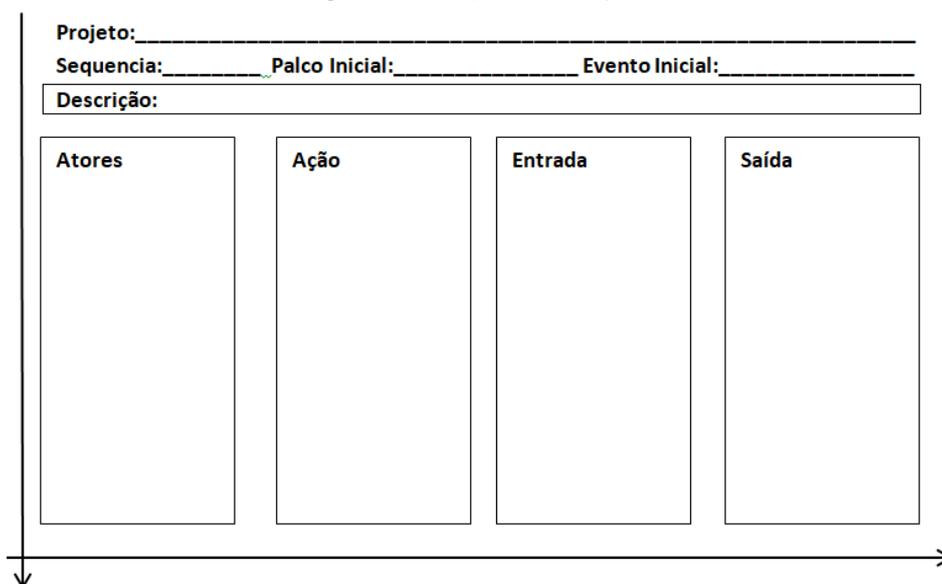
Fonte: Próprios autores (2018).

A figura 3 ilustra que um software deve conter entradas, processamentos e saídas de dados, em que podemos destacar: entradas: qualquer valor que pode iniciar um processamento, podendo ser um dado simples como um número ou sinal que inicia uma ou mais ações a serem realizadas; processamento: ações a

serem realizadas, pré-programadas, que podem ou não usar valores de entrada, podendo produzir saídas; e saída: é todo e qualquer resultado do processamento.

Assim, é esperado que o educador tenha em mente quais serão suas entradas, processamentos e saídas, que podem ser especificados nos requisitos, devendo projetar as ações. Expressamos essas ações em nosso modelo de *storyboard* com o objetivo dessa reflexão para ser programada no *Scratch* de forma mais compreensiva. Apresentamos nosso modelo de *storyboard* na Figura 4:

Figura 4 - *Storyboard* Proposto



Fonte: Próprios autores (2018).

Conforme a figura 4 nosso *storyboard* proposto, contém os seguintes campos a serem informados: Projeto: nome do projeto a ser desenvolvido; Sequência: número da sequência do *storyboard*; Palco Inicial: palco no qual a história inicia; Evento Inicial: evento que inicia a cena descrita; descrição: descrição da cena que será programada; Atores: os atores envolvidos na cena que se deseja implementar; Ação: neste espaço temos a sequência de ações que um ou mais atores realizam. As ações referem-se a comandos disponíveis do *Scratch* descritos nas opções de: movimento, aparência, caneta, controles, eventos e operadores; entrada: possíveis interações que a cena poderá conter, refere-se a entrada de valores de variáveis e sensores; e Saída: possíveis saídas que podemos ter durante a cena, que podem estar relacionado a valores de variáveis, sons e itens de aparência.

As setas para baixo e para a esquerda da Figura 4, fazem referências à sequência que podemos ter dos comandos que serão implementados e indicam que devemos seguir esta ordem (ordenação das ações). Comandos na mesma linha deverão estar em paralelos, ou seja, ações que ocorrerão de forma simultânea conforme as ações previstas nas colunas. Setas poderão ser usadas para criar sequências específicas de ordem de execução entre os comandos, conforme detalharemos na implementação.

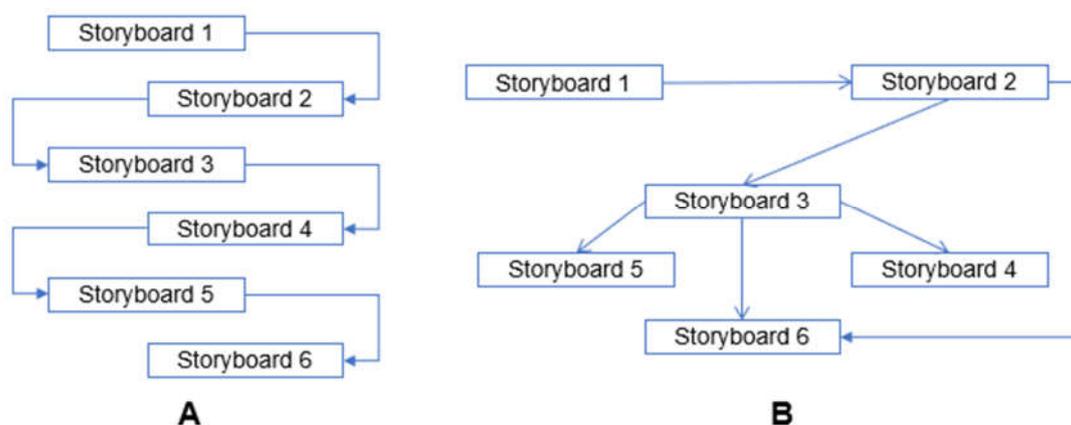
O diagrama proposto na Figura 4, tem por objetivo implementar uma história, ou parte de uma, a partir das entradas, processamentos e saída de dados, levando em consideração uma forma sequencial para a história, seguindo uma visão do ator, que pode ser o objeto principal na programação. Dessa forma, o diagrama apresenta a sequência de ações iniciada por um evento inicial, tendo um palco inicial para a cena, que pode preceder ou suceder diversos outros diagramas na sua representação.

O *storyboard* pode conter também a descrição mais detalhada de eventos como conversas, sons, decisões dentre outros que ajudem a melhor descrever a cena a ser implementada.

Diagrama de Navegação e instrução

Este diagrama tem como função principal a organização dos *storyboards* e também a função de apresentar como estes aparecerão durante a execução. Podemos ter com relação à navegação a forma sequencial como descrito no próprio *storyboard* por meio da sequência ou de forma interativa. A figura 5 ilustra a disposição destes diagramas.

Figura 5 - Diagrama de Navegação



Fonte: Próprios autores (2018).

Na figura 5 apresentamos como podemos descrever a sequência de telas que compõem uma história que pode ser de forma sequencial (A), na qual cada *storyboard* precede em sequência para outro, ou de forma interativa (B), em que as *storyboards* podem aparecer de forma aleatória. Este caso é comum quando temos decisões a serem tomadas em uma história.

Notemos que este diagrama permite compor a descrição do OA quanto a navegabilidade de acordo com o objetivo desejado para o objeto, sendo importante estar documentado para que possa compor a descrição do objeto e melhor apresentá-lo aos demais interessados.

Além do diagrama, podemos citar também as interações que poderemos ter em cada *storyboard*, como entradas e saídas em postos específicos de decisão, para melhor orientação em caso de navegação. Isto pode ser necessário em modelos interativos, embora tenhamos as entradas e saídas refletidas nos *storyboard*, possibilitando a visão da navegabilidade no OA.

Estudo de caso

Para discutir a aplicação da metodologia proposta e os insumos propostos neste trabalho, ilustraremos parte de um OA desenvolvido durante uma disciplina em um programa de Pós-Graduação cursada pelos autores no ano de 2017 - Programa de Pós-Graduação em Ensino de Ciência e Tecnologia (PPGECT) da UTFPR, disciplina de Fundamentos de Tecnologias Digitais. O exemplo pode ser encontrado em: <<https://scratch.mit.edu/projects/157118758/>> -, servindo esse como exemplo de aplicação da metodologia e discussões sobre as etapas. O objeto em questão foi produzido sem que houvesse um processo de produção anterior a ele e serviu como estudo de caso para a metodologia.

Fase 1: Etapa pedagógica

Nesta etapa, os atores devem refletir sobre as questões propostas e formalizarem:

Título do objeto: simulador de colisões;

Tipo: simulador;

Objetivos:

Geral: estabelecer relações entre causas e efeitos do movimento de corpos rígidos, calculá-los e interpretá-los sobre diferentes contextos;

- **Específicos:** identificar os diferentes tipos de movimentos de sistemas simples; estabelecer as condições de conservação do momento; relacionar a quantidade de movimento à dinâmica dos corpos em colisões; e trabalhar com situações práticas da lei da conservação da quantidade de Movimento de uma força no *software Scratch*.

Descrição do Objeto: o simulador trabalha a colisão entre dois corpos interagindo com o aluno a fim de que ele revise questões sobre o assunto, para que os alunos possam perceber as relações físicas existentes na colisão entre os corpos, suas aproximações e distanciamentos simulando uma hipotética situação real de colisões entre veículos no trânsito.

Componente curricular: disciplina de Física; tema: movimentos, variações e conservações; conteúdo: conservação da quantidade de movimento/colisões.

Público Alvo: alunos da 1ª série do Ensino Médio;

Benefícios: resgatar os conceitos da Física envolvidos em diferentes tipos de movimentos: acelerado e retardado da Cinemática; desmistificar situações cotidianas em relação a sistemas ideais - para as Leis de Newton e para o Impulso de uma força; enfatizar as relações da dinâmica dos corpos para a conservação da energia – Energia Cinética; estabelecer a dedução matemática das condições da Conservação da Quantidade de Movimento Linear ou *Momentum* Linear; classificar os diferentes tipos de colisões em relação ao coeficiente de restituição nos corpos rígidos em sistemas ideais e não-ideais.

Limitações: por se tratar de aplicação de formas físicas em uma situação hipotética ideal os resultados simulados podem diferir dos resultados em uma situação real.

Após discussões iniciais sobre o OA a ser construído e os pontos determinantes ao projeto serem estabelecidos, seguimos para a próxima fase.

Fase 2: Análise e Projeto

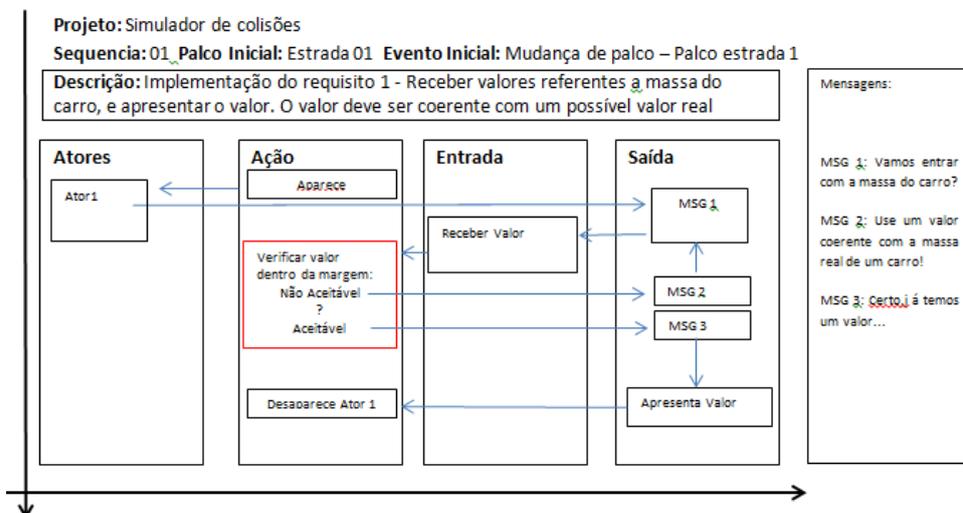
Esta fase inicia com os requisitos que o sistema deverá atender. Selecionamos apenas alguns requisitos para ilustrar essa fase tendo em vista que são vários requisitos para cada objetivo proposto.

Então, inicialmente, com base no objetivo *Estabelecer as condições de Conservação do Momentum*, trabalha-se a entrada de valores para um carro em movimento e calculamos sua quantidade de energia no momento. Selecionamos os seguintes requisitos para este objetivo:

1. Receber valores referentes a massa do carro, e apresentar este valor, que deve ser coerente com um possível valor real;
2. Receber valores referentes a velocidade do carro e apresentar o valor resposta que deve ser coerente com um possível valor real;
3. Solicitar que o aluno efetue o cálculo da quantidade de movimento do carro sem apresentar a fórmula (Quantidade de movimento = massa do carro * velocidade), verificando se a resposta do aluno condiz com o cálculo realizado pelo programa. Se a resposta do aluno condiz com o cálculo do sistema, o sistema emitirá uma mensagem de elogio. Caso a resposta não esteja de acordo com o cálculo do programa, o programa apresentará a fórmula do cálculo ao aluno e pedirá que este refaça o cálculo novamente.

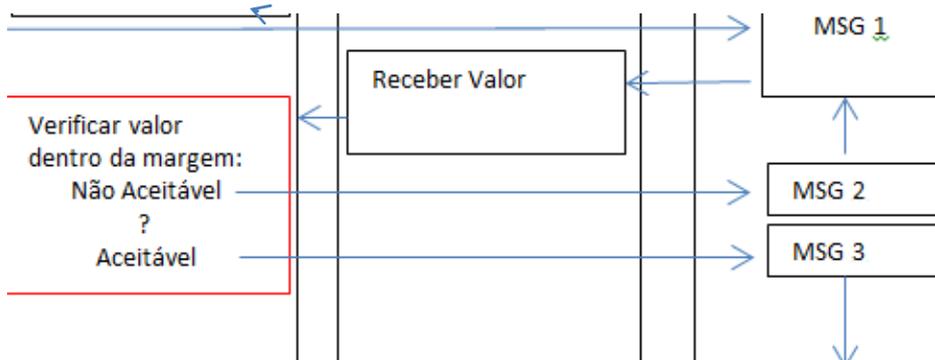
Após levantarmos os requisitos, prosseguimos com a construção dos *storyboards*. A Figura 6 ilustra a implementação do Requisito 1, que deve iniciar com a troca de palco, apresentando o ator 1 que interagirá com o aluno para que ele possa inserir um valor referente a massa do carro. O valor deverá estar em uma faixa de valores hipotéticos aceitáveis, uma condição em que o valor recebido deve estar entre 100 a 2000 Kg (por exemplo), congelando a tela de entrada caso esse valor não seja condizente com a faixa de entrada desejada. Essa decisão é evidenciada pelo caractere “?” e as opções que conduzem a alternativas diferentes conforme pode ser observado na sequência das setas. Note que a sequência de setas, dependendo da escolha, condiciona a repetição de alguns comandos, conforme destacamos na figura 7.

Figura 6 - Criação dos Storyboards sequência 01



Fonte: Próprios autores (2018).

Figura 7 - Ilustração do processo de decisão e repetição



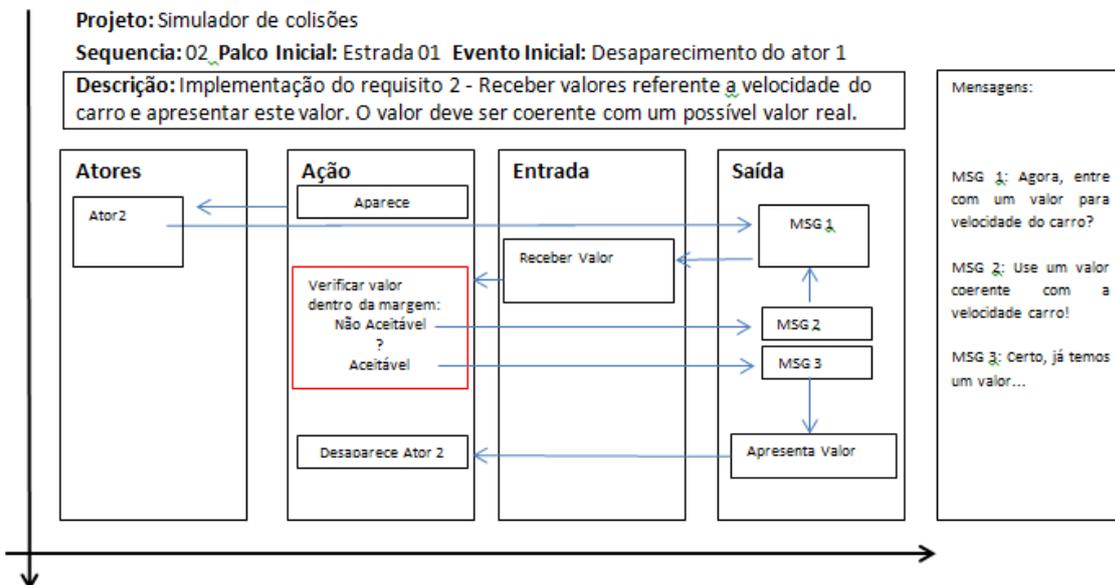
Fonte: Próprios autores (2018).

Essa estratégia de representação detalhada na Figura 7 mostra que se a opção não for aceitável o fluxo normal é revertido em sentido contrário (seta para cima) repetindo ações já realizadas, condizendo com um laço de repetição.

A Figura 8 ilustra a implementação do Requisito 2 que inicia quando o Ator 1 desaparece e tem uma sequência parecida com a anterior quanto as especificações do projeto e detalha o recebimento de valor referente a velocidade do carro e a verificação destes, segundo os parâmetros aceitáveis para o cálculo.

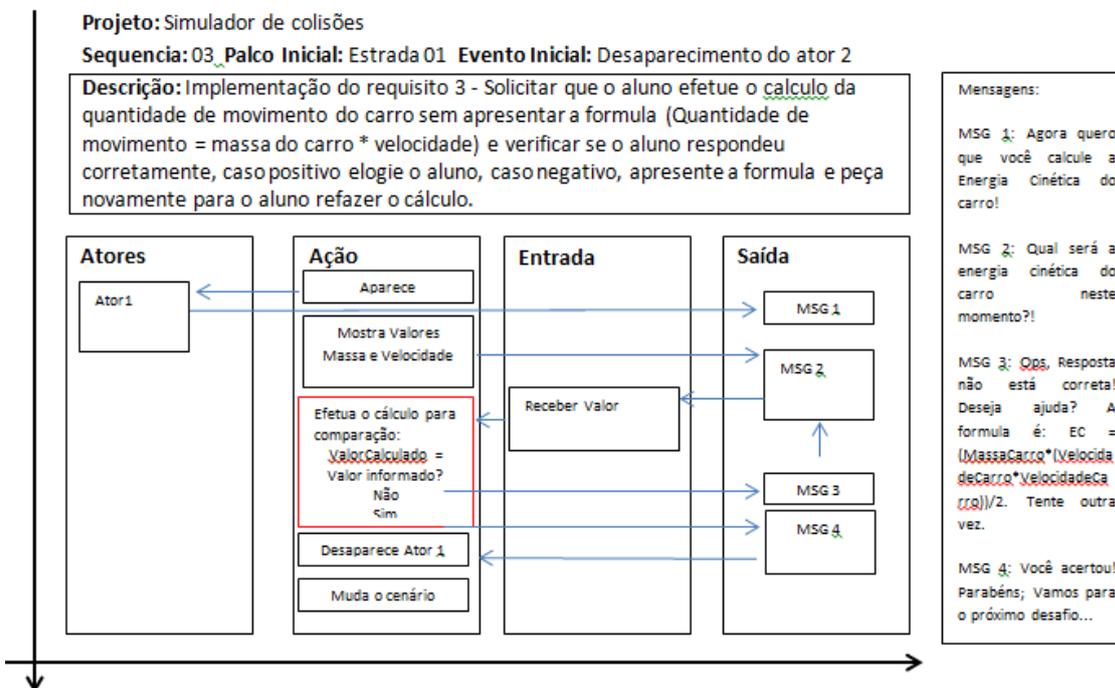
A Figura 9 detalha uma cena onde o Ator 1 deve interagir com o usuário a fim de que este faça o cálculo da conservação de energia e o sistema conferirá se este está condizente com os valores calculados pelo sistema segundo as informações inseridas anteriormente. Este *storyboard* reflete uma das metas pedagógicas e deve ser trabalhado com essa intenção, que é uma fixação de conceito de aplicação de uma fórmula. Caso o aluno insira valores não corretos, segundo o cálculo, haverá uma interação para conduzir o aluno ao cálculo correto.

Figura 8 - Criação dos Storyboards seqüência 02



Fonte: Próprios autores (2018).

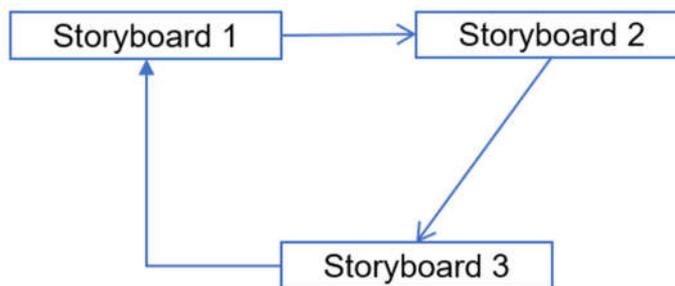
Figura 9 - Criação dos Storyboards seqüência 03



Fonte: Próprios autores (2018).

Após todos os storyboards estarem prontos, procedemos com o diagrama de navegação e instrução, conforme apresentamos na Figura 10, segundo nosso exemplo:

Figura 10 - Diagrama de Navegação e Instrução



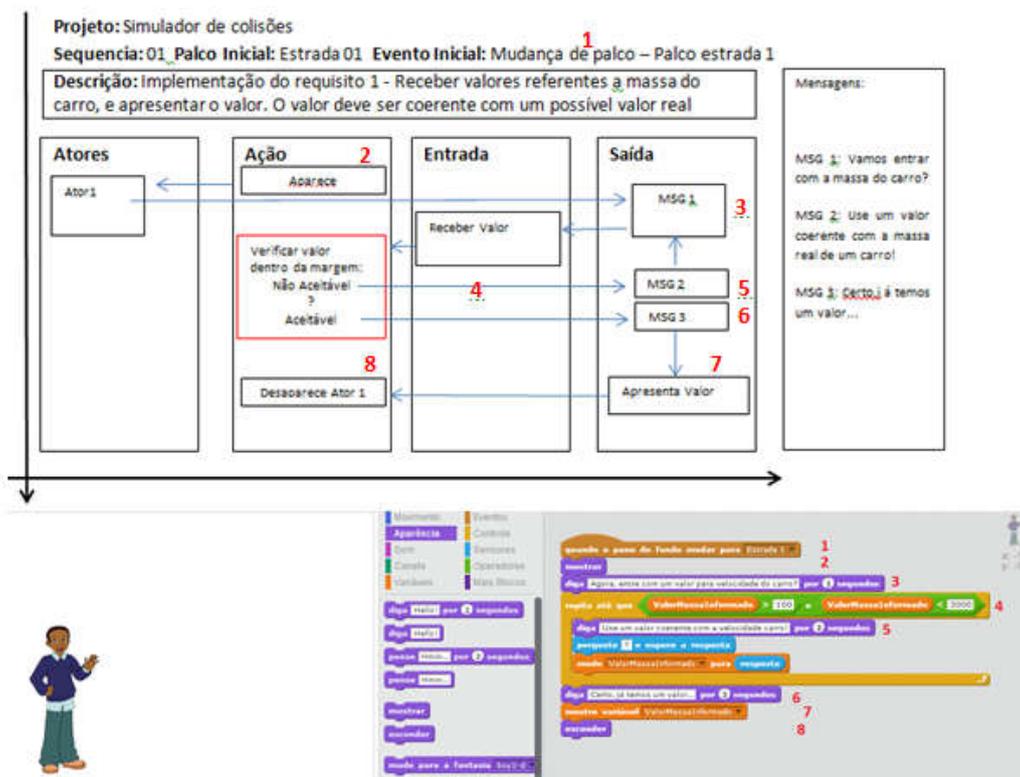
Fonte: Próprios autores (2018).

Como detalhamos apenas um objetivo, nosso diagrama de navegação, neste caso, seria do tipo sequencial quanto à navegação.

Fase 3: Programação e Testes

Nesta fase usaremos os *Storyboards* para nortear a codificação na linguagem *Scratch*. Ilustraremos uma técnica para esse processo por meio da Figura 11:

Figura 11 - De Storyboard ao código



Fonte: Próprios autores (2018).

A Figura 11 é composta de dois momentos, um é o *storyboard* 01, o qual desejamos implementar e o outro é a codificação do mesmo usando o *Scratch*. A numeração na figura 11 reflete as correlações existentes entre o modelo do projeto e o código, que explicamos no Quadro 2.

Quadro 2 - Correlações entre o *storyboard* e a programação em *Scratch*.

Número	<i>Storyboard</i>	<i>Scratch</i>
1	Evento Inicial – Mudança de palco marca o início da sequência proposta no <i>storyboard</i> .	Comando de início de um bloco de comandos “Quando o fundo mudar para”. Inicia a sequência abaixo dos comandos.
2	Aparecer o Ator 1.	Comando “mostrar”. Mostra o ator ao qual o código pertence.
3	Saída de uma mensagem, conforme descrito para MSG 1.	Comando “diga”. Apresenta uma mensagem na tela por um tempo determinado.
4	Comando de repetição com uma comparação lógica para que o valor a ser recebido esteja dentro de um escopo aceitável. Espera que o aluno entre com o valor.	Comando “Repita até que” com uma expressão lógica interna a ser verificada. Este comando é um bloco de repetição, deixa tudo preso a condição a ser satisfeita.
5	Exibir a mensagem MSG2.	Comando “diga”. Apresenta uma mensagem na tela por um tempo determinado, preso ao laço.
6	Exibir a mensagem MSG3.	Comando “diga”. Apresenta uma mensagem na tela por um tempo determinado, saída do laço.
7	Apresenta o valor digitado pelo Aluno.	Comando “Mostre a Variável” Mostra a variável na qual o valor foi armazenado.
8	Desaparece Ator 1, esconde o ator 1 do palco da cena.	Comando “esconder”, esconde o ator na qual os comandos pertencem.

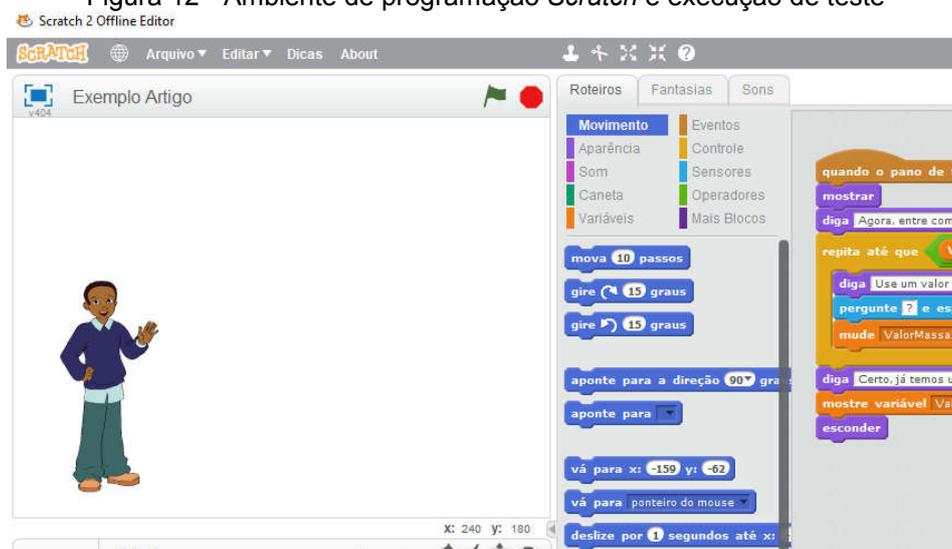
Fonte: Próprios autores (2018).

Podemos observar no Quadro 2 a existência de uma aproximação entre o modelo de projeto e a implementação, tornando essa passagem mais tranquila para o autor do OA com pouca experiência. Lembramos que é essencial conhecer as particularidades da linguagem *Scratch* assim como um pouco de lógica computacional para desenvolver OA com um pouco mais de interação.

As setas são as sequências na qual o código deve estar, notem que a sequência das setas é semelhante a sequência dos códigos de programação. Todos os códigos devem pertencer a algum objeto, normalmente a um ator especificado no *storyboard* e em alguns casos ao palco. Pode haver sequência de códigos de um ou mais autores em uma mesma cena.

A sequência das cenas e as interligações entre elas no projeto devem obedecer ao diagrama de navegação e instrução e a mudança entre as cenas devem seguir a ordem prevista no evento inicial detalhado no *storyboard*, sendo que o último comando previsto deve ser o início da sequência da próxima cena, refletida no *storyboard* posterior. Com essa estratégia ligamos as cenas entre os *storyboards*.

Figura 12 - Ambiente de programação *Scratch* e execução de teste



Fonte: Ambiente de programação *Scratch*.

Os testes devem ser realizados a cada implementação de *storyboard*, para dessa forma observarmos o comportamento de nosso OA conforme desejado e projetado. O *Scratch* permite isso de uma forma simples, apenas clicando na bandeira verde no ambiente de programação, conforme pode ser observado no palco da Figura 12.

Fase 4: Divulgação e Aprimoramentos

A divulgação consiste inicialmente em após os testes, aplicar com o público alvo o OA e observar e refletir sobre sugestões de melhorias e aprimoramento que podem surgir nesse primeiro momento. Isso é importante para verificarmos se o OA cumpre sua função pedagógica.

Caso haja melhorias a serem feitas, devemos voltar a fase 1 e iniciar todo o processo novamente.

O objeto pode ser divulgado também na página de projetos do *Scratch*, na qual também poderão ser colhidas sugestões para o aprimoramento do mesmo.

Conclusões e trabalhos futuros

Apresentamos aqui uma estratégia para educadores desenvolverem OA de autoria própria. A metodologia apresentada contempla aspectos desejados a uma metodologia de desenvolvimento de *software* que é a de fornecer um produto com um mínimo de qualidade por meio de um processo estabelecido e documentado. Além da proposta, apresentamos algumas técnicas que podem ser usadas para a modelagem e programação com *Scratch* e buscamos incentivar a formalização do processo de produção de um OA.

Nosso maior destaque neste trabalho está na ênfase da fase de análise e projeto, em que adaptamos a ferramenta de *storyboard* à programação com *Scratch*, com o objetivo de diminuir a distância entre o planejamento do *software* e a programação, fases em que pessoas sem o mínimo de conhecimento em

modelagem computacional podem considerar complexas. Aqui, tentamos tornar mais simples todo o processo de construção de um OA para uma melhor compreensão, sem no entanto nos aprofundarmos em outras questões computacionais complexas que envolvem a criação destes objetos.

Apresentamos uma metodologia simples, voltada a educadores que desejam criar OA com um mínimo de formalização computacional, detalhando um pouco esta com um exemplo de sua aplicação, buscando ilustrar a simplicidade deste processo.

No futuro, a metodologia poderá ser testada em sala de aula na disciplina da qual nasceu a proposta, ou por outra disciplina na qual a produção de OA se faça necessário, por meio de um trabalho com os alunos sobre o assunto de desenvolvimento de Objetos Educacionais usando esta metodologia proposta, assim como relatar as experiências dos alunos em desenvolverem e aplicar seus próprios OA com o público alvo desejado. Também julgamos interessante a possibilidade de produzir uma ferramenta computacional que nos ajude no processo de modelagem e documentação para a construção de um OA.

Agradecimentos e apoios

"O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001"

Referências

AUDINO, D.F.; DA SILVA NASCIMENTO, R.. Objetos de Aprendizagem—diálogos entre conceitos e uma nova proposição aplicada à educação. **Revista Contemporânea de Educação**, v. 5, n. 10, 2012.

AVILA, B. G. et al. A autoria nos Mundos Virtuais: um novo desafio ao docente. **Renote**, v. 12, n. 2, 2014.

BASTOS, B. L.; BORGES, M.; D'ABREU, J. Scratch, Arduino e o Construcionismo: ferramentas para a educação. **Seminário de Tecnologia Educacional de Araucária**, 2010.

BATTISTELLA, P. E. et al. Classificação de Objetos de Aprendizagem e análise de Ferramentas de Autoria. **XX Simpósio Brasileiro de Informática na Educação**, Florianópolis-SC, 2009.

BEHAR, P.A. et al. Objetos de aprendizagem para educação a distância. **Modelos pedagógicos em educação a distância. Porto Alegre: Artmed**, p. 66-92, 2009.

BRAGA, J. C. et al. Desafios para o Desenvolvimento de Objetos de Aprendizagem Reutilizáveis e de Qualidade. In: **Anais do Workshop de Desafios da Computação Aplicada à Educação**. p. 90-99, 2012.

BRAGA, J.C.; PIMENTEL, E.; DOTTA, S.. Metodologia INTERA para o desenvolvimento de Objetos de Aprendizagem. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2013.

CARNEIRO, M.L.; SILVEIRA, M.S. Impactos do Uso de uma Metodologia na Produção de Objetos de Aprendizagem. **Conferencias LACLO**, v. 3, n. 1, 2012.

CARVALHO, A. E.; TAVARES, H. C. A. B.; CASTRO, J. Uma Estratégia para Implantação de uma Gerência de Requisitos visando a Melhoria dos Processos de Software. **WER01**. Buenos Aires, Argentina, 2001.

FALKEMBACH, G.A.M. Concepção e desenvolvimento de material educativo digital. **RENOTE**, v. 3, n. 1, 2005.

FILATRO, A.; PICONEZ, S.C.B. Design instrucional contextualizado. **São Paulo: Senac**, 2004.

GELLER, M.T.B.; SOBRINHO, M.C. Utilização de um processo adaptado para desenvolvimento de software educativo em um modelo de ação interdisciplinar. In: **Congreso Internacional de Informática Educativa**, Chile. p. 95-99, 2011.

GRANDO, A.R.C.S.; KONRATH, M.L.P.; TAROUCO, L.M.R. Alfabetização visual para a produção de objetos educacionais. **RENOTE: revista novas tecnologias na educação [recurso eletrônico]**. Porto Alegre, RS, 2003.

MONTEIRO, B.S. et al. Metodologia de desenvolvimento de objetos de aprendizagem com foco na aprendizagem significativa. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. p. 388-397, 2006.

MOREIRA, M.A. Mapas conceituais e aprendizagem significativa. **O ENSINO, Revista Galáico Portuguesa de Sócio-Pedagogia e Sócio-Linguística**. Pontevedra/Galícia/Espanha e Braga/Portugal, N0 23 a 28: 87-95. 1988.

NITZKE, J.A.; CARNEIRO, M.L. F.; PASSOS, P. C. S. J. Gestão do desenvolvimento de objetos de aprendizagem digitais. In: **Sexta Conferencia Latinoamericana de Objetos de Aprendizaje y Tecnología de la Educación-LACLO**. p. 111-120, 2011.

OLIVEIRA, K.A.; AMARAL, M. A.; BARTHOLO, V.F. Uma experiência para definição de storyboard em metodologia de desenvolvimento colaborativo de objetos de aprendizagem. **Ciências & Cognição**, v. 15, n. 1, p. 19-32, 2010.

PASSOS, M.L.S. SCRATCH: UMA FERRAMENTA CONSTRUCIONISTA NO APOIO A APRENDIZAGEM NO SÉCULO XXI. **Revista Eletrônica Debates em Educação Científica e Tecnológica**, v. 4, n. 02, p. 68-85, 2015.

PAULA FILHO, W.P. **Engenharia de software**. LTC, 2003.

PESSOA, M.C.; BENITTI, F.B.V. Proposta de um Processo para Produção de Objetos de Aprendizagem. **Hífen, Uruguaiana**, v. 32, n. 62, p. 172-180, 2008.

REZENDE, D. A. Engenharia de Software e Sistema de **Informações**. Editora Brasport, Ed. 3. Rio de Janeiro – RJ. 2005.

RODRIGUES, A.; STANO, R.C.M.T. A FORMAÇÃO DOCENTE COMO CONSTRUÇÃO COLETIVA DE CONHECIMENTO EM UMA EXPERIÊNCIA DE WEB CURRÍCULO. **Revista Contemporânea de Educação**, v. 11, n. 21, p. 64-81, 2016.

SANT'ANNA, H.C.; NEVES, V.B. Scratch Day UFES: oficina itinerante de introdução a programação para professores. **IV Simpósio Hipertexto e Tecnologia na Educação, Recife-PE**, 2012.

SOMMERVILLE, I. **Software Engineering**, Ed. 6th, Addison- Wesley, 2000.

VALENTE, J.A. Diferentes usos do computador na educação. **Em aberto**, v. 12, n. 57, 1993.

VALENTE, J.A. et al. Informática na educação: instrucionismo x construcionismo. **Manuscrito não publicado, NIED: UNICAMP**, 1997.

Submetido em 25/08/2018.

Aceito em 19/10/2018.

