

USO DO MIT-SCRATCH PARA O ENSINO DE PROGRAMAÇÃO DE COMPUTADORES NUMA ABORDAGEM HISTÓRICO-CULTURAL

Jaylson Teixeira

UFRB

jaylsont@ufrb.edu.br

Paulo Cesar Cerqueira dos Santos Junior

UFRB

pc.jr.14@hotmail.com

Iani Caroline da Rocha Santana

UFRB

ianicaroline@hotmail.com

Resumo:

Este trabalho tem a intenção de contribuir e explorar as possibilidades do ensino de programação de computadores para o curso de licenciatura em matemática. Para isso iniciarei o minicurso com uma parte prática de programação de computadores e em seguida relacionarei as atividades com os aspectos histórico-culturais, com a pedagogia de projetos e com o construcionismo de Papert.

Palavras-chave: MIT_Scratch; Construcionismo; Ensino; Programação.

1. Introdução

Formei-me em Licenciatura em Matemática no ano de 1991 e segui uma carreira profissional desenvolvendo softwares para a iniciativa privada com diversas tecnologias chegando a fazer mestrado em Informática, em especial em Engenharia de Software. Engenharia de Software se ocupa de várias fases e aspectos de um projeto de software. Em 2010 iniciei uma nova perspectiva profissional sendo responsável por ministrar a disciplina de Introdução a Lógica de Programação para os cursos de Licenciatura em Matemática e Licenciatura em Física na Universidade Federal do Recôncavo da Bahia - UFRB. Naquele instante constatei que pouco se modificou no ensino de programação para os cursos de licenciatura desde a minha formatura. Nas aulas eu utilizava uma linguagem hipotética conhecida como português Estruturado ou Portugol. Os conceitos eram apresentados em aulas expositivas e os exercícios eram realizados com tecnologia de papel e lápis. Do meio para o fim da disciplina era introduzida uma linguagem real de programação (eu utilizei a linguagem C e minha antecessora utilizava a linguagem PASCAL) gerando programas em console de caracteres com tela preta semelhante ao MS-DOS. Os exercícios propostos

eram essencialmente numéricos com enunciados do tipo “Apresente os números primos entre 1 e 100”, “Gere a sequência de Fibonacci”, “Descubra o valor máximo em vetor dado”.

Esta situação me causou diversas inquietações. Não podemos usar interfaces com multimídias (sons, imagens, animações)? Trabalhando como desenvolvedor de software observei que o conceito de orientação a objetos é largamente utilizado desde os anos 90. Este conceito não deveria ser apresentado aos alunos? Além disso, os alunos se mostravam bastante irritados com os erros de sintaxe, que apareciam em grande quantidade, os quais deveriam ser superados antes de se ocuparem com os algoritmos.

Desse instante em diante estou refletindo e modificando a forma de ensino dessa disciplina. Comecei utilizando os computadores desde a primeira aula, acreditando que a prática da programação melhora o entendimento possibilitando experimentação. Depois passei a utilizar o MIT-Scratch, evitando os erros de sintaxe e podendo utilizar melhor a atenção dos alunos para os algoritmos. O MIT-Scratch também possibilitou o uso de recursos multimídia, orientação a objetos e execução de algoritmos em paralelo, com relativa facilidade.

As inquietações acima explicitadas me motivaram a investigar de que forma ou como o uso do ambiente Scratch contribui para a compreensão da lógica de programação sob uma visão histórico-cultural.

2. Metodologia do Minicurso

Neste minicurso pretendo mostrar na prática alguns exemplos de roteiros que utilizo no laboratório de informática. Os participantes desse minicurso devem executar estes roteiros em computadores com o software MIT-Scratch instalado (O instalador desta ferramenta se encontra em http://info.scratch.mit.edu/Scratch_1.4_Download). Os roteiros fazem parte de uma estratégia de ensino-aprendizagem como mencionaremos mais adiante.

Para o minicurso pretendo utilizar um conjunto de 5 roteiros totalizando 28 páginas por participante. Pretendo que os participantes executem pelo menos 3 dos seguintes roteiros:

- **Roteiro 01:** Hello Wolrd, Gato Caminha.
- **Roteiro 02:** Voo do Morcego.
- **Roteiro 03:** Dancing Queen.
- **Roteiro 04:** Labirinto.
- **Roteiro 05:** Caça Fantasma.

Recomendo este minicurso para **20 pessoas** sendo 25 o máximo recomendado.

Seguiremos a seguinte agenda em 4 horas de minicurso:

- Verificação das Instalações (10min)
- Execução de Roteiros no Computador (1h50min)
- **Intervalo** (30min)
- Apresentação da Metodologia de Ensino-Aprendizagem (30min)

- Discussão (60min)

3. Ensino Tradicional

Para se ter ideia do como se procedem as aulas em um modelo tradicional, vamos tomar como base dois livros largamente utilizados nas universidades brasileiras que são : Algoritmos : lógica para desenvolvimento de programação de computadores de José Augusto N. G. Manzano e Jayr Figueiredo de Oliveira (MANZANO, 2009) e Lógica de programação de Andre Luiz Villar Forbellone e Henri Frederico Eberspacher (FORBELLONE, 2005). Os dois livros iniciam apresentando o fluxograma como ferramenta de planejamento de uma programação estruturada, logo seguem capítulos apresentando separadamente, tipos de dados, expressões aritméticas, estruturas de tomada de decisão, operações lógicas, matrizes, vetores e estrutura de dados heterogêneos. O livro de Forbellone vai mais além tratando de arquivos de dados, módulos de programas e estruturas avançadas de dados (listas, filas, pilhas e árvores). Os exercícios são limitados ao que foi exposto no capítulo, por exemplo, após o capítulo Tipos de Dados e Instruções Primitivas de Manzano temos enunciados os exercícios:

- “Marque com um X quais dos tipos de dados abaixo é do tipo Inteiro.” (MANZANO, 2009, p. 39)

- “Desenvolva os algoritmos, diagrama de blocos e codificação em português estruturado dos seguintes programas: a) Ler uma temperatura em graus Centígrados e apresentá-la convertida em graus Fahrenheit...” (MANZANO, 2009, p. 41).

Após o capítulo Estruturas de Controle de Fobellone, temos os exercícios propostos que estão separados por estruturas. No grupo de exercícios de estrutura de repetição encontra-se:

- “Elabore um algoritmo que obtenha o mínimo múltiplo comum (MMC) entre dois números”(FOBELLONE, 2005, p. 64).

- “Escreva um algoritmo que imprima todas as possibilidades de que no lançamento de dois dados tenhamos o valor 7 como resultado da soma dos valores de cada dado” (FOBELLONE, 2005, p. 64).

Estes livros estão baseados no paradigma da programação estruturada, sem mencionar orientação a objeto nem execução de tarefas em paralelo.

4. MIT-Scratch

O MIT-Scratch é uma linguagem de programação e um ambiente de desenvolvimento de software gratuito que permite a criação de histórias interativas, animações, jogos, músicas e artes (MARQUES, 2008). A programação é efetuada através da criação de sequencias de comandos simples, que correspondem a blocos de várias categorias, encaixados e encadeados de forma a produzirem as ações desejadas. O que estamos chamando de MIT-Scratch é o projeto Scratch do Instituto de Tecnologia de Massachusetts – MIT. O MIT-Scratch foi desenvolvido para pessoas acima de 8 (oito)

anos, no sentido de facilitar o processo de ensino-aprendizagem de Matemática e Informática, estimulando as competências requeridas pelo mundo contemporâneo (RUSH et ali, 2012). Segundo estes autores, tais competências são: as competências relacionadas à utilização de informação e comunicação, ao raciocínio e resolução de problemas e às relações interpessoais e de auto-direcionamento. Ainda segundo Rush et ali (2012), Competências de Informação e Comunicação se refere a capacidade de criar e gerenciar múltiplas mídias (imagens, textos e animações, gravação de áudio etc) desenvolvendo uma relação crítica com os meios de comunicação de hoje, que não se restringe apenas ao texto escrito. Competências de Raciocínio e Resolução de Problemas é a capacidade estimulada pelo MIT-Scratch de um raciocínio crítico e sistêmico na concepção (*design*) de resolução de problemas. Estimulando a resolução de problemas em contextos inesperados e de forma inovadora. Competências Interpessoais e de Auto- Direcionamento é consequência da facilidade de escrever e compartilhar programas em MIT-Scratch estimulando o relacionamento interpessoal. Os projetos compartilhados a partir de ideias que o autor considera pessoalmente interessantes estimulam na superação de desafios e frustrações encontrados no processo de resolução de problemas.

O documento Conceitos de Programação e capacidades Suportadas pelo Scratch disponibilizado pelo Instituto de Tecnologia de Massachusetts em <http://scratch.mit.edu/>, menciona os conceitos ensinados pelo MIT-Scratch e fala dos conceitos: sequencia, interação (laços ou looping), comandos condicionais, variáveis, lista (vetores), tarefas (execução paralela), coordenação e sincronização, entrada pelo teclado, interação dinâmica, desenho de interface de usuário. E dá como inexistentes do Scratch os seguintes conceitos: procedimentos e funções; passagem de parâmetros e retorno de valores; definição de classes de objetos; tratamento de exceções; arquivos de entrada e saída.

5. Roteiros e Zonas de Desenvolvimento Proximal

Segundo Emilia Ferreiro (apud PRETTO, 1999) diz que insistimos que a aprendizagem deve ocorrer sempre do concreto para o abstrato, do próximo para o distante, do fácil para o difícil. De maneira fragmentada e ordenada. No entanto, Pretto (1999) lembra-se da geração que ele chama de Novas Tribos. Esta geração convive com videogames, televisão, Internet, esportes radicais, tudo simultaneamente, de forma múltipla, fragmentada e não ordenada, revelando um novo processo de produção de conhecimento, ainda desconhecido pela escola. Além disso, ao programar um computador os conceitos de variável, loops condicionais, execução em paralelo etc dificilmente aparecem de forma isolada. Constatado essa realidade da programação de computadores e a aceitação de uma didática fragmentada, resolvi adotar uma prática de programação por exemplos, criando programas que iam aumentando em tamanho e complexidade. A partir desses exemplos criei roteiros para facilitar o procedimento em laboratório.

Através desses roteiros os alunos conseguem implementar e executar os exemplos. Além dos roteiros, os alunos são ajudados pelo professor e pelos seus próprios colegas. Os roteiros se sucedem, intercalados por aulas teóricas. Os conceitos são apresentados em

aulas teóricas, geralmente após a execução de um roteiro que usa o conceito a ser ensinado. Acreditamos que nesse ambiente os alunos são submetidos a uma aprendizagem coletiva e colaborativa, passando por zonas de desenvolvimento proximal, aumentando gradualmente suas habilidades na programação de computadores, numa perspectiva histórico-cultural (REGO, 1994) (FREIRE, 1999).

Ainda tomando proveito da cultura estabelecida nessa geração de nativos digitais, a maioria dos roteiros trata de animações e jogos, aumentando o interesse através da ludicidade.

6. Construcionismo e Pedagogia de Projetos

A pedagogia de projetos acredita que deve-se colocar um desafio ao aluno que possibilite a atuação dele sobre o mundo (ALMEIDA, 2002). Segundo Machado (2000), a raiz da palavra projeto é a mesma de projetar, lançar, remetendo a idéia de se imaginar (lançar) em um futuro que está por vir, em um exercício de previsão, que pode levar à modelação desse futuro conforme à nossa vontade. A pedagogia de projetos tem como característica a intencionalidade, flexibilidade, originalidade, interdisciplinaridade (PRADO, 2003) (PPD, 2012).

Mitchel Resnick, líder do projeto Scratch no Laboratório de Mídia do Instituto de Tecnologia de Massachusetts, propõe uma abordagem construcionista (RESNICK 2002). O construcionismo é uma abordagem proposta por Seymour Papert na qual o computador deve ser utilizado como ferramenta para o aluno construir o seu conhecimento ao desenvolver um produto. O construcionismo se opõe ao instrucionismo, perspectiva esta que se baseia na idéia de que o computador instrui o aluno a fazer algo. Tanto o construcionismo quanto o instrucionismo entendem que o computador se relaciona com o aprendiz através de um software com a mediação do professor (LIMA 2009).

Baseado no construcionismo e na pedagogia de projetos, costumo pedir como avaliação final do componente curricular de Introdução a Lógica de Programação, que os alunos desenvolvam, em grupo de no máximo 4 pessoas, um programa com o tema de livre escolha.

7. Considerações Finais

A abordagem histórico-cultural (REGO, 1994), (OLIVEIRA, 1992), (SANTOS, 2002) fornece a coerência teórica necessária que justificam às práticas adotadas. Precisamos levar em consideração a cultura destes alunos e os conhecimentos preexistentes como domínio da internet, vídeo games, sua formação do ensino médio, entre outros. A Pedagogia de Projeto se mostrou eficiente e motivadora, sendo capaz de proporcionar uma oportunidade de realização de uma aprendizagem significativa. Outro aspecto importante se refere ao desenvolvimento da autonomia e a leitura do mundo propostas por Paulo

Freire (FREIRE, 2005), (FREIRE, 2009) quando eles utilizam os softwares para expressar sua cultura com independência do professor.

O ambiente de desenvolvimento MIT-Scratch proporcionou facilidade de programação e experimentação de comandos que, a nosso ver, possibilitaram uma abordagem construcionista como propõe Resnick (2002) materializado pela metodologia da pedagogia de projetos. Esta forma particular de trabalhar os conteúdos deu-nos a oportunidade de vivenciar concretamente um processo educativo que transformou ideias iniciais em um produto de software.

É interessante notar que os temas escolhidos são diversos e mostram a cultura e personalidade dos alunos. Nesse aspecto destacamos os projetos Vida de Estudante e As leis de Newton. Em Vida de Estudante é apresentado conteúdos de matemática financeira, discutindo os gastos e o modo de vida dos estudantes da UFRB. Em As Leis de Newton, uma aluna do grupo passa a ser personagem da história, cedendo imagens para a realização do projeto. Deste modo, o projeto em MIT-Scratch também serviu como forma de expressão para estes alunos.

Referências

- ALMEIDA, M. E. B. de. *Como se trabalha com projetos* (Entrevista). Revista TV ESCOLA. Secretaria de Educação a Distância. Brasília: Ministério da Educação, SEED, n 22, março/abril, 2002.
- FORBELLONE, André Luiz Villar; EBERSPACHER, Henri Frederico. **Lógica de programação**. 3. ed. São Paulo: Makron Books, 2005. 197 p.
- FREIRE, F. M. P e PRADO, M. F. B. B. *Projeto Pedagógico: pano de fundo para escolha de um software educacional*. In: J.A. Valente (org.) O computador na Sociedade do Conhecimento. Campinas, SP: UNICAMP-NIED, 1999.
- FREIRE, P. **Pedagogia da autonomia: saberes necessários à prática educativa**. 39.ed. São Paulo: Paz e Terra, 2009. 148 p.
- FREIRE, Paulo. **Educação como prática da liberdade**. 28. ed. Rio de Janeiro: Paz e Terra, 2005. 158 p.
- LIMA M.R. *Construcionismo de Papert e ensino-Aprendizagem de programação de computadores no Ensino Superior*. 2009. 143 p. Dissertação (Mestrado) - Universidade Federal de São João Del-Rei, São João Del-Rei, MG.
- MACHADO, N. J. *Educação: projetos e valores*. São Paulo: Escrituras Editora, 2000.
- MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de. **Algoritmos: lógica para desenvolvimento de programação de computadores**. 22. ed. São Paulo: Érica, 2009. 240 p. MARQUES, T. M.O *Scratch dá o que falar: dos usos domésticos aos usos na escola*. Revista Educação e Matemática, Lisboa, número 96, 44-48, 2008.

OLIVEIRA, M.K. **O problema da Afetividade em Vygotsky**. In LA TAILLE, Yves; OLIVEIRA, M.K. & DANTAS, H. Piaget, Vygotsky e Wallon: Teorias psicogenéticas em discussão. 6ª ed. São Paulo: Summus, 1992.

PPD - Projetos Pedagógicos Dinâmicos. *Pedagogia de Projetos*. Disponível em: <http://www.projetopedagogicosdinamicos.com/projetos2.html>. Acesso em: 14 de maio, 2012.

PRADO, M. *Pedagogia de Projetos*. Serie “Pedagogia de projetos e Integração de Mídias” – Programa Salto para o Futuro, Setembro, 2003.

PRETTO, Nelson. Educação e inovação tecnológica: um olhar sobre as políticas públicas brasileiras. **Revista Brasileira de Educação**, Rio de Janeiro, n.11, p.75, Mai/Jun/Jul/Ago. 1999.

REGO, T.C. **Vygotsky: uma perspectiva histórico-cultural da educação**. 19. ed. Petrópolis: Vozes, c1994. 138 p.

RESNIK. M. *Rethinking Learning on the Digital Age*. 2002. Disponível em: <http://ilk.media.mit.edu/papers/mres-wef.pdf>. Acesso em maio de 2012.

RUSK, N., RESNICK, M., MALONEY, J. *Competências de aprendizagem para o séc. XXI*. Disponível em: <http://info.scratch.mit.edu/sites/infoscratch.media.mit.edu/files/file/translated-docs/CompetenciasdeaprendizagemSec21.pdf>. Acesso em: 14 maio, 2012.

SANTOS, M.C. **Algumas concepções sobre o ensino-aprendizagem da matemática**. Educação Matemática em Revista, ano 9, no 12, 2002.