

Uma dinâmica para ensino de conceitos fundamentais de programação

Flávia Linhalis Arantes¹, José Michael Leandro da Silva Ferreira²

¹Núcleo de Informática Aplicada à Educação (NIED)
UNICAMP – Campinas – SP – Brasil

²ProFIS – Programa de Formação Interdisciplinar Superior
UNICAMP – Campinas – SP – Brasil

farantes@unicamp.br, josefh.leandro@hotmail.com

Abstract. *In this article, we describe a dynamic named by students as the “dynamics of bis”, whose goal is to teach the concepts of variables, conditionals and concurrent threads in a more lightweight and fun. We describe two scenarios in which the dynamics was applied and evaluated by the students, trying to discuss positive and negative points of the scenarios. With this work, we aim to contribute to the teaching of the concepts mentioned, because the dynamic that can be applied in various scenarios and with different programming languages.*

Resumo. *Neste artigo, descrevemos uma dinâmica cunhada pelos próprios alunos como a “dinâmica do bis”, cujo objetivo é ensinar os conceitos de variáveis, condicionais e concorrência de uma maneira mais leve e divertida. Descrevemos dois cenários nos quais a dinâmica foi aplicada e avaliada pelos alunos, procurando discutir pontos positivos e negativos dos cenários. Com este trabalho, procuramos contribuir com o ensino dos conceitos citados, pois trata-se de uma dinâmica que pode ser aplicada em cenários variados e com diferentes linguagens de programação.*

1. Introdução

Projetos envolvendo a programação de computadores aplicada à educação tiveram o seu auge na década de 80, com a linguagem Logo [Papert 1980]. O uso do Logo pregava uma modificação estrutural quanto ao papel da tecnologia, colocando o computador como um poderoso aliado na construção e materialização do conhecimento. Atualmente, muitos voltaram a defender a ideia de que saber programar é uma habilidade importante na sociedade contemporânea. Não basta saber criar textos, usar redes sociais e usar a Internet, é preciso saber também como os computadores funcionam e estimular o desenvolvimento do pensamento computacional [Resnick et al. 2009].

Com o intuito de contribuir com a formação de uma geração que seja mais autônoma tecnologicamente, surgiu o Projeto Jovem Hacker¹ [Arantes et al. 2014]. As ideias e sugestões de um grande grupo de pessoas foram reunidas para organizar a estrutura do que seria um curso para jovens. A estrutura do curso foi proposta como uma série de oficinas modulares. A proposta buscava fomentar o interesse do jovem pela programação com vista a criar algo que fosse socialmente útil e pessoalmente gratificante. O consenso

¹<http://jovemhacker.org/>

apontou para um currículo que tinha como base (1) fazer uma introdução à computação e ao computador, com enfoque em software/hardware livre e o trabalho colaborativo e distribuído; (2) conceitos básicos de lógica de programação como forma preparatória para (3) a programação para web (HTML/CSS/Javascript), e finalizando com uma (4) linguagem de alto nível com aplicação além da web [Amiel et al. 2015].

A dinâmica descrita neste artigo foi realizada durante a oficina de Scratch² [Resnick et al. 2009], que consiste em um módulo do Projeto Jovem Hacker para ensinar conceitos básicos de programação. As próximas seções descrevem a dinâmica e dois cenários nos quais ela foi aplicada, bem como o resultado das avaliações dos alunos, pontos positivos e negativos dos cenários.

2. A dinâmica do bis

A dinâmica foi motivada pelas dificuldades que os alunos apresentaram para compreender alguns conceitos importantes de programação, durante a oficina de Scratch realizada na edição piloto do Projeto Jovem Hacker.

Nas primeiras aulas da oficina, fizemos a apresentação da interface do Scratch (atores, palcos, comandos de movimento, caneta, som e aparência), dos laços e condicionais. Os alunos exploraram livremente o Scratch e fizeram exercícios dirigidos.

Na metade da oficina, o conceito de condicionais com variáveis foi introduzido. O objetivo dessa aula era que eles resolvessem um problema usando uma variável com condicional. No início da aula, mostramos um exemplo e fizemos juntos no Scratch. Em seguida, deixamos que resolvessem sozinhos outro exemplo similar. Entretanto, o objetivo não foi cumprido a contento, pois houve dificuldade em posicionar a variável no programa e fazer a condição. Percebemos também que o tempo usado para sincronizar os atores no Scratch estava confundindo os alunos. Além disso, eles não sabiam sobre concorrência de execução, conceito importante quando temos mais de um *thread* no programa. Diante dessas dificuldades, decidimos por introduzir uma dinâmica para mostrar esses e outros conceitos de uma maneira mais divertida.

Os conceitos abordados na dinâmica foram: variáveis, condicionais e concorrência de *threads*. Além disso, aproveitamos para explicar um pouco sobre a organização básica dos computadores: a função do processador, da memória, do sistema operacional e do barramento de dados.

A dinâmica é feita em 2 partes. A primeira parte consiste em colocar os comandos em Scratch (ou programas) na “memória”. Os materiais necessários para essa parte são os seguintes: comandos preferencialmente em cartolina ou EVA colorido de acordo com as cores de cada comando no Scratch, fita crepe para colar os comandos, espaço de “memória”, representado pela lousa ou por uma parede. Pequenos potes de plástico transparentes são utilizados para representar o espaço de memória das variáveis. O método para realizar a primeira parte é distribuir os comandos entre os alunos para que os mesmos possam montar os programas na memória. Essa fase da dinâmica é importante para os alunos fixarem os comandos e pensarem na lógica que será usada para compor o programa.

A segunda parte consiste em simular a execução do programa. O instrutor faz o papel de processador e alterna a execução de cada *thread* concorrente. As variáveis são

²<https://scratch.mit.edu/>

incrementadas com chocolate bis, o que dá um ar de descontração à dinâmica. Com isso, sempre que uma variável é incrementada, um chocolate bis é colocado no pote plástico que representa o espaço de memória daquela variável. Similarmente, se a variável for decrementada, o chocolate bis é retirado do pote. O objetivo é mostrar o que acontece com cada ator e com as variáveis em cada ciclo de execução. Além disso, o instrutor pode mostrar o papel do processador, da memória, do barramento de dados e do sistema operacional. No final da dinâmica, os chocolates bis são distribuídos para os alunos.

3. Os cenários

A dinâmica descrita na seção anterior foi aplicada em dois cenários, os quais são descritos nas próximas subseções.

3.1. Cenário 1 – Computação Desplugada ao ar livre

A dinâmica foi aplicada pela primeira vez durante a realização do piloto do Projeto Jovem Hacker, no dia 06 de maio de 2014, no espaço do CIS-Guanabara em Campinas³. Cinco jovens de escolas públicas, com idades entre 16 e 18 anos, participaram da dinâmica. Os jovens eram usuários de Internet, redes sociais e jogos, com acesso a computadores e dispositivos móveis, mas com praticamente nenhum conhecimento sobre como o hardware funciona ou como os softwares são desenvolvidos.

O objetivo da dinâmica foi fazê-los compreender conceitos que geraram dúvida no encontro anterior – variáveis, concorrência na execução de *threads* e condicionais. Como não tínhamos lousa e giz, fizemos uma dinâmica ao ar livre, procurando mostrar esses conceitos de uma maneira mais descontraída, utilizando a computação desplugada [Bell et al. 2009]. A computação desplugada consiste em realizar atividades com o objetivo de ensinar os fundamentos da Ciência da Computação sem a necessidade de computadores. Uma vantagem dessa abordagem reside na sua independência de recursos de hardware ou software, bem como na possibilidade de abordar temas de computação de uma maneira lúdica e mais divertida. A Figura 1 mostra o cenário utilizado.

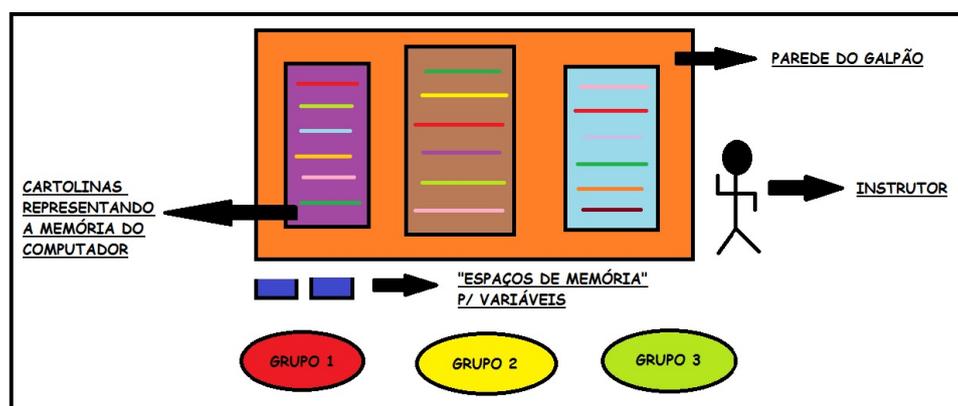


Figura 1. Primeiro cenário da dinâmica – galpão ao ar livre.

O exemplo que fizemos usava três *threads* concorrentes, pois havia três atores, cada um com seu *script*. Inicialmente, explicamos que faríamos uma dinâmica para si-

³<http://www.cisguanabara.unicamp.br/>

mular a execução dos três *scripts* (arara, cão e menina) que fizemos na aula anterior. O objetivo era entender o que cada ator fazia em cada segundo de execução.

Colamos cartolinas nas paredes para simular o espaço de memória onde os programas iriam executar. Na primeira parte da dinâmica, os alunos colaram os três *scripts* na memória. Para realizar essa tarefa, os alunos foram divididos em três duplas, cada dupla foi responsável por montar o *script* de um ator (um dos alunos fez dupla com o auxiliar da oficina). As duplas receberam tiras de cartolina colorida, onde estavam escritos os comandos correspondentes ao *script* do seu ator. Eles conseguiram realizar a tarefa sem muita ajuda dos instrutores. A Figura 2 mostra os *scripts* de cada ator. Nesse exemplo, a menina diz “Olá, quando o cão latir, ajude a arara a voar”. Depois de 5 segundos, o cão começa a latir e a arara voa, acompanhando o ponteiro do mouse. Quando o cão late pela décima vez, a arara diz “Cuidado!”. Em seguida, o cão começa a andar em direção à menina, que diz “Socorro!!!” e sai correndo. É uma animação simples, na qual os atores estão sincronizados com tempo e são usadas duas variáveis, uma para contar os latidos do cão e outra para contar os passos do cão.

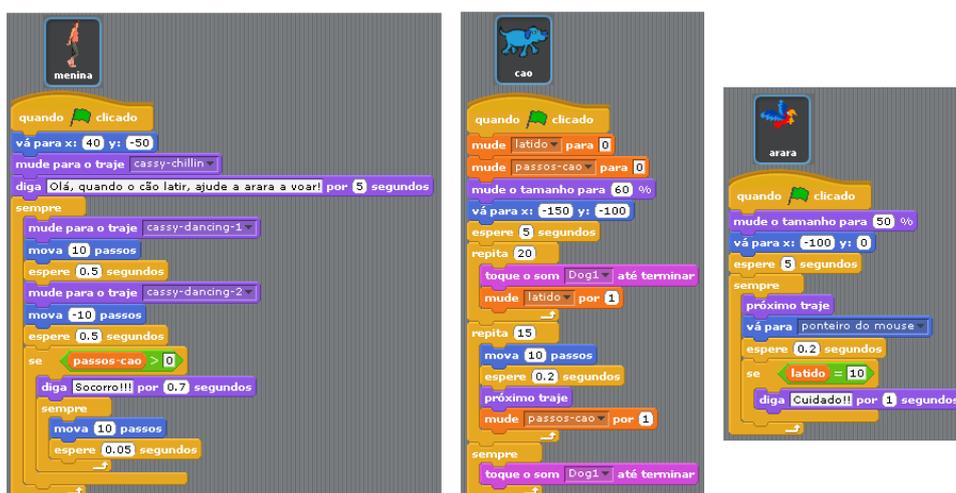


Figura 2. Scripts de cada ator para o primeiro cenário da dinâmica.

Depois dos scripts prontos (alocados na “memória”), iniciamos a segunda parte da dinâmica – a simulação da execução do programa. Um dos instrutores fez o papel de processador. Nesse exemplo, o processador dava um segundo de cada vez para o script de cada ator. As variáveis foram incrementadas com chocolate bis. Com isso, mostramos o que acontecia com cada ator e com as variáveis em cada segundo de execução.

3.2. Avaliação da dinâmica no cenário 1

Realizamos uma avaliação no último encontro da oficina com o objetivo de registrar a opinião, as dificuldades e as preferências dos jovens. A avaliação aconteceu em dois momentos. No primeiro, fizemos uma roda da conversa com todos os presentes. No segundo momento, distribuimos um questionário para eles responderem.

Nesta seção, descrevemos as impressões dos jovens relacionadas apenas ao dia da dinâmica, mais detalhes sobre outros itens avaliados e sobre a oficina no geral podem ser encontrados em Arantes e colegas [Arantes et al. 2014].

Durante a roda da conversa, os alunos disseram que a maior dificuldade que tiveram foi criar a condição utilizando a variável e posicioná-la corretamente no programa. Explicamos que essa dificuldade é considerada normal quando se está aprendendo lógica de programação. Eles ficaram conosco 1 hora e 30 minutos, durante 8 encontros – são apenas 12 horas para aprender lógica de programação. Explicamos que, para aprender programação é preciso estudar, praticar, dedicar-se em casa.

A dinâmica foi feita justamente por termos percebido a dificuldade supracitada. Ao perguntar se a dinâmica ajudou a compreender variáveis e condicionais, as respostas foram bem positivas. Um dos alunos afirmou que “foi melhor para explicar, porque no computador você só explicava com voz e aqui você mostrou”. Além disso, eles mencionaram a dinâmica, referida como “o dia do bis”, como sendo o dia que mais gostaram.

3.3. Cenário 2 – Laboratório com computadores

O fato do “dia do bis” ter sido avaliado como o encontro que os alunos mais gostaram durante o piloto do Projeto Jovem Hacker, nos motivou a realizar a dinâmica novamente na edição de 2015 do Projeto.

A dinâmica foi aplicada no mesmo momento da oficina anterior, isto é, ao introduzir o conceito de variáveis e condicionais. Na edição de 2015, a dinâmica foi realizada no dia 17 de junho, no espaço do CDI Campinas⁴. Treze jovens, com idades entre 13 e 17 anos participaram. Os jovens tinham um perfil parecido com os da edição piloto. Apenas dois deles disseram ter tido contato prévio com programação – um havia feito uma oficina de Scratch na escola e outro ouvia explicações do pai sobre programas na linguagem C.

Diferente do cenário da edição piloto, as aulas da oficina foram realizadas em um laboratório amplo, com projetor, lousa, pincel e um bom espaço para os instrutores circularem. Por outro lado, o espaço externo era bem limitado. Por isso, decidimos por realizar a dinâmica no próprio laboratório. Os alunos permaneceram sentados, de frente para a lousa e de costas para os computadores, outro ponto que diferiu do cenário anterior, onde os alunos estavam de pé. A Figura 3 ilustra o cenário utilizado.

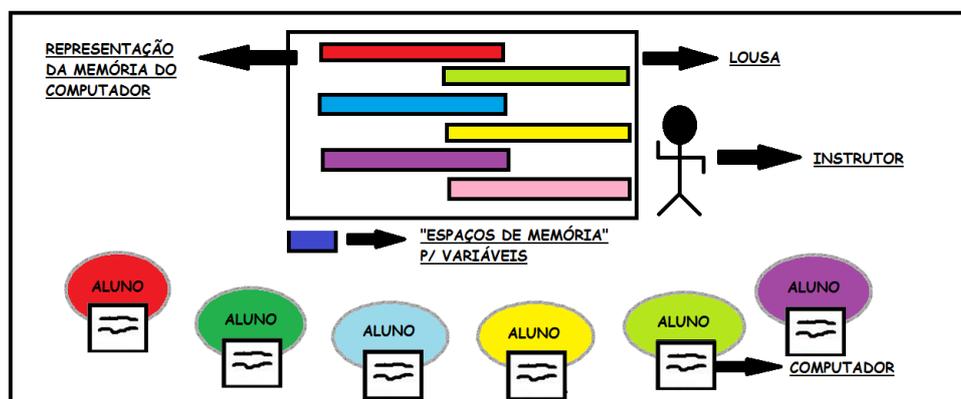


Figura 3. Segundo cenário da dinâmica – laboratório com computadores.

O exemplo utilizado foi o jogo do Pong, no qual uma bolinha fica percorrendo a tela e não pode cair no chão. O jogador controla uma plataforma com o mouse e, sempre

⁴<http://www.cdicampinas.org.br/>

que a bolinha quicar sobre a plataforma, o jogador marca um ponto. Se a bolinha cair no chão, o jogo acaba. Mais uma vez, havia três *threads* concorrentes, mas todos pertenciam ao mesmo ator (a bolinha, no caso). A Figura 4 mostra a tela do jogo e os *scripts*.

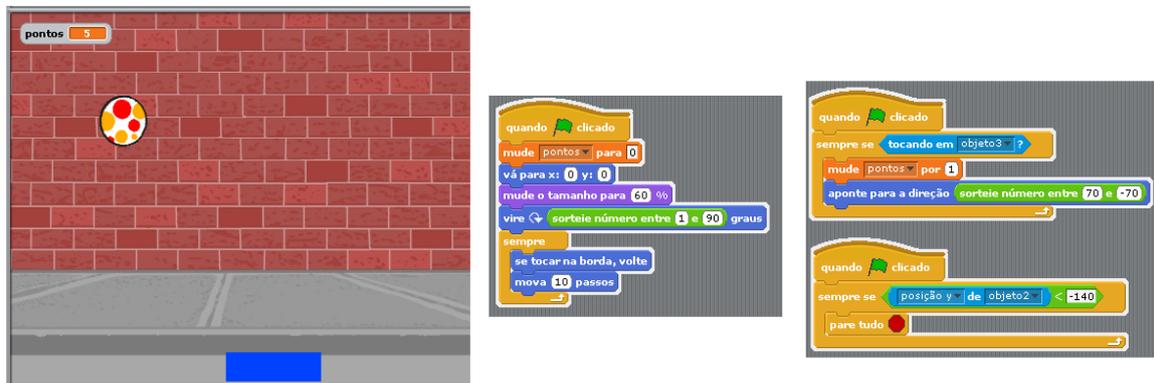


Figura 4. Tela do jogo Pong e os *scripts* da bolinha.

Na primeira parte da dinâmica, a instrutora colou os *scripts* na memória (lousa) com a ajuda dos alunos. Para realizar essa tarefa, os alunos se dividiram em quatro grupos – dois grupos com quatro integrantes cada, um grupo com três integrantes e um grupo com dois integrantes. Os grupos receberam tiras de EVA coloridas, onde estavam escritos os comandos correspondentes ao *script* da bolinha. A instrutora perguntava qual comando eles achavam que deveria ser utilizado para realizar os movimentos da bolinha. O grupo que tinha o comando consigo se manifestava e juntos – alunos e instrutora – montaram os três *scripts* da bolinha. Nesse exemplo, o primeiro *script* é responsável por fazer a bolinha se mover aleatoriamente pela tela; o segundo verifica se a bolinha está tocando na plataforma – se sim, o jogador ganha um ponto; o terceiro verifica se a bolinha tocou no chão – se sim, o jogo acaba. Trata-se de um exemplo simples, no qual o ator possui três *threads* concorrentes e uma variável é utilizada para realizar a contagem dos pontos.

Depois dos *scripts* prontos alocados na “memória”, iniciamos a segunda parte da dinâmica – a simulação da execução do programa. A instrutora fez o papel de processador e aproveitou para explicar o papel de outros elementos importantes de um sistema computacional – processador, memória, entrada e saída, barramento e sistema operacional. Nesse exemplo, o processador executava uma iteração do laço de cada *thread* por vez. Assim como no primeiro cenário, a variável foi incrementada com chocolate bis. Com isso, foi possível mostrar o que acontecia com a bolinha e com a variável a cada iteração do laço de cada *thread* concorrente.

3.4. Avaliação da dinâmica aplicada ao cenário 2

A avaliação da dinâmica foi realizada no final da aula, por meio de um questionário. Ao perguntar se dinâmica do bis ajudou a compreender o conceito de variável, 12 alunos responderam que sim e apenas 1 respondeu que ajudou “mais ou menos”. Algumas respostas que nos chamaram a atenção foram as seguintes: “*Sim, pois foi bem explicado, gostei muito da aula!*”, “*Sim, pois foi um método diferente para entender a variável*”, “*Sim, foi bem diferente chamando a atenção e facilitando a compreensão*”. Esses são exemplos de respostas que evidenciam que uma abordagem diferente da tradicional motiva a participação e prende mais a atenção dos alunos.

A dinâmica do bis também introduziu outros conceitos importantes em computação. Foi perguntado aos alunos quais desses conceitos foram compreendidos; os resultados estão no gráfico apresentado na Figura 5(a). Depois da dinâmica, houve um intervalo e, em seguida, os alunos praticaram exercícios com variáveis. Perguntamos como eles se saíram na resolução dos exercícios; as respostas estão no gráfico da Figura 5(b).

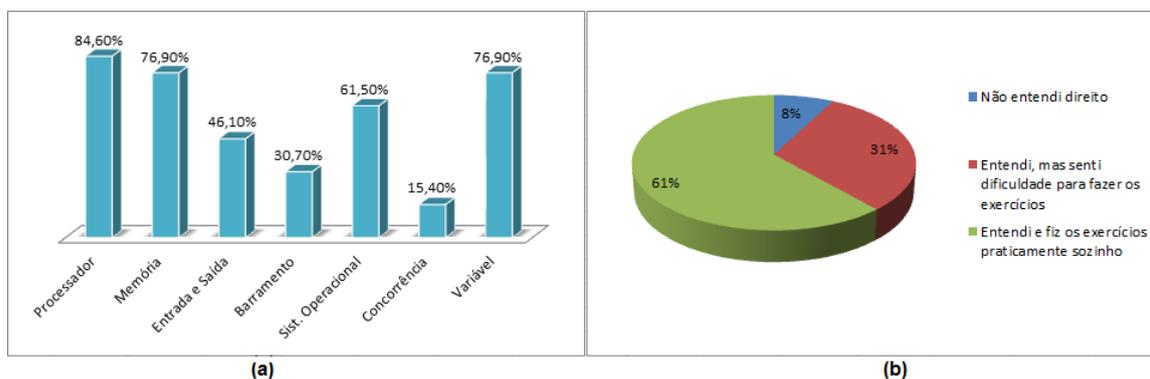


Figura 5. Gráficos mostrando as respostas dos alunos sobre a avaliação dos conceitos aprendidos com a dinâmica do bis.

4. Discussão

A dinâmica foi realizada em dois cenários diferentes, com fatores e condições que permitiram identificar aspectos que afetam a qualidade da implementação. O acumulado de experiências, a cada nova aplicação, leva a um conhecimento mais abrangente e integral da dinâmica e do contexto. Sendo assim, a dinâmica em si não é uma proposta imutável, sendo alvo de investigação constante. Nesta seção, procuramos refletir cada cenário de implementação, com um olhar para os contextos, limitações e diferenças de cada um, que vai além da avaliação feita com os alunos.

O laboratório que foi utilizado para realização da oficina piloto tinha algumas limitações, principalmente com relação ao espaço físico, falta de Internet e falta de lousa e pincel. Isso nos levou a realizar a dinâmica do primeiro cenário (descrito na seção 3.1) ao ar livre, com os alunos de pé e completamente “desplugados”. Aspectos que inicialmente foram vistos como limitações, passaram a ser pontos positivos para a realização da dinâmica, pois os alunos se mostraram mais descontraídos e com mais foco, quando comparados aos alunos no cenário 2. No segundo cenário não podemos dizer que estávamos desplugados. Apesar dos alunos estarem de costas para o computador, eles permaneceram sentados e com a possibilidade de distrair a atenção com celulares durante a dinâmica. Por esse motivo, o cenário desplugado se mostrou mais interessante para realização da dinâmica – procuraremos experimentar outros cenários similares nas próximas edições.

Outro diferencial entre os dois cenários foi o momento de colocar o programa na memória. No cenário 1 os alunos estavam em duplas, receberam os trechos dos programas e tentaram montar o programa sem a ajuda dos instrutores. Esse foi um ponto positivo porque os alunos se envolveram mais com a construção do programa. No cenário 2 o número de alunos era maior e o espaço que podia ser utilizado como “memória” era menor. Por esse motivo, a instrutora montou os programas na lousa, sempre perguntando aos alunos qual comando utilizar. Nessa situação, alguns alunos não participavam porque

viram que outros já estavam respondendo. Além disso, o fato da instrutora estar montando o programa, tirou deles essa responsabilidade. Para as próximas edições, deixaremos os alunos montarem os programas em grupos pequenos (duplas de preferência).

Por fim, vale observar pelas Figuras 2 e 4 que no cenário 2 a condição não depende do valor da variável, diferente do cenário 1, onde o comportamento do programa é diretamente influenciado pelos valores das variáveis que estão nas condições. Apesar da sincronização dos *threads* no exemplo do cenário 1 não ser a ideal por ter sido feita com tempo e não com eventos, o referido exemplo foi mais proveitoso porque possibilitou mostrar condicionais em conjunto com variáveis, o que é muito comum em programação.

5. Conclusões

Neste artigo, descrevemos a dinâmica do bis, como foi cunhada pelos próprios alunos. A dinâmica tem como objetivo ensinar conceitos fundamentais de programação – variáveis, condicionais e concorrência. Durante a dinâmica, é importante mostrar que os programas e as variáveis são alocados na memória, processados pelo processador, e que os valores das variáveis são lidos, escritos e trazidos para o processador, por meio do barramento, para realizar operações lógicas e aritméticas. Para tornar a dinâmica mais divertida e promover a socialização, as variáveis foram incrementadas com chocolate bis.

Com a dinâmica do bis abordamos conceitos de programação, muitos deles avançados, de uma maneira leve e agradável. A dinâmica proporcionou socialização e descontração ao grupo, o que vai de encontro a um dos objetivos do Projeto Jovem Hacker, que é fomentar o interesse do jovem pela programação.

Com este artigo, esperamos contribuir com o ensino dos conceitos supracitados de uma maneira mais divertida. A dinâmica do bis foi realizada durante as oficinas de Scratch do Projeto Jovem Hacker. Trata-se de uma dinâmica flexível que pode ser reproduzida em outros projetos, cenários e com outras linguagens de programação.

Referências

- Amiel, T., de Souza Fedel, G., Arantes, F. L., and Aguado, A. G. (2015). Dominando para não ser dominado: Autonomia tecnológica com o projeto jovem hacker. In *XVI Workshop Internacional de Software Livre (WSL 2015). Evento realizado em conjunto com o XVI FISL. Porto Alegre, Brasil.*, pages 1–13.
- Arantes, F. L., Amiel, T., and Fedel, G. (2014). Nos rumos da autonomia tecnológica – desafios e lições aprendidas para a formação de jovens. In *Anais do XX Workshop de Informática na Escola.*, pages 1–10, Dourados, MS. Retrieved from <http://www.br-ie.org/pub/index.php/wie/article/view/3113>.
- Bell, T., Alexander, J., Freeman, I., and Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13(1):20–29.
- Papert, S. (1980). *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books, Inc., New York, NY, USA.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y. (2009). Scratch: Programming for all. *Commun. ACM*, 52(11):60–67.