

Nos rumos da autonomia tecnológica ó desafios e lições aprendidas para a formação de jovens

Flávia Linhalis Arantes¹, Tel Amiel¹, Gabriel Fedel²

¹Núcleo de Informática Aplicada à Educação (NIED) ó Universidade Estadual de Campinas (UNICAMP) ó Campinas ó SP ó Brasil

²FATEC Americana ó Campinas ó SP ó Brasil

farantes@unicamp.br, tamiel@unicamp.br, fedel@fedel.net.br

Abstract. *The generation that some call "digital natives" can demonstrate a fluid relationship with the use of various devices. However, they tend to ignore how such devices work. The Jovem Hacker Project aims to investigate ways to assist in the education of a generation that can have a more autonomous relationship with technology. In this article, we report our first experiences in the context of the Jovem Hacker Project, based on a pilot using primarily Scratch as a programming language. We discuss lessons learned and challenges towards the technological empowerment of young people. Through further pilots, we hope to devise a model for youth training that will move beyond Scratch to more powerful languages and understanding of the workings of our technological society.*

Resumo. *A geração que alguns atualmente denominam õnativos digitaisõ demonstra uma relação fluida com o uso de dispositivos diversos, entretanto, tendem a ignorar como tais dispositivos funcionam. O Projeto Jovem Hacker tem como objetivo investigar maneiras de auxiliar na formação de uma geração que seja autônoma tecnologicamente. Neste artigo, relatamos nossa a primeira experiência no contexto do Projeto Jovem Hacker, por meio da realização de um piloto com a linguagem Scratch. Apontamos lições aprendidas e desafios rumo ao empoderamento tecnológico de jovens. Esperamos, com isso, desenhar um modelo inicial para formação de jovens em direção à autonomia tecnológica, para além do Scratch, com linguagens mais poderosas e um melhor entendimento da nossa sociedade tecnológica.*

1. Introdução

O contato com as novas mídias dentro e fora do ambiente escolar acontece cada vez mais cedo, e já é reconhecido o interesse em explorar o uso da tecnologia no ensino infantil [Périssé & Garboggini 2004; 2005; Flannery et. al. 2013]. As crianças crescem cercadas de dispositivos como *tablets*, *smartphones* e conectadas às redes sociais. Entretanto, a competência tecnológica de jovens e adolescentes tem sido colocada em cheque, particularmente no que tange à denominação de uma geração como õnativaõ digital [Amiel & Amaral 2013; Oblinger & Hawkins 2006]. Apesar da relação mais fluida com o uso de dispositivos, sabemos que existem grandes desafios em ensinar jovens sobre outros temas essenciais. Estes incluem a dimensão ética do uso das tecnologias, além de noções sobre privacidade e exposição midiática. No que tange este projeto,

buscamos entender outro desafio de importância: como explicar o interior, o funcionamento de dispositivos e a lógica de seu funcionamento através da programação. No entanto, pouco sabemos sobre quais são exatamente esses desafios, particularmente quando buscamos ir além o uso de linguagens voltadas para o público mais jovem.

Com o intuito de investigar esses desafios e contribuir com a formação de uma geração que seja mais capacitada tecnologicamente, surgiu o Projeto Jovem Hacker¹. Sendo assim, podemos dizer que o objetivo do Projeto Jovem Hacker é investigar maneiras de auxiliar na formação de uma geração que seja autônoma tecnologicamente e se empodere dos rumos da nossa sociedade. Para isso, o projeto irá compartilhar conhecimento nas áreas de programação de computadores, software livre (trabalhando a dimensão ética) e trabalho colaborativo para jovens de escolas públicas.

O empoderamento tecnológico é fundamental para evitar uma grande massa de usuários que seja dependente da tecnologia sem entender seu funcionamento, além de ser um passo que pode incentivar esses jovens a se tornarem futuros profissionais da área de TI, uma área com grande déficit de mão de obra [Venceslau 2013].

A formação no Projeto Jovem Hacker está distribuída em módulos e tem seu foco no aprendizado através de projetos práticos e colaborativos. Inicialmente, o Projeto prevê utilizar a ferramenta Scratch² para ensinar lógica de programação. Em seguida, pretendemos introduzir o desenvolvimento web com HTML e CSS. O Projeto prevê ainda ensinar a base de uma linguagem de programação como Python. Oficinas sobre *apps* para celular, arduino³ e outras tecnologias são interessantes e desejadas.

Além do objetivo de incentivar a autonomia tecnológica, mais especificamente a programação, o Projeto Jovem Hacker busca encontrar formas alternativas de construir tal conhecimento. Partindo de referências construídas de maneira coletiva e descentralizada, tais como Transparência Hacker⁴, Ônibus Hacker⁵, Laboratório Hacker da Câmara Legislativa⁶ e mesmo Hackatons/Hackatonas, o projeto buscará transferir aos estudantes a motivação do conhecimento construído colaborativa e compartilhadamente.

Neste artigo, relatamos nossa primeira experiência no contexto do Projeto Jovem Hacker, por meio da realização de um piloto. O intuito do piloto foi investigar o conhecimento, as preferências, os interesses e as competências dos jovens sobre o tema iniciação em programação, além de compor desafios e direcionamentos para o Projeto. A contribuição deste artigo é, portanto, apresentar lições aprendidas e apontar desafios rumo ao empoderamento tecnológico de jovens. Esperamos, com isso, desenhar um modelo inicial de projeto para formação de jovens no que diz respeito à tecnologia que o circunda.

¹ <http://jovemhacker.org>

² <http://scratch.mit.edu/>

³ <http://www.arduino.cc/>

⁴ <http://www.canalibase.org.br/a-ideologia-por-tras-da-transparencia-hacker/>

⁵ <http://onibushacker.org/>

⁶ <http://www2.camara.leg.br/transparencia/laboratorio-hacker>

2. Trabalhos Relacionados

Projetos envolvendo a programação para crianças tiveram o seu auge no uso da linguagem Logo [Papert 1980]. Para além da programação e raciocínio lógico, o uso do sistema Logo pregava uma modificação estrutural quanto ao papel da tecnologia, colocando o computador sob o controle das crianças, transformando-o em um poderoso aliado na construção e materialização do conhecimento criado.

Atualmente, muitos voltaram e 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. As iniciativas mais recentes para incentivar o ensino de programação para crianças surgiram nos Estados Unidos, com projetos como `code.org` e `codeyear.com`, pois se estima que em poucos anos falte mão de obra qualificada para atuar no mercado de trabalho americano na área de programação.

A necessidade de aumentar o conhecimento sobre o funcionamento das novas tecnologias é um tópico já abordado inclusive por instituições como a Mozilla Foundation⁷, que desenvolvem projetos como o *Webmaker*⁸, que visa desenvolver ferramentas e metodologias para ensinar o uso e o funcionamento da web.

O objetivo da nossa iniciativa com o Projeto Jovem Hacker não é necessariamente contribuir com o futuro do mercado de trabalho. Entretanto, se isso acontecer, será bem vindo em um mercado com déficit de mão de obra. Acreditamos que saber programar não seja útil apenas para ser um programador no futuro. A programação ajuda em outras habilidades, como resolver problemas, desenvolver o raciocínio lógico, o que pode contribuir com o cidadão em diversos momentos de sua vida. Saber programação pode possibilitar, inclusive, desenvolver soluções de software para problemas do dia a dia, ou mesmo dar manutenção em softwares que utiliza.

No Brasil surgiram outras iniciativas que são basicamente extensões dos projetos americanos. Um exemplo é o Projeto CodeClub⁹, cujo objetivo é oferecer a oportunidade para crianças aprenderem a programar. Para isso, o projeto oferece material de ensino para uma rede de voluntários apoiar a realização de atividades extracurriculares ligadas à programação de computadores. Os voluntários atuam em escolas próximas das casas dos alunos, uma hora por semana, orientando projetos. A base do projeto é ensinar crianças a programar através de jogos, animações e páginas da Internet. A primeira parte do projeto utiliza Scratch para ensinar as bases de programação de computadores. Em seguida, a proposta é introduzir o desenvolvimento web com HTML e CSS. No terceiro e último momento, o projeto prevê ensinar Python.

Um ponto em comum que temos com o Projeto CodeClub são as linguagens e o fato de ensinar programação como atividade extracurricular. Entretanto, nosso público é diferente. Trabalhamos com jovens, enquanto o CodeClub tem seu foco em crianças. Os jovens podem reter a atenção por mais tempo, o que nos abre as portas para fazer oficinas que duram uma manhã inteira, ou mesmo um dia inteiro, no estilo *hackathon*.

⁷ <https://www.mozilla.org/en-US/foundation/>

⁸ <https://webmaker.org/>

⁹ <http://codeclubbrasil.org>

Além disso, pelo fato dos jovens já terem estudado matérias que as crianças ainda não estudaram, temos a possibilidade de realizar práticas e dinâmicas com conteúdos mais aprofundados e diversificados.

Outro diferencial do nosso projeto com relação ao CodeClub é que nossa proposta é oferecer uma formação aos jovens em pouco tempo, com o auxílio de práticas descentralizadas e colaborativas. Enquanto no CodeClub os clubes podem se estender por anos, apenas com encontros presenciais.

Outro projeto similar no Brasil é o Ano do Código¹⁰. Assim como o CodeClub, trata-se de um projeto inspirado em iniciativas dos Estados Unidos, cujo objetivo é aproximar a programação aos interessados por tecnologia.

Um ponto que temos em comum com o Projeto Ano do Código é que o público alvo deles também são os jovens. Outro ponto em comum diz respeito a oferecer uma formação básica para o jovem em pouco tempo. Entretanto, em nossa proposta, essa formação acontecerá também com encontros presenciais, enquanto no Ano do Código a formação acontece por meio da disponibilização de conteúdo e com a ajuda de parceiros via redes sociais.

3. Desenvolvimento do Piloto

O projeto Jovem Hacker foi construído a partir da motivação e esforço colaborativo de alunos, pesquisadores, programadores e entusiastas da sociedade civil e da universidade. Através de múltiplos encontros e discussões online, buscamos definir o escopo de um curso de curto prazo para jovens, com os objetivos detalhados acima.

Em sua versão de planejamento, o curso foi desenhado para ter dois momentos divididos em módulos semanais. Um módulo comum, ou básico que contemplaria noções básicas de hardware e lógica de programação. O segundo momento seria focado em projetos de interesse específico dos jovens, que desenvolveriam um projeto sozinhos ou em grupo, e contariam com um mentor durante todo o processo. A meta, ambiciosa, era tentar avaliar até que ponto seria possível introduzir de maneira básica, um jovem ao mundo da programação de alto nível em um curto período (2-4 meses).

Não foi possível realizar o projeto nos moldes acima, pois logo nos primeiros encontros com os jovens percebemos a limitação do nosso modelo. Foi necessário diminuir o escopo e focar em noções básicas por mais tempo do que estimado. Utilizamos esse piloto para testar os limites do que pretendíamos realizar, recebendo constante *feedback* dos jovens e adaptando nossos encontros de acordo com o *feedback*.

No restante desta seção, relatamos a realização do piloto com a linguagem Scratch, entre abril e maio de 2014, no espaço do CIS-Guanabara¹¹ em Campinas. Nove jovens de escolas públicas, com idades entre 16 e 18 anos, participaram da oficina de Scratch. Foram 8 encontros semanais, com duração de 1 hora e 30 minutos cada. A seguir, apresentamos os encontros que realizamos e os objetivos de cada um.

¹⁰ <http://www.anodocodigo.org.br>

¹¹ <http://www.cisguanabara.unicamp.br/>

Encontro 1 ó Apresentação e motivação

Nosso objetivo no primeiro encontro foi conhecê-los melhor e explicar um pouco sobre o Projeto Jovem Hacker. Primeiramente, cada um se apresentou e eles falaram sobre o uso que faziam do computador. Todos disseram ter o costume de usar jogos e redes sociais, em especial o Facebook. Todos têm computador em casa, três deles têm *laptop*, dois têm *tablet* e sete têm *smartphone*. Apenas dois fizeram cursos de computação ó um de Linux e outro de *web design*.

Depois dessa conversa inicial, mostramos um vídeo como motivação a aprender programação. No vídeo, diversos famosos como Bill Gates, Mark Zuckerberg, e outros, falam sobre programação e sobre os primeiros programas que desenvolveram¹².

Em seguida, fizemos uma breve apresentação do Scratch, como software para iniciantes terem um primeiro contato com a lógica de programação [Resnick 2009]. Apresentamos alguns dos principais comandos, o palco, os atores. Mostramos ainda uma sequência de comandos simples, com repetição. Por fim, mostramos algumas animações prontas em Scratch.

A turma se apresentava com o perfil que esperávamos: jovens 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. Parte da motivação consistiu em mostrar aos alunos que esse é um conhecimento útil, mesmo para aqueles que não pretendem seguir carreira na computação. Também foi parte da motivação desmistificar a ideia de que programação é difícil.

No final do primeiro encontro, perguntamos se eles se sentiam motivados a continuar e por que. As respostas foram todas positivas. Transcrevemos duas delas:

õSim, eu me sinto motivada porque eu sei mexer nas redes sociais, mas gostaria muito de saber tudo sobre um computador. Afinal estamos na era da tecnologia e por isso gostei muito da aulaõ.

õSim, me sinto motivado a continuar, pois é uma coisa que me chama a atenção e algo que desejo para meu futuroõ.

Encontro 2 ó Primeiro contato com Scratch

O primeiro contato com o Scratch foi feito com base no material do Projeto CodeClub¹³. Nosso objetivo era explorar o Scratch e os comandos iniciais de movimento e laços.

Inicialmente, fizemos a apresentação da interface do Scratch (*scripts*, atores e cenários). Deixamos os alunos explorarem livremente e, em seguida, pedimos para fazerem um programa com três personagens pré-definidos (papagaio, cão e menino).

O encontro foi proveitoso e o objetivo foi cumprido, entretanto, achamos que era preciso praticar mais laços e falar sobre algoritmos. Nosso objetivo para o próximo

¹² O que você quer ser quando crescer: Programador. Vídeo disponível em <https://www.youtube.com/watch?v=L06AIFBRXls>. Acesso em 14/07/2014.

¹³ Material disponível em <http://codeclubbrasil.org/material/>

encontro, portanto, seria ensiná-los a identificar situações onde era preciso usar um laço e mostrar como utilizá-lo corretamente.

Encontro 3 ó Aula expositiva com noções de algoritmos

Fizemos uma aula expositiva, cujo objetivo era mostrar algoritmos com laços e condicionais. Para tanto, fizemos juntos alguns algoritmos, tais como trocar pneu e abrir uma porta. Em seguida, pedimos para eles fazerem um *script* com repetição no Scratch.

O objetivo da aula foi cumprido, pois eles fizeram o *script* sem muita ajuda e entenderam a diferença entre os dois principais laços do Scratch.

Encontro 4 ó Scratch com variáveis e condicionais

O objetivo dessa aula era que eles resolvessem um problema usando numa variável com condicional. No início da aula, mostramos um exemplo e fizemos juntos no Scratch. Em seguida, deixamos que resolvessem sozinhos outra condicional com variável.

Dessa vez o objetivo não foi cumprido a contento, pois houve dificuldade em posicionar a variável no *script* 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 paralelismo de execução, conceito importante quando temos mais de um *thread* no programa. O exemplo que fizemos usava três *threads*, pois havia três atores, cada um com seu *script*, e todos executavam ao mesmo tempo.

Encontro 5 ó Dinâmica ao ar livre

O objetivo desse encontro foi fazê-los compreender conceitos que geraram dúvida no encontro anterior, tais como variáveis, escopo de variáveis, paralelismo na execução de *threads*, condicionais, e como o tempo influencia na execução dos *threads*. Como não tínhamos lousa e giz, fizemos uma dinâmica ao ar livre, procurando mostrar esses conceitos de uma maneira lúdica.

Inicialmente, explicamos que faríamos uma dinâmica para simular a execução dos três *scripts* (papagaio, cão e menino) que fizemos na aula passada. O objetivo era que eles entendessem o que cada um dos atores fazia em cada segundo de execução.

Colamos cartolinas nas paredes para simular o espaço de memória onde os programas iriam executar. Aproveitamos para explicar um pouco sobre a organização básica dos computadores: a função do processador, do *clock*, da memória, dos registradores e dos dispositivos de entrada e saída.

A primeira parte da dinâmica consistiu em colocar os três *scripts* na memória. Para realizar essa tarefa, os alunos foram divididos em três grupos, cada grupo foi responsável por montar o *script* de um ator. Os grupos receberam tiras de cartolina, onde estavam escritos os comandos correspondentes ao *script* do seu ator. Eles conseguiram realizar a tarefa sem muita ajuda dos instrutores. A Figura 1 mostra os alunos colando os *scripts* na cartolina que representa o espaço de memória.

Depois dos *scripts* prontos, iniciamos a segunda parte da dinâmica ó a simulação da execução do programa. Um dos instrutores fez o papel de processador e o outro fez o papel da memória. 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.



Figura 1 É Alunos colando os *scripts* nos í espaços de memória.

A dinâmica foi bem proveitosa e esclarecedora em vários pontos, tais como mostrar o papel do processador e da memória, mostrar leitura e escrita de variáveis, mostrar que cada *script* tem um tempo para execução no processador, mostrar como o tempo influencia na sincronização do programa. Em um dos depoimentos, um aluno afirmou que *foi melhor para explicar, porque no computador você só explicava com voz e aqui você mostrou*.

Encontro 6 ó Um jogo com Scratch

O objetivo desde encontro foi mostrar um jogo com Scratch e motivá-los a desenvolver seus próprios jogos. A instrutora apresentou o jogo ó criar um fantasma que fugia de uma bruxa. A cada segundo sem ser pego, o fantasma (jogador) ganhava um ponto. Se a bruxa pegasse o fantasma, o jogador perdia 10 pontos¹⁴. O objetivo foi cumprido com um pouco de dificuldade, pois eles precisaram da ajuda da instrutora em alguns momentos, principalmente para contagem dos pontos do jogo.

Nos momentos finais da aula, eles foram motivados a descrever o jogo que gostariam de desenvolver.

Encontro 7 ó Criando seu próprio jogo em Scratch

O objetivo desse encontro era que cada um criasse seu próprio jogo, praticando os conceitos adquiridos. A maioria dos alunos conseguiu desenvolver o jogo satisfatoriamente. Percebemos que eles entenderam as variáveis e os loops, que tinham raciocínio lógico para resolver os problemas que o jogo demandou. A liberdade dada para eles criarem seu próprio foi uma motivação para executarem o trabalho com capricho, para buscarem suas próprias soluções e aperfeiçoamentos. Alguns terminaram e melhoraram os jogos em casa. A Figura 2 mostra alguns jogos criados por eles, os quais estão disponíveis no site do Projeto em <http://jovemhacker.org/>.

Apenas um dos alunos não conseguiu terminar o jogo, mesmo com a ajuda dos monitores, e se mostrou frustrado com isso. O jogo que ele propôs era bem diferente dos exemplos vistos nos encontros anteriores, diferente das propostas dos colegas. A frustração não foi apenas dele, mas também de quem estava no apoio. Faltou alertarmos sobre a dificuldade em fazer aquele jogo com Scratch antes do início da atividade.

¹⁴ Créditos para o material do Projeto CodeClub disponível em <http://codeclubbrasil.org/material/>

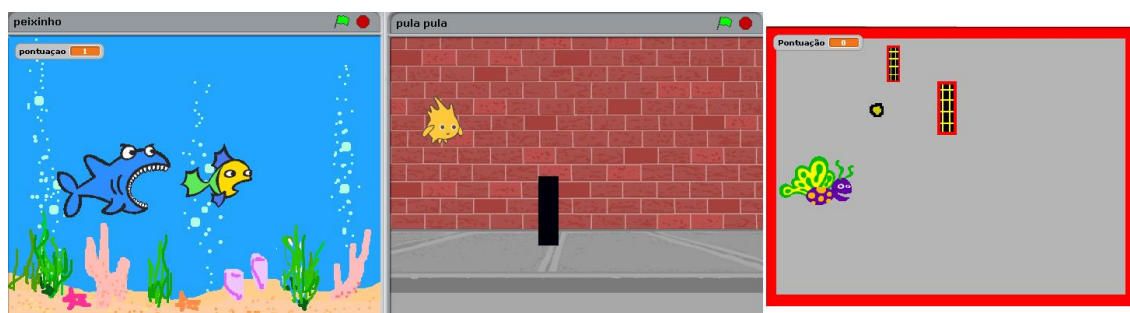


Figura 2 É Imagens dos jogos desenvolvidos pela turma piloto do Projeto.

Encontro 8 ó Avaliação

O objetivo do último encontro foi avaliar o trabalho realizado e registrar a opinião, as dificuldades e as preferências dos jovens. Nesse dia, apenas cinco deles estavam presentes. Explicamos que esta era a primeira turma do Projeto Jovem Hacker e que outras oficinas similares seriam feitas em outras ocasiões e, por isso, a opinião deles era muito importante para melhorarmos em edições futuras.

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 e nos retiramos da sala para que ficassem mais à vontade para conversarem sobre suas impressões e respostas.

Todos disseram ter gostado do contato com programação de computadores. Eles se mostraram dispostos a conhecer outras linguagens de programação, disseram que queriam conhecer *õde tudo um pouco*. A maioria se mostrou interessada em participar de outras oficinas do Projeto Jovem Hacker.

Sobre os encontros realizados, eles mencionaram a dinâmica feita ao ar livre, referido como *õo dia do Bisõ*, como sendo o dia que mais gostaram. Em contrapartida, o encontro que menos gostaram foi o da aula expositiva, onde explicamos e mostramos exemplos de algoritmos. Eles avaliaram esse encontro como *õchato*.

Sobre o tempo e a quantidade de encontros, eles acharam uma vez por semana pouco, além disso o tempo do encontro foi considerado curto ó apenas 1 hora e 30 minutos. *õQuando a gente começa a pegar o jeito a aula acaba*. Eles acharam interessante ter uma tarde ou um dia *õimerso* na programação, no estilo *hackathon*.

Disseram ainda ter gostado do dia que fizeram o joguinho porque foi uma oportunidade de colocarem em prática o que aprenderam, mas que o tempo foi muito curto. Apenas um deles disse não ter gostado do dia do joguinho por não ter conseguido criar um jogo com sua ideia, mesmo com a ajuda dos monitores. Então perguntamos: *õE se o joguinho tivesse dado certo?* e ele respondeu *õÁ teria sido legal!*

4. Lições aprendidas

Muitos dos nossos encontros foram elaboradas sob demanda, conforme as respostas da turma. Apresentamos, a seguir, as lições que aprendemos com esse piloto, que servirão de base para definirmos nosso modelo de formação:

- 1) Fugir da aula expositiva tradicional. De acordo com a avaliação que realizamos, a aula que eles menos gostaram foi a expositiva (encontro 3). Precisamos de uma maneira mais dinâmica, participativa e exploratória de apresentar a teoria para eles. É perceptível a mudança na motivação quando a aula é exploratória.
- 2) Promover aulas dinâmicas, com socialização. O encontro que os alunos mais gostaram foi *o dia do Bisö* (encontro 5). De fato, nesse dia conseguimos que eles assimilassem diversos conceitos, muitos deles avançados, de uma maneira lúdica. Além disso, o momento de *comer o Bisö* proporcionou socialização e descontração ao grupo, o que torna os encontros mais leves e agradáveis.
- 3) Usar exemplos minimalistas. Os alunos sentiram dificuldade no encontro 6 (jogo da bruxa e do fantasma) porque o exemplo era muito grande. Nesse caso, o ideal seria mostrar exemplos menores, que eles possam explorar e, passo a passo, construir um programa maior.
- 4) Ter encontros mais longos. Os alunos reclamaram do fato das aulas serem curtas, com pouco tempo para explorarem os programas.
- 5) Eles precisam se sentir capazes. A motivação aumenta quando eles se sentem capazes de fazer os programas. As lições que aprendemos, tais como ter encontros mais dinâmicos, deixa-los explorar mais as interfaces e os programas, usar exemplos minimalistas, pode contribuir com o sentimento de *ser capaz* de realizar uma tarefa e, conseqüentemente, melhorar a motivação da turma.

5. Desafios e limitações

A falta de Internet foi um limitador de nossas atividades. Não pudemos usar a última versão no Scratch, o que permite publicar os trabalhos e ver o que as outras pessoas estão fazendo. Além disso, um dos objetivos do Projeto Jovem Hacker é compartilhar conhecimento e construí-lo preferencialmente de uma maneira coletiva. A falta da Internet nos encontros presenciais dificulta a realização desse objetivo. Para nossas próximas turmas, vale a pena investigar projetos como o *Webmaker*, já citado anteriormente, com iniciativas para ensino em espaços com pouco ou nenhum acesso¹⁵.

Outra limitação que tivemos foi com relação ao espaço físico. A sala era apertada, faltou espaço para os instrutores circularem. Além disso, a sala não contava com lousa e pincel, itens importantes para explicar alguns conceitos de programação.

Durante a roda da conversa, os alunos disseram que acharam difícil criar a condição utilizando a variável e posicioná-la corretamente no *script*. 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 ó são apenas 12 horas para aprender lógica de programação. Explicamos que, para aprender uma linguagem de programação é preciso estudar, praticar, dedicar-se em casa. Então, um dos nossos desafios é motivá-los a explorar e estudar fora das aulas presenciais, em casa e outros ambientes.

¹⁵ <https://keyboardkat.makes.org/thimble/LTIxMDA3NTY0ODA=/lofi-nofi-teaching-kit>

A motivação da turma também precisa ser melhorada por meio de dinâmicas e aulas mais exploratórias. Começamos com nove alunos e, no dia do último encontro, apenas cinco deles compareceram. Podemos dizer que, realizar encontros mais motivadores, onde eles se sintam capazes, constitui-se um desafio deste projeto.

Por fim, um grande desafio desta pesquisa é a avaliação. Nesta primeira experiência, não quantificamos o quanto eles de fato aprenderam lógica de programação. Podemos dizer que a maioria conseguiu fazer o joguinho. Foi possível observar também que os alunos que faltaram duas aulas seguidas tiveram mais dificuldade para fazer o joguinho, precisaram de mais ajuda dos monitores. Entretanto, essas são observações empíricas, sem dados quantitativos ou qualitativos sobre o aprendizado dos alunos. Encontrar uma forma de mensurar e avaliar o aprendizado sem utilizar os métodos tradicionais, tais como provas, é um tópico desafiador e que merece investigação para nossas próximas oficinas.

6. Conclusões e trabalhos futuros

O Projeto Jovem Hacker começou a dar os primeiros passos de sua trajetória; ainda são muitos os desafios a serem explorados para que consigamos chegar mais perto dos objetivos traçados. Neste artigo, apresentamos um piloto com a linguagem Scratch que nos proporcionou aprender lições e conhecer um pouco dos desafios que temos pela frente. Esperamos ainda, que a experiência adquirida com o piloto venha a contribuir com a construção de um projeto de formação para jovens, no sentido de ter uma geração que seja mais autônoma tecnologicamente.

Agradecimentos

Agradecemos aos alunos que fizeram parte do nosso piloto com Scratch, bem como aos voluntários (monitores) que auxiliaram nos encontros, em especial ao Carlos Robson de Oliveira, Raniere Silva e Carolina Bonturi. Agradecemos também ao CIS-Guanabara, de Campinas, pelo espaço cedido para realizarmos o piloto.

Referências

- Amiel, T., & Amaral, S. F. do. (2013). Nativos e Imigrantes: Questionando a Fluência Tecnológica de Alunos e Professores. *Revista Brasileira de Informática Na Educação*, 21(3), 1611.
- Flannery, L., Kazakoff, E., Bonta, P., Silverman, B., Bers, M., & Resnick, M. (2013). Designing ScratchJr: Support for Early Childhood Learning Through Computer Programming. *Proceedings of the 2013 Interaction Design and Children (IDC) Conference*. New York.
- Oblinger, D. G., & Hawkins, B. L. (2006). The myth about student competency. *Educause Review*, (March/April), 12613.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York, NY: Basic Books.
- Périssé, P., & Garboggini, I. (2004, 2005). *Informática com crianças pequenas*. Pátio Educação Infantil, 33-36.
- Resnick, M., Maloney, J., Monroy-Hernandez, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, vol. 52, no. 11, pp. 60-67 (Nov. 2009).
- Venceslau, M. (2013) Falta de profissionais de TI se agravará no Brasil, diz IDC. *EXAME.com*. Disponível em <http://exame.abril.com.br/carreira/noticias/falta-de-profissionais-de-ti-se-agravara-no-brasil-diz-idc>. Acesso em 15/07/2014.