

Saberes D'Avó: Uma Abordagem para o Ensino de Programação no Ensino Médio

Laíza Ribeiro Silva, Dárlinton Barbosa Feres Carvalho

Universidade Federal de São João del-Rei
Departamento de Ciência da Computação
Av. Visconde do Rio Preto, s/nº, Colônia do Bengo, CEP 36301-360
São João del-Rei – MG – Brasil

laizaribeiroinfor@gmail.com, darlinton@acm.org

Abstract. *In pursuit of innovation in the introductory programming education for young people, we propose a new approach by adding an emotional bias to the technical one, which we call Saberes D'Avó (Grandma Wisdom). This approach integrates a gamification bias regarding the use and production of computer applications as study tools. In this sense, we propose a course in which students get in contact with grandparents to discuss regional knowledge and, through the training, create quiz applications games, to be use by children. This article presents the proposal approach and a consolidated of the conducted case study.*

Resumo. *Em busca por inovação no ensino de programação introdutória para jovens, propõe-se uma nova abordagem agregando um caráter mais emocional ao técnico, intitulada Saberes D'Avó. Nesta abordagem, integra-se um caráter lúdico de modo a utilizar e produzir aplicativos computacionais como ferramentas de estudo. Neste sentido, é proposto um minicurso onde alunos são colocados em contato com avós a fim de discutir sobre saberes regionais e, por meio de um treinamento, criam aplicativos do tipo jogo de pergunta e resposta (quiz) para serem utilizados por crianças. Neste artigo apresenta-se a abordagem proposta e um consolidado sobre um estudo de caso realizado.*

1. Introdução

Ensino de programação para jovens é um tema em discussão atualmente na sociedade. Vinton Cerf, Chief Internet Evangelist do Google, discute na revista *Communications of the ACM* de Março/2016 que a programação de computadores faz com que as pessoas tenham uma espécie de mentalidade disciplinada que se faz útil em diversas áreas, como engenharia, pesquisas científicas, empresas e política. Logo, ele enfatiza: "Mesmo se um aluno não ir para uma carreira em ciência da computação ou um assunto relacionado, essas habilidades são susceptíveis de ser útil em qualquer empreendimento em que o pensamento analítico é valioso". [Cerf 2016].

Barack Obama, presidente dos Estados Unidos da América (EUA), no discurso *State of the Union* de 2016, apresenta o programa Ciência da Computação para Todos. Este programa aborda o ensino de programação em escolas como grande meta de seu governo para este ano, no intuito de garantir sustentabilidade na empregabilidade de seu povo no futuro. Em entrevista disponível no projeto para popularização de programação Code.org¹, Obama destaca a importância do ensino de computação para os jovens.

¹<https://code.org/>

Ele ressalta o quão importante é para os jovens dominarem as ferramentas e tecnologias que mudarão o jeito com que eles farão todas as coisas e enfatiza a importância de serem produtores de tecnologia. Em suas palavras²: *"Não baixe aplicativos, mas ajude a desenvolvê-los. Para as pessoas que trabalham e estudam firme, é possível que moldem seu próprio futuro"*.

Entende-se que estas são visões inspiradoras e, além de buscar soluções pragmáticas, deve-se também pesquisar sobre abordagens inovadoras para realizá-las. Para isso, vale considerar aspectos sobre o consumo de tecnologia por jovens. Refletindo a este respeito, percebe-se como é notável o uso de tecnologia para entretenimento. Portanto, aproveitando-se desta prática, é importante desafiar a transformar os jovens de modo que não sejam apenas consumidores de tecnologia, mas também produtores, e consigam usufruir de toda potencialidade e possibilidades que as novas tecnologias oferecem.

O ensino de programação por meio de jogos digitais é um tema bem explorado, com notável destaque para o bem sucedido Agent Sheets [Repenning et al. 2015]. De acordo com os criadores³, esta é uma ferramenta revolucionária que permite as pessoas criarem seus próprios jogos ou simulações e publicá-los na web através de uma interface amigável de arrastar e soltar. A missão desta ferramenta é levar a participação dos alunos do ensino primário e secundário para o estudo de ciência, fazendo com que desenvolvam tecnologias através de aplicativos motivacionais. No entanto, nossa pesquisa trabalha com uma proposta mais no sentido da ludificação de saberes tradicionais, ou seja, de outras disciplinas do currículo escolar ou mesmo da cultura regional. Busca-se capacitar os jovens ainda na escola, com um caráter interdisciplinar, e de modo a prepará-los a lidar melhor com novas tecnologias e deixá-los cientes das possibilidades existentes para quem conhece um pouco mais sobre seu funcionamento.

Além disso, ainda é considerado nesta proposta o desafio de desenvolver uma abordagem inovadora no sentido de aliar um caráter mais emocional ao técnico. Além de promover a interdisciplinaridade no estudo técnico, busca-se também estimular o estreitamento de laços entre gerações. É analisado a condução do ensino de programação introdutória por meio de um processo baseado na interação entre gerações, tanto em relação aos mais velhos (avós) como crianças (alunos do ensino fundamental).

Neste artigo, são apresentados os fundamentos considerados na criação e operacionalização da abordagem proposta, chamada de Saberes D'Avó. Vale ressaltar que aliada à fundamentação teórica que consistiu do estudo de projetos similares e artigos encontrados na literatura, a proposta da abordagem também contempla um viés pragmático consonante à realidade das escolas de nossa região. Também são apresentados resultados empíricos considerando um estudo de caso onde foi utilizada a abordagem proposta.

Este artigo está organizado como apresentado a seguir. A Seção 2 apresenta os fundamentos teóricos considerados no desenvolvimento deste estudo, bem como consolidando alguns trabalhos relacionados. Já a Seção 3 mostra em detalhes a abordagem Saberes D'Avó. Na Seção 4, apresenta-se um consolidado do estudo de caso realizado em relação a aplicação da abordagem proposta em uma escola da região e os resultados obtidos. Por fim, as considerações finais são discutidas na Seção 5.

²Retirado de vídeo disponível no site <https://code.org/>

³<http://www.agentsheets.com/>

2. Fundamentação Teórica

Primeiramente, vale advertir que há uma pluralidade de trabalhos sobre ensino de programação introdutória, com intenções e aceitabilidade de certa forma até divergentes. Por exemplo, o ensino de programação no ensino médio no sentido de formação técnica de programadores não é amplamente recomendado, ou mesmo aceito, por especialistas. Embora, o ensino de programação como meio de estimular o pensamento crítico computacional seja mais aceito como salutar a todos [Cerf 2016].

Nesta pesquisa, é compartilhada a visão de que aprender programação na escola é necessário e pesquisamos por um viés positivista neste sentido. É compartilhada uma linha de pesquisa como a explorada em [Scaico et al. 2013], onde aprender a programar é extremamente importante e deveria ser ensinado não só para estudantes da computação, sendo que quem aprende a programar em idade escolar tende a desenvolver habilidades significativas no que se diz respeito a resolução de problemas e a fluência ao lidar com o uso de tecnologias.

Nas escolas, acostumados com metodologias tradicionais baseadas em quadro e giz, é um desafio para os docentes do ensino médio compreenderem e utilizarem novas ferramentas tecnológicas. Porém, após o aprendizado sobre novas práticas tecnológicas, podem aperfeiçoar sua prática de ensino e utilizá-las para introduzir uma nova abordagem na transmissão do conhecimento [Cox 2003]. Os docentes verão que além de usar tecnologia, é possível criá-la, trazendo uma infinidade de possibilidades de inserção do conhecimento como um todo para os alunos por meio da computação. Segundo [Thomas and Brown 2011], ao se deslocarem dos modelos pedagógicos tradicionais, os docentes reconhecem o saber e o fazer como ações importantes para a aprendizagem, mas eles introduzem o brincar como uma nova dimensão relevante para a educação. É necessário apresentar motivação aos jovens que estão iniciando na atividade de programação, para que isso faça com que tenham mais interesse e vontade de adquirir conhecimento. Os adolescentes adoram esse tipo de entretenimento, e ao dar uma base de como desenvolver seu próprio jogo, tornará o aprendizado muito mais interessante.

Segundo [Kelleher and Pausch 2005], a questão da sintaxe é a maior dificuldade encontrada pelos iniciantes em programação, pois nem todas as linguagens são claras. Simplificar a sintaxe, fazendo com que fiquem mais próximas da linguagem natural, ajudaria muito no aprendizado mais rápido da programação. Neste sentido, existem diversas iniciativas que consolidam ambientes de criação de aplicativos com intuito educativo, com destaque para as linguagens baseadas em blocos. A utilização destes ambientes são tema de pesquisas interessantes, como a realizada por [da Cruz and Lima 2014] em relação aos ambientes Scratch⁴ e Alice⁵, e o Agent Sheets [Repenning et al. 2015].

Uma tendência tecnológica mais moderna é a utilização de *smartphones*. Entende-se que um ambiente para criação de aplicativos para estes dispositivos é mais apropriado e interessante para alunos da geração atual. Neste sentido, é explorado o uso do ambiente para desenvolvimento MIT App Inventor⁶, que foi criado no intuito de facilitar o aprendizado de programação e criação de aplicativos Android.

⁴<https://scratch.mit.edu/>

⁵<http://www.alice.org/>

⁶<http://appinventor.mit.edu/explore/about-us.html>

No ano de 2014, o Instituto Federal do Paraná - Câmpus Irati adotou o ambiente de desenvolvimento MIT App Inventor para elaboração de aplicativos em um projeto de extensão que tinha como intuito desenvolver o raciocínio dos alunos participantes, além de ensiná-los sobre esse ambiente [Duda et al. 2015]. Entretanto, essa iniciativa explorou apenas o desenvolvimento do uso de linguagem matemática para a modelagem e escrita de soluções de problemas. Nossa pesquisa busca explorar uma abordagem com mais interdisciplinaridade, considerando também ciências, história, etc.

Vale ressaltar que um dos principais motivos da resistência à introdução de mais tecnologias na didática além da falta de conhecimento em tecnologia dos docentes, é a precariedade na infraestrutura tecnológica disponível. Ao visitar escolas da região, percebe-se que elas não dispõem de internet de boa qualidade e de computadores em bom funcionamento. A internet é interrompida muitas vezes durante o dia e poucos computadores realmente funcionam. Contudo, cada vez mais os alunos já possuem *smartphones* e os recursos disponíveis podem ser aproveitados sim, de certa forma, para viabilizar um minicurso, como o que é proposto neste trabalho.

Por fim, é preciso definir um instrumento de avaliação da aprendizagem. Por se tratar de um conteúdo mais relacionado a conceitos abstratos, se torna ainda mais difícil definir uma avaliação. Esta pesquisa fundamenta-se na taxonomia de Bloom ([Bloom 1956], [Ferraz and Belhot 2010]), que foi proposta para ser utilizada na definição e escolha dos instrumentos de avaliação, assim como para definir os objetivos instrucionais. Esta taxonomia foi criada para auxiliar na concepção de uma avaliação mais adequada, principalmente quando se trata de disciplinas abstratas, como programação introdutória. A taxonomia de Bloom possui entendimentos diversos, e até mesmo divergentes. Além disso, segundo [Fuller et al. 2007], apesar de ser a taxonomia mais utilizada, existem categorias que não são fáceis de serem interpretadas. Por esse motivo, [Anderson et al. 2001] criaram a taxonomia de Bloom revisada, utilizada nesta pesquisa.

Na taxonomia de Bloom revisada existe um verbo associado para cada categoria, sendo que o nível de complexidade das questões de cada categoria obedecem uma ordem crescente de dificuldade. Estes verbos são responsáveis por classificar o nível de aprendizado do aluno. Logo, para que o professor consiga desenvolver uma avaliação precisa e eficiente, é necessário que o mesmo saiba interpretar corretamente os verbos desta taxonomia. A Tabela 1 mostra as categorias da taxonomia de Bloom revisada, seguida por suas interpretações. Vale lembrar que para cada categoria existem diversas interpretações, porém foram consideradas aquelas que se enquadram melhor na abordagem apresentada neste artigo.

Nesta pesquisa, utiliza-se a taxonomia de Bloom revisada na elaboração de um instrumento de avaliação. São propostas questões a fim de verificar o aprendizado por área, buscando evidenciar áreas e conteúdos deficientes, bem como avaliar se o minicurso cumpriu com seus objetivos de modo esperado. No entanto, a elaboração de instrumentos de avaliação é por si só um objeto de pesquisa complexo, além do escopo do que é proposto e apresentado neste artigo. Portanto, vale ressaltar, que neste artigo esta questão não é explorada em profundidade e apresenta-se apenas um instrumento de avaliação considerando questões com o intuito de avaliar o estudo de caso. É importante advertir que o instrumento de avaliação elaborado provavelmente não é generalizável. Logo, mais pesquisas neste sentido são necessárias, como discutido mais a seguir.

Tabela 1. Taxonomia de Bloom revisada e suas interpretações

Categoria	Interpretação da categoria aplicada em programação
Lembrar	Reconhecer a implementação de um determinado conceito.
Entender	Completar partes faltantes de um programa utilizando fragmentos de código.
Aplicar	Resolver um problema familiar, mas com dados ou ferramentas não familiares.
Analisar	Identificar componentes críticos para o desenvolvimento e identificar componentes ou requisitos não importantes.
Avaliar	Encontrar um erro de lógica em um trecho de código dado.
Criar	A partir de blocos de códigos distribuídos aleatoriamente, ordená-los para resolver o problema em questão.

3. Saberes D'Avó

A abordagem proposta pelo Saberes D'Avó é desenvolvida como um minicurso de programação introdutória com objetivo de estimular e treinar os jovens, mais especificamente alunos do ensino médio, a utilizar e criar aplicativos Android. Por meio de um minicurso presencial sob orientação de um instrutor, os alunos desenvolvem aplicativos utilizando a ferramenta App Inventor. São exploradas questões de disciplinas de matemática, ciência, história e cultura regional para ilustrar exemplos utilizados no ensino dos conceitos básicos de programação. O trabalho final proposto é a criação de um jogo considerando conteúdo sobre temas regionais a serem obtidos a partir de entrevistas com idosos, especificamente, seus avós. Este jogo deve ser um aplicativo do tipo perguntas e respostas (quiz) para ser jogado por outros alunos do ensino fundamental.

De modo a apresentar um caso real de utilização da abordagem proposta e verificar na prática a sua execução, foi realizado um estudo de caso considerando um cenário típico de uma escola com um laboratório de informática com acesso à internet e utilização de *smartphones* dos próprios alunos. Na conclusão do minicurso, foi pedido aos alunos que respondessem um questionário de avaliação do minicurso e uma prova teórica sobre o conteúdo apresentado no minicurso. O questionário continha algumas perguntas sobre o que conheciam a respeito de programação antes de iniciar o minicurso e como foi a experiência após a conclusão dele, juntamente com o aplicativo desenvolvido. Na próxima seção são apresentados mais detalhes destes questionários, bem como os resultados.

Para avaliar a aprendizagem de modo objetivo, foi utilizada uma prova teórica, elaborada de acordo com a taxonomia de Bloom revisada, onde o objetivo é mensurar o nível de aprendizado do aluno de acordo com cada categoria dessa taxonomia. A prova continha 24 questões, sendo 4 questões para cada nível de dificuldade diferente, de acordo com as categorias da taxonomia de Bloom revisada.

4. Estudo de Caso

O Saberes D'Avó teve sua primeira realização na Escola Estadual "Doutor Viviano Caldas", localizada em Prados, Minas Gerais, Brasil. O minicurso presencial foi dimensionado para atender 16 alunos do ensino médio, contando com 5 aulas (15horas-aula), que foram realizadas às segundas-feiras em período vespertino, visto que as aulas regulares

dos alunos ocorrem em período matutino. Os alunos interessados em participar fizeram uma inscrição no minicurso e a direção da escola selecionou aqueles com mais mérito, de acordo com o conhecimento declarado em relação a informática, inglês e matemática. Foram selecionados 9 alunos do 1º ano, 6 alunos do 2º ano e 1 aluno do 3º ano, sendo 7 do sexo feminino e 9 do sexo masculino.

Ao longo do minicurso houveram 2 desistências, ambas do sexo masculino, sendo um do 1º ano e outro do 3º ano. Um dos alunos desistentes chegou a responder o questionário final de avaliação, entretanto não realizou nenhuma prova. O outro aluno desistente não respondeu ao questionário nem as provas, justificando que não gosta de aprender assuntos relacionados à tecnologia em um local que exige disciplina. A outra justificativa foi devido ao fato do aluno não estar conseguindo se ausentar do local de trabalho para participar das aulas.

A escola possui uma infraestrutura tecnológica problemática. Como a internet não funcionava em todos os computadores, optou-se por uma dinâmica de estudo em grupos de 4 alunos, possibilitando assim utilizar os recursos disponíveis sem desapontamentos. Os alunos relataram a dificuldade em acessar a internet, que mesmo sendo utilizada em apenas 4 computadores, caiu algumas vezes, embora sem inviabilizar a realização das atividades propostas. Estas limitações acabaram por favorecer aspectos positivos para o aprendizado não considerados anteriormente. Os alunos gostaram da interação com os colegas de grupo no desenvolvimento dos aplicativos, já que se tratava de uma tarefa mais descontraída do que as atividades convencionais de outras disciplinas. Após a desistência dos dois alunos, dois grupos ficaram com apenas três integrantes, o que também não afetou no desenvolvimento das atividades propostas em aula.

Ao final do minicurso, os alunos responderam um questionário que continha perguntas relacionadas a experiência de participação no minicurso. Eles relataram que antes de iniciar o minicurso acreditavam que seria uma tarefa difícil desenvolver um pequeno aplicativo, porém, com o decorrer das tarefas, perceberam que era mais fácil do que imaginavam. Todos os alunos relataram que gostariam que houvesse uma disciplina de programação durante o ensino médio e acreditam que estudar programação melhora a criatividade e o interesse dos mesmos em relação a outras disciplinas.

Os alunos também ressaltaram que antes de conhecer sobre programação, acreditavam que desempenhar as atividades propostas seria muito desafiador, tido como impossível para alguns. Porém, ao começarem a ter os primeiros contatos, perceberam que não era algo tão difícil quanto parecia. Para eles, aprender programação é importante e desejam aplicar o que aprenderam em outras disciplinas.

Ainda de acordo com o questionário, os alunos, em sua maioria, declararam ter conhecimento básico ou nenhum conhecimento da língua inglesa (73,3%), não sabiam programar (86,7%) e também não conheciam o MIT App Inventor (93,4%). O conhecimento da língua inglesa considerando um nível intermediário ou avançado, declarado por 4 alunos (26,7%), ajudaria em uma maior compreensão das ferramentas do MIT App Inventor já que o mesmo não é disponível no idioma português, porém não foi um fator limitante no aprendizado.

Em relação aos resultados do instrumento de avaliação de aprendizagem, é considerado um total de 14 alunos e 4 questões referentes a cada categoria, sendo 56 o valor

máximo de acertos possível. Analisando o Gráfico 1, verifica-se um fenômeno inusitado. Pode-se observar que as questões que obtiveram um maior número de acertos foram as questões referentes a categoria Entender e Avaliar. Entretanto, algumas questões de categorias mais complexas obtiveram um maior número de acertos do que questões menos complexas.

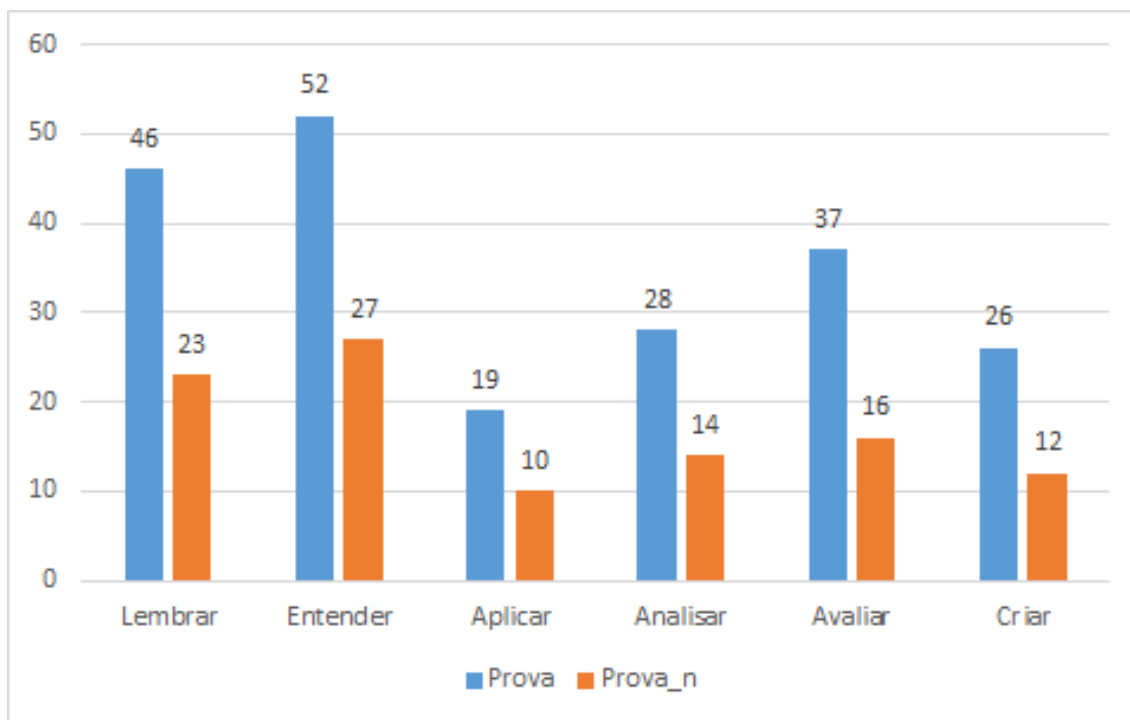


Figura 1. Categorias da taxonomia revisada de Bloom x Número de acertos

Com o intuito de remover questões que pudessem gerar interpretações equivocadas por parte dos alunos, definiu-se a exclusão das questões de maior e menor quantidade de acertos (Prova_n). Mesmo assim, o resultado é similar ao obtido anteriormente (Prova).

A questão referente a categoria Entender teve o maior número de acertos (52), enquanto a questão referente a categoria Lembrar obteve 46 acertos. A categoria com o menor número de acertos foi a Aplicar, que contou com 19 acertos, sendo 33,9% do total possível de acertos. A categoria Criar foi a segunda categoria com o menor número de acertos, tendo 26 acertos contabilizados. Como esta última se trata da categoria mais complexa da taxonomia, o baixo número de acertos não é um fator preocupante, principalmente pelo baixo número de aulas oferecidas.

Para verificar o número de acertos por aluno, a Figura 2 mostra um gráfico para que seja possível melhor observar os dados. O gráfico é ordenado de forma crescente, de acordo com a soma das notas da primeira e segunda prova. O máximo de acertos permitido por aluno é 24. A menor porcentagem de acertos é referente aos alunos A11 e A14, tendo acertado 8 questões (33,4%). A maior porcentagem de acertos é também de dois alunos (A2 e A7), em que erraram uma questão cada um, totalizando em uma porcentagem de 95,8% de acertos. Metade dos alunos conseguiram aproveitamento maior ou igual a 66,7% (A3, A5, A13, A8, A4, A2 e A7) e a outra metade tiveram um aproveitamento entre 33,4% e 58,4%.

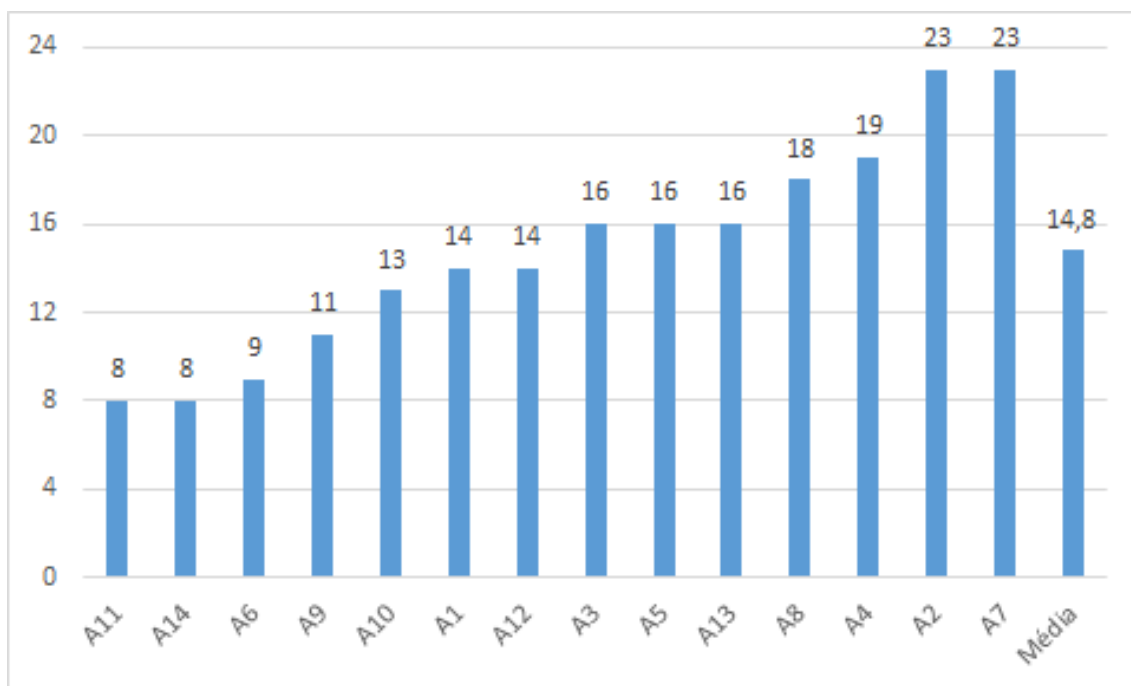


Figura 2. Número de acertos x Aluno

5. Considerações Finais

Sabendo da dificuldade em assimilar as abstrações de programação [Júnior et al. 2005], e também levando em consideração que foram apenas 5 aulas ministradas, os resultados obtidos nas provas nos mostram que os alunos aproveitaram muito bem o minicurso. Eles conseguiram um aproveitamento satisfatório considerando o conjunto de todas as categorias da taxonomia de Bloom revisada, apesar da categoria Aplicar, onde identificamos uma observação abaixo da expectativa. Sobre a categoria Criar, apesar de ter contado também com um baixo número de acertos, entendemos ser compreensível devido ao nível de sua dificuldade, sendo a categoria mais complexa desta taxonomia. Para explicar essa discrepância de acertos na categoria Aplicar, estamos inclinados a crer que esteja relacionado ao não entendimento das questões, mas uma discussão mais aprofundada a este respeito requer mais pesquisa para melhor embasamento e discussão.

Em relação aos resultados qualitativos, as manifestações dos alunos enaltecem o sucesso da abordagem proposta. O caráter lúdico proporcionou um ambiente confortável para tratar de assuntos complexos. O mito de que programação de computadores é impossível para os alunos, que acaba gerando aversão em relação ao uso avançado de tecnologia, foi facilmente superado logo nas primeiras aulas, o que entendemos validar o ambiente de ensino utilizado, baseado no App Inventor. As declarações em relação a dinâmica do trabalho em grupo trazem mais evidências sobre a importância de se incorporar mais este caráter humanístico no ensino de programação, sendo que este em especial não havia sido considerado originalmente e que fortuitamente se revelou um importante aspecto a ser explorado.

Os quatro jogos desenvolvidos foram utilizados em uma gincana organizada com alunos do ensino fundamental. Esta iniciativa foi viabilizada com o prêmio de Agosto

de 2015 da The Awesome Foundation⁷, uma comunidade global dedicada a disseminar projetos irados pelo universo. Esta gincana foi considerada inicialmente como forma de avaliação dos aplicativos, buscando mais motivação para os alunos desenvolverem bons jogos. De forma geral, todos os aplicativos foram similarmente muito bem avaliados. Nesta participação os alunos tiveram apoio dos pais e avós para responder as questões. Ao final, todos foram premiados e elegeu-se a comunidade unida como a campeã desta gincana. Entendemos também que a gincana não faz parte deste estudo, por isso não aprofundamos nesta discussão.

A partir desta experiência bem-sucedida, planejamos a realização do minicurso em outras escolas da região. Considera-se aplicar o mesmo instrumento de avaliação, mas ressalta-se a necessidade de pesquisa mais aprofundada para compreender o que se passa em relação a categoria Aplicar e também produzir um instrumento mais qualificado para o seu propósito. Vale ainda destacar que se pode utilizar o aprendizado de programação para resolver problemas de outras disciplinas, tornando a programação como parte de uma abordagem interdisciplinar em disciplinas tradicionais. Neste sentido, estamos trabalhando também na virtualização da proposta, ou seja, a consolidação de um minicurso online por meio da criação de material didático explorando justamente esta interdisciplinaridade.

Agradecimentos

À UFSJ e FAPEMIG pelo apoio no desenvolvimento e apresentação desta pesquisa.

Referências

- Anderson, L. W., Krathwohl, D. R., and Bloom, B. S. (2001). *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Allyn & Bacon.
- Bloom, B. S. (1956). *Taxonomy of Educational Objectives: The Classification of Education Goals. Cognitive Domain. Handbook 1*. Longman.
- Cerf, V. G. (2016). Computer Science in the Curriculum. *Communications of the ACM*, 59(3):7–7.
- Cox, K. K. (2003). *Informática na educação escolar*. Autores Associados.
- da Cruz, A. K. B. S. and Lima, L. C. M. (2014). Estudo e Testes de Usabilidade em Sistemas de Autoria de Software: Scratch e Alice. *Blucher Design Proceedings*, 1(4):3673–3685.
- Duda, R., Zontini, D. D., da Silva, S. d. C. R., and Grossi, L. (2015). Elaboração de Aplicativos para Android com Uso do App Inventor: Uma Experiência no Instituto Federal do Paraná–Câmpus Irati. *Revista Brasileira de Ensino de Ciência e Tecnologia*, 8.
- Ferraz, A. and Belhot, R. V. (2010). Taxonomia de Bloom: revisão teórica e apresentação das adequações do instrumento para definição de objetivos instrucionais. *Gest. Prod., São Carlos*, 17(2):421–431.

⁷<http://www.awesomefoundation.org/>

- Fuller, U., Johnson, C. G., Ahoniemi, T., Cukierman, D., Hernán-Losada, I., Jackova, J., Lahtinen, E., Lewis, T. L., Thompson, D. M., Riedesel, C., et al. (2007). Developing a computer science-specific learning taxonomy. *ACM SIGCSE Bulletin*, 39(4):152–170.
- Júnior, J., Rapkiewicz, C. E., Delgado, C., and Xexeo, J. A. M. (2005). Ensino de algoritmos e programação: uma experiência no nível médio. In *XIII Workshop de Educação em Computação (WEI'2005)*. São Leopoldo, RS, Brasil.
- Kelleher, C. and Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys (CSUR)*, 37(2):83–137.
- Repenning, A., C., D., Koh, K. H., Nickerson, H., Miller, S. B., and Brand, C. (2015). Scalable game design: A strategy to bring systemic computer science education to schools through game design and simulation creation. *Transactions on Computing Education (TOCE)*, 15:1–31.
- Scaico, P. D., de Lima, A. A., Azevedo, S., da Silva, J. B. B., Raposo, E. H., Alencar, Y., Mendes, J. P., Scaico, A., et al. (2013). Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch. *Revista Brasileira de Informática na Educação*, 21(02):92.
- Thomas, D. and Brown, J. S. (2011). *A new culture of learning: Cultivating the imagination for a world of constant change*, volume 219. CreateSpace Lexington, KY.