

Ensino de Programação para crianças com o uso da ferramenta Scratch

Teaching Programming to Children Using the Scratch Tool

Autores:

Franciely Alves de Souza. Mestranda em Informática Aplicada na Universidade Federal Rural de Pernambuco. Recife, PE, Brasil.

E-mail: francielyalves835@gmail.com

Autora responsável pela correspondência

Ivaldir de Farias Junior. Doutor em Ciência da Computação pela Universidade de Pernambuco Campus Garanhuns. Recife, PE, Brasil.

E-mail: ivaldir.farias@upe.br

Recebido em: 07/03/2022 **Aprovado em:** 22/05/2023

DOI: 10.12957/interag.202365822

Relato

Resumo

Contexto: Este trabalho relata uma experiência vivenciada em um curso de férias com alunos entre 9 e 12 anos, que durante 1 mês e 1 semana de aula, tiveram a oportunidade de aprender conceitos básicos de programação utilizando a ferramenta Scratch. **Metodologia:** Nas aulas foram realizadas atividades práticas com o uso dos blocos de programação na ferramenta Scratch, que possibilitaram o embasamento para o desenvolvimento de dois jogos lúdicos, que foram avaliados de forma automática através da ferramenta Dr. Scratch. **Resultados esperados:** O curso foi concluído com 11 alunos, e entre os resultados destaca-se a evolução do aprendizado no decorrer das atividades realizadas, já que os alunos não possuíam nenhum conhecimento prévio. Além disso, os resultados demonstram o quanto a avaliação automática é eficiente para visualizar a evolução do aprendizado dos alunos, e consequentemente adaptar as aulas para suprir alguma dificuldade

Abstract

Context: This work reports on an experience lived in a vacation course with students between 9 and 12 years old, who during 1 month and 1 week of class, had the opportunity to learn basic programming concepts using the Scratch tool. **Methodology:** In the classes, practical activities were carried out with the use of programming blocks in the scratch tool, which allowed the basis for the development of two playful games, which were automatically evaluated through the Dr. Scratch tool. **Expected results:** The course was completed with 11 students, and among the results, the herence of learning during the activities carried out stands out, since he had no prior knowledge. In addition, the results demonstrate how herence automatic assessment is to visualize such herence of student learning, and consequently adapt classes to overcome any difficulty that is visualized from the evaluated criteria. **Conclusion:** We concluded that the course provided a herence learning moment, here

que seja percebida a partir dos critérios avaliados. Conclusão: Concluímos que o curso proporcionou um momento de aprendizado diferente, lúdico e dinâmico, onde os alunos tiveram a oportunidade de conhecer novos conceitos e apresentaram resultados significativos ao que foi proposto inicialmente no projeto.

students had the opportunity to learn about new concepts and presented significant results to what was proposed.

Palavras- chave: Programação em blocos; scratch; crianças; Dr. Scratch

Keywords: Block programming; Scratch; kids; Dr. Scratch

Área Temática: Tecnologias na Educação

Linha Temática: Programação para Crianças

Introdução

Os avanços da tecnologia impulsionaram a busca por profissionais de computação, por isso é importante ter uma grande quantidade de estudantes nessa área. As descobertas da computação são um dos fatores que aceleram as constantes mudanças na sociedade contemporânea, de modo que não é mais possível imaginar uma sociedade sem computadores e suas tecnologias¹.

É fundamental que os estudantes possam ter a oportunidade de adquirir conhecimentos básicos de computação ainda na infância, pois conseqüentemente, para se tornar um bom profissional, são necessárias habilidades específicas tais como: raciocínio lógico, interpretação de problemas, além de conhecimento técnico das linguagens de programação². Conseguir se deparar com tais conceitos ainda no período do ensino fundamental, possibilita que os estudantes possam analisar um possível ingresso na área, já que possuem o conhecimento do que se trata. Desta forma evitando um grande número de desistências, devido ao elevado nível de exigências presentes em cursos que ensinam programação³.

O ensino de computação favorece o desenvolvimento de habilidades que são úteis em qualquer área de atuação profissional, e uma delas é a aplicação de técnicas para resolução de problemas que combinam o pensamento crítico com os fundamentos da computação, denominada por Wing⁴ de “Pensamento Computacional” (PC).

O PC é o processo cognitivo usado pelos seres humanos para encontrar algoritmos para resolver problemas⁵. E envolve a resolução de problemas através dos fundamentos da computação⁴. Nesta perspectiva, esse relato apresenta resultados de um curso de férias onde foi proposto o desenvolvimento de algoritmos para a construção de jogos na ferramenta *Scratch*, levando em consideração que um ambiente de programação em blocos é fundamental para introduzir noções de linguagem de programação para usuários iniciantes como crianças.

Pois de acordo com Zanatta⁶, a programação não trata apenas de aprender a programar computadores, ela muda e amplia a visão de mundo, passando a entender de uma forma mais

clara os processos que envolvem o dia a dia. E segundo Bopprê⁷, o ensino de programação é de extrema importância para que as crianças desenvolvam e estimulem cada vez mais sua criatividade e a capacidade de resolver problemas. E para isso existem diferentes ferramentas de programação em blocos que proporcionam o início do aprendizado, assim como o *Scratch*.

O *Scratch* é um ambiente de programação visual que possibilita aos usuários desenvolver várias competências como: raciocínio lógico, criatividade e resolução de problemas, além de estimular novas habilidades e ideias tecnológicas através de seus ambientes criativos⁸. Foi desenvolvido no *Media Lab do Instituto de Tecnologia de Massachusetts*, e esse ambiente introdutório de ensino tem como objetivo facilitar a introdução de conceitos de computação⁹. E desde 2013, o *Scratch* está disponível de forma online, onde pode ser acessado de qualquer lugar utilizando apenas um computador ou smartphone com acesso à internet. Todas as atividades realizadas são salvas de forma automática na conta dos usuários.

Para avaliar códigos desenvolvidos no *Scratch*, já existe uma ferramenta livre de código aberto, o *Dr. Scratch*, que possibilita realizar avaliações de forma automática, ajudando a identificar pontos fracos como códigos mortos e variáveis não utilizadas; e também pontos fortes como a utilização dos sete pilares relacionados ao pensamento computacional¹⁰.

Ao avaliar um projeto, é informado quais dos conceitos estão presentes na programação e a pontuação alcançada, numa escala de 0 a 3. Somando o total das pontuações em cada conceito, é obtida a nota final do projeto, sendo 21 a pontuação máxima. Projetos com notas de até 7 pontos são considerados simples, entre 8 e 14 pontos são considerados como satisfatórios, e acima de 15 pontos são considerados como ótimos¹⁰.

Relato de Experiência

O curso de férias foi planejado para ocorrer remotamente através da plataforma *zoom*, tendo como público-alvo crianças com idades entre 9 e 12 anos, que para participar do curso, foram convidadas a realizar uma inscrição através de um formulário *online*, criado no *Google Drive*, onde informações dos pais ou responsáveis precisavam ser informadas. No ato da inscrição, houveram 18 alunos inscritos, porém por se tratar de um curso *online*, que tinha como requisito o uso do computador e internet, alguns alunos não conseguiram iniciar, e pelo mesmo motivo outros acabaram desistindo no percurso, sendo que apenas 15 iniciaram o curso, e ao finalizar apenas 11 alunos concluíram.

O curso foi desenvolvido com o intuito de estimular a criatividade, a autonomia e o desenvolvimento do raciocínio lógico, através do aprendizado de conceitos básicos de programação como o uso de: Variáveis; Operadores relacionais; Operadores lógicos e operadores aritméticos. Todas as aulas eram gravadas e disponibilizadas por meio da plataforma *Google Classroom*. Esse ambiente permite que todo o material e conteúdo possam ser compartilhados e agendados com a data, horário de liberação das atividades e fechamento de forma automática. Dessa forma, também foram disponibilizados na plataforma, atividades, materiais de apoio e *links* de vídeos que complementavam os conteúdos.

O curso ocorreu durante um mês e uma semana, no período de férias dos alunos, tendo início em dezembro de 2021 e finalizado no início de fevereiro de 2022. Neste período ocorreram 10 aulas, sendo duas aulas por semana que ocorriam na terça-feira e sexta-feira com duração de 1 hora e 30 minutos. A cada aula finalizada, uma atividade era disponibilizada

para os alunos fixarem o conhecimento. Essa atividade fazia parte da avaliação dos alunos que precisavam enviar a mesma feita um dia antes da próxima aula.

Inicialmente, as aulas possuíam o suporte de um conjunto de slides para a apresentação do conteúdo teórico antes da aula prática, onde a ferramenta *Scratch* era utilizada, e essa dinâmica ocorreu nas duas primeiras aulas, porém por ter se tornado perceptível a falta de atenção dos alunos em relação aos slides, fizemos uma reflexão sobre as percepções da sala de aula online e em seguida pensamos em soluções para tornar as aulas mais atrativas.

Após essa reflexão, todas as aulas foram mediadas apenas com a apresentação da tela da ferramenta *Scratch*, onde a teoria passou a ser apresentada juntamente com a prática, e os alunos já podiam observar o funcionamento das funções. Sempre que alguma função por algum erro passava a ser executada de forma incorreta, os próprios alunos analisavam o código um do outro, na intenção de encontrar o erro de forma colaborativa e como consequência, ajudar o colega. Essa tática de análise dos códigos no momento da aula, contribuiu positivamente para os alunos tornarem-se mais participativos e confiantes, principalmente quando surgiam dúvidas ou erros na execução da programação.

Após os ajustes realizados na dinâmica das aulas, os alunos se mostraram mais empenhados em realizar as atividades, desta forma concluíram o curso com 12 atividades realizadas e 2 jogos desenvolvidos.

As atividades eram voltadas para a fixação do conteúdo mostrado nas aulas. Os alunos precisavam responder às atividades após a aula e compartilhar o link para a visualização da professora. Sempre que uma atividade era realizada, a mesma poderia ser compartilhada através de um *link* que a própria ferramenta disponibiliza dando acesso a qualquer pessoa. Com isso, as atividades poderiam ser acessadas e visualizadas, visando apreciar a produção desenvolvida pelo aluno(a) e qual programação foi realizada. Diante deste contexto, essa funcionalidade foi considerada muito importante na ferramenta *Scratch online*, pois através desses links de compartilhamento, todas as programações também poderiam ser avaliadas de forma automática por meio da ferramenta *Dr. Scratch*.

Em uma das atividades realizadas e compartilhadas, os alunos precisavam utilizar estrutura de repetição para programar um personagem mergulhando no fundo do mar. Observe na Figura 1 que representa o desenvolvimento de uma das alunas do curso. Com essa programação, os aluno(a)s obtiveram o conhecimento de movimentar repetidamente um personagem, sem que ele pudesse sair do cenário.

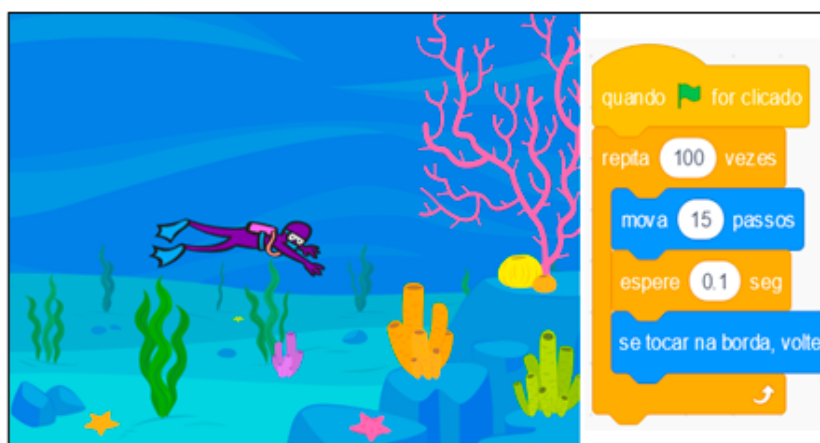


Figura 1 - Atividade realizada por uma aluna.

Fonte: Ferramenta Scratch.

Em seguida, foram adicionados novos elementos no cenário da Figura 3 para que as variáveis pudessem ser utilizadas, com isso, podemos observar o cenário com os novos elementos na Figura 2, onde o mergulhador precisa coletar diamantes, e a quantidade de diamantes coletados seriam armazenados na variável, desta forma fazendo uso dos comandos de condição, se o personagem tocar em um diamante, um valor será adicionado na variável.

Levando em consideração que várias atividades já haviam sido desenvolvidas e diferentes conceitos já trabalhados, foi proposto o desenvolvimento de um jogo de perguntas e respostas, onde os alunos programaram 10 perguntas, e o jogador poderia responder se a pergunta era verdadeira ou falsa, tantos os pontos de acertos como os de erros ficavam presentes no cenário, para que o usuário pudesse verificar sua pontuação. Observe na Figura 3 as telas do jogo proposto.

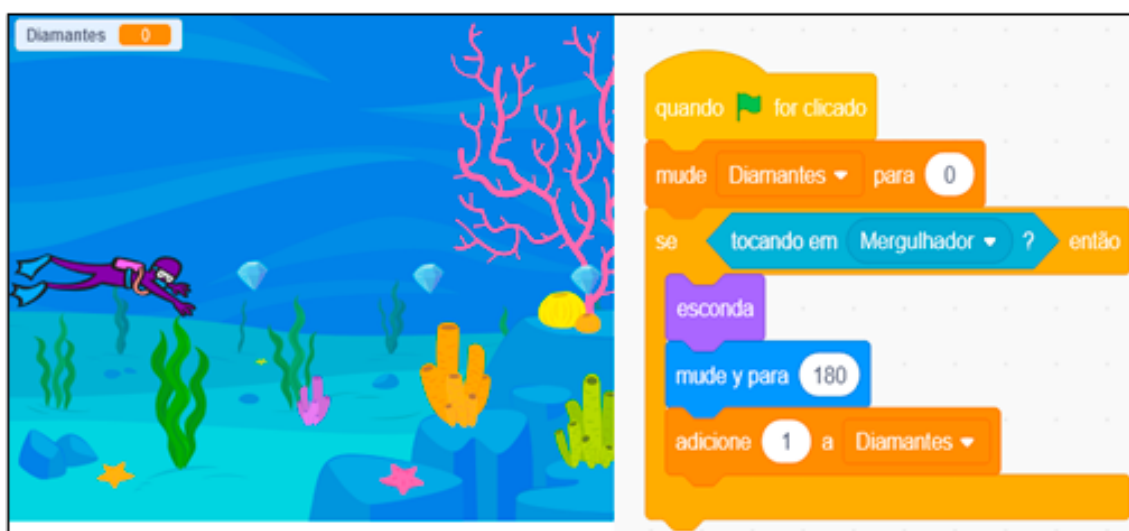


Figura 2 – Atividade com novos elementos no cenário.

Fonte: Ferramenta Scratch.



Figura 3 – Telas do jogo de perguntas e respostas.

Fonte: Ferramenta Scratch.

Após concluir o primeiro jogo proposto, os alunos iniciaram o desenvolvimento de mais um jogo. O mesmo foi composto por 3 diferentes telas, nas quais desenvolveram a tela inicial do jogo, apresentada com o nome do jogo e um botão para começar a jogar, onde foram programados efeitos visuais com cores. Em seguida iniciou-se o desenvolvimento da tela do jogo, onde o objetivo era mover o balde para o lado direito e esquerdo, na intenção de coletar apenas as frutas que estivessem caindo. Observe na Figura 4 as telas do jogoproposto.



Figura 4 – Telas de jogo coletando frutas.

Fonte: Ferramenta Scratch.

O jogo coletando frutas foi desenvolvido no período de 6 aulas, por se tratar de uma programação maior e mais complexa, onde foram programados sensores, efeitos de sons e de cores, além de vários códigos que seriam executados em paralelo.

Os dois jogos desenvolvidos foram as programações mais complexas realizadas no curso, que exigiram aprendizados obtidos através das aulas e também das atividades realizadas. Para atividades mais complexas é interessante observarmos alguns indicadores como: (abstração, controle de fluxo, representação de dados, paralelismo, lógica, sincronização e interação com usuário) que são analisados através da ferramenta especializada em avaliação automática, o *Dr Scratch*. De acordo com a quantidade de jogos entregue no Gráfico 1, é possível observar que dos 11 alunos, apenas 7 entregaram o jogo de perguntas e respostas. Os 7 jogos foram avaliados de forma automática e apresentaram pontuações diferentes nos indicadores analisados pelo *Dr. Scratch*.

Analisando o Gráfico 1, **o conceito de abstração**, em nenhum momento esteve em alta, pois o bloco clássico que representa este conceito é o clone, que pode ser utilizado para criar cópias de partes da programação, evitando que o mesmo código precise ser programado, e essa funcionalidade por ser avançada, foi pouco enfatizada no início das aulas, desta forma sendo pouco utilizada.

Já outro conceito que esteve em baixa foi o **controle de fluxo**, que é acionado quando a programação é construída para que haja controle de fluxo das ações do personagem, por exemplo, determinando quantas vezes ele poderá repetir uma ação. Neste caso nem todos os alunos utilizaram os determinados blocos voltados para essa função, pois visualizaram que poderiam realizar a mesma programação fazendo uso de outros blocos.

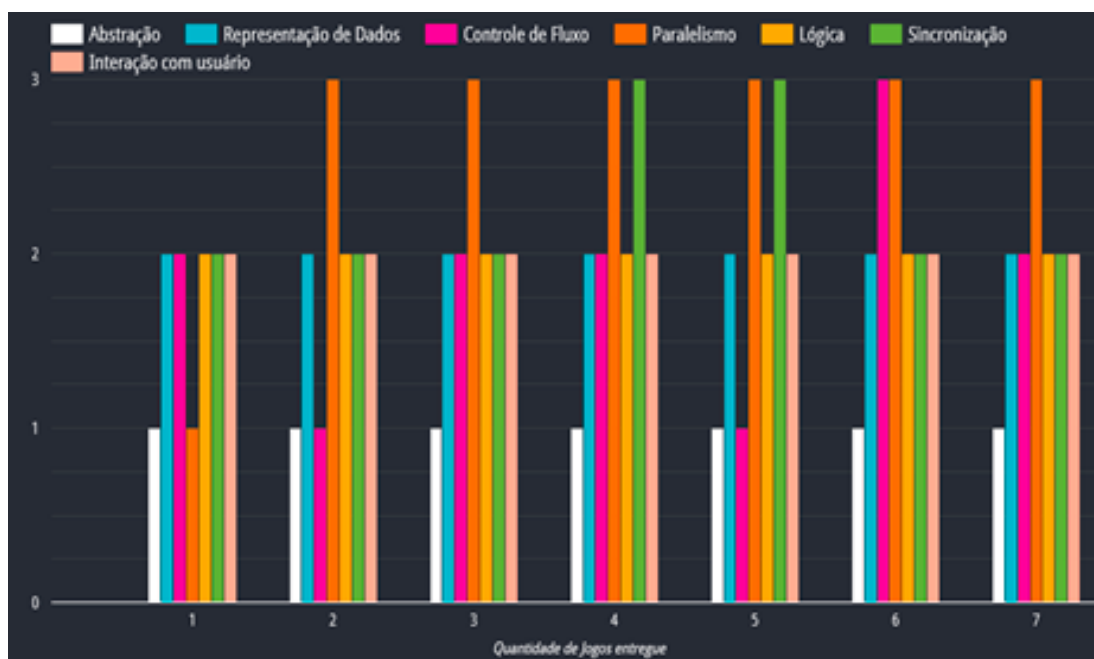


Gráfico 1 – Avaliação dos conceitos do jogo de perguntas e respostas.

Fonte: Autoria Própria

Já analisando o Gráfico 2, na avaliação do jogo coletando frutas, 9 jogos foram avaliados, e a maioria obteve as mesmas pontuações nos conceitos. Os dois conceitos com as maiores pontuações foram controle de fluxo e paralelismo, já que para desenvolver esse jogo vários personagens possuíam movimentos ao mesmo tempo, tendo seus códigos executados em paralelo, e esses movimentos precisaram de um controle de fluxo para controlar até que momento uma ação poderia ocorrer, quando a mesma iria iniciar e quando seria o seu término.

Ao fim do curso todos os objetivos propostos foram supridos, pois de fato as aulas conseguiram estimular a criatividade, a autonomia e o desenvolvimento do raciocínio lógico dos alunos, através de atividades que apresentavam problemas que necessitavam de atenção e criatividade para serem resolvidos.

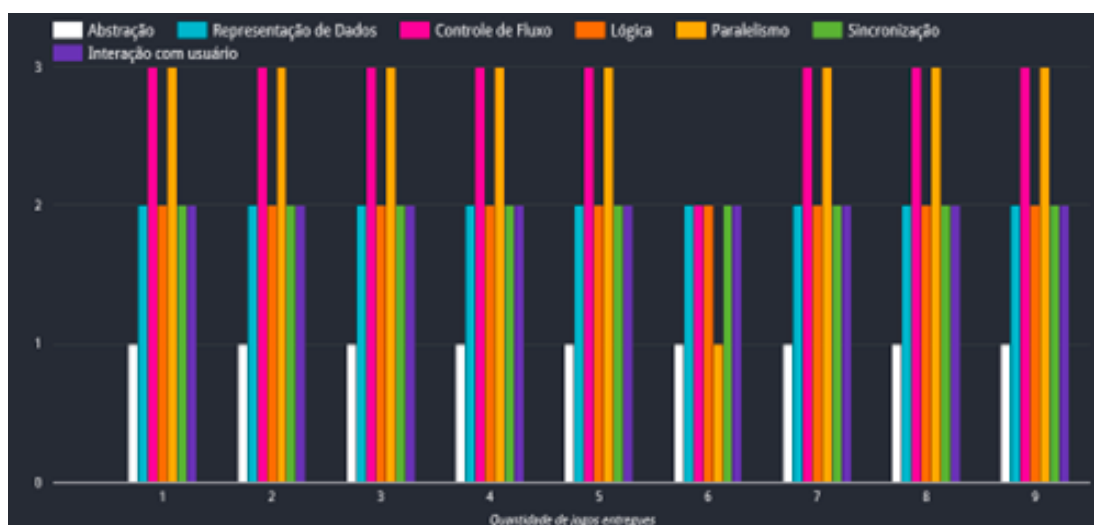


Gráfico 2 – Avaliação dos conceitos do jogo coletando frutas.

Fonte: Autoria Própria

Os 11 alunos concluíram o curso de forma satisfatória, obtendo o conhecimento de programação em blocos para desenvolver suas próprias animações e jogos com diferentes graus de complexidade. Além disso, pode-se constatar a eficácia do curso, por meio dos dados mostrados neste relato. Acreditasse que a utilização da programação, além de fascinar os alunos, colaborou significativamente para se ter um período de férias diferente e engajador, onde foram transmitidos conhecimentos de forma lúdica e divertida, proporcionando novos aprendizados.

Contribuições dos Autores

Os autores contribuíram de forma igual em todas as seções na elaboração do relato de experiência.

Referências

1. França, R. S. et al. (2014). "A disseminação do pensamento computacional na educação básica: lições aprendidas com experiências de licenciandos em computação". **In: Anais de Congresso da Sociedade Brasileira de Computação, Brasília, SBC**, p. 1473- 1475.
2. Hoed, R. M. (2016). Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de Computação. Brasília, DF: Universidade de Brasília.
3. De Castro, T. H. C., De Castro Júnior, A. N., De Menezes, C. S., Boeres, M. C. S., & Rauber, M. C. P. V. (2003). Utilizando programação funcional em disciplinas introdutórias de computação. **Anais do WEI**.
4. Wing, J. (2016). Pensamento computacional: um conjunto de atitudes e habilidades que todos, não só cientistas da computação, ficaram ansiosos para aprender e usar. **In: Revista Brasileira de Ensino e Tecnologia**, <https://periodicos.utfpr.edu.br/rbect/article/view/4711/pdf>.
5. Nunes, D. J. (2011). "Ciência da computação na educação básica". <http://www.adufrgs.org.br/artigos/ciencia-da-computacao-na-educacao-basica>. Fevereiro.
6. Zanatta, A. C. (2017) Desenvolvimento de Jogos para Estimular a Programação/Lógica em crianças de 9 a 12 anos I Simpósio Ibero-Americano de Tecnologias Educacionais - **SITED 2017** Universidade Federal de Santa Catarina - UFSC.
7. Bopprê, V. (2013) "Ensinar a programar é ensinar a pensar: Especialistas defendem o ensino da programação nas escolas como meio de estimular a criatividade dos alunos". Disponível em: <https://porvir.org/ensinar-programar-e-ensinar-pensar/>
8. MALONEY, J. et al. (2010) The scratch programming language and environment. **In: Communications of The ACM**, 2010. v. 10, n. 4, Article 16.
9. SCRATCH. About Scratch (Scratch Documentation Site, 2015).
10. MORENO-LEON, Jesus; ROBLES, Gregorio; ROMAN-GONZALEZ, Marcos. Comparing computational thinking development assessment scores with software complexity metrics. **IEEE Global Engineering Education Conference, EDUCON**, v. 10-13-April-2016, n. April, p. 1040-1045, 2016.