

# **A importância da análise e replicação de códigos para o desenvolvimento da lógica de programação.**

**Yuri S. S. de Oliveira<sup>1</sup>, Ewerton G. S. de Souza<sup>1</sup>, Eglison R. V. Capezzeria<sup>1</sup>,  
Humberto Constantino de Andrade Silva Neto<sup>1</sup>, Victor F. L. Silva<sup>1</sup>, Jonnatha J. O.  
Vitalino<sup>1</sup>**

<sup>1</sup>IESO, Instituto de Ensino Superior de Olinda, Brasil  
CEP 53010-240 – Olinda, PE

yurisobral00@gmail.com, ewerton.gustavo097@gmail.com, rodrigo.capezzeria20@gmail.com,  
humbertoconstantino18@gmail.com, victor-fernando-15@hotmail.com,  
jonnatha.profissional@hotmail.com

**Resumo.** *Percebe-se cada vez mais que o desenvolvimento da lógica de programação entre os alunos que iniciam nos cursos de TI do ensino superior e até entre profissionais da área no início de seus estudos vem sendo uma grande barreira, afirmação essa embasada por uma pesquisa por amostragem realizada com cinquenta indivíduos. Assim sendo, este documento buscou mostrar quão importante é o uso de métodos de aprendizado para o desenvolvimento da mesma, enfatizando o método de análise e replicação de códigos e a sua importância para a lógica de programação e para os profissionais e futuros profissionais da área de TI.*

**Abstract.** *It is increasingly perceived that the development of programming logic among students who start in IT courses in higher education and even among professionals in the area at the beginning of their studies has been a major barrier, a statement based on a survey by sampling conducted with fifty individuals. Thus, this document sought to show how important is the use of learning methods for the development of the same, emphasizing the method of analysis and replication of codes and its importance for the programming logic and for professionals and future professionals in the IT area.*

## **1. INTRODUÇÃO**

O objetivo deste documento é apresentar a importância da Análise e Replicação Códigos (ARC) para o desenvolvimento da lógica de programação. O assunto será explorado para podermos entender melhor como ajudar os estudantes de cursos que envolvam programação de computadores. Assim sendo, é apresentada a importância do uso de exemplos de códigos prontos para amadurecimento do raciocínio lógico. O foco da ARC é contribuir com o aprendizado de uma linguagem de programação, como por exemplo, Java, C ou Python. O raciocínio lógico não muda muito entre linguagens, pois - ao desenvolver o raciocínio lógico para uma linguagem de programação a barreira do aprendizado é minimizada ao iniciar estudos em novas linguagens de programação. Por fim, a lógica de programação é capaz de estimular habilidades como, por exemplo, o pensamento analítico.

O pensamento analítico é uma forma de pensamento. O objetivo dele é explicar as coisas por meio da decomposição em partes menores, que são mais facilmente explicadas ou solucionadas, e uma vez entendidas tornam possível a compreensão do

todo. O comportamento da totalidade é assim deduzido e explicado pelo comportamento das partes. O mau desenvolvimento deste e outros conceitos geram a atual falta de profissionais na área de TI que serve de alerta para os problemas na base do aprendizado e esses problemas podem gerar desmotivação, perda de interesse e até a evasão do curso, estes são grandes desafios da educação em informática (FERREIRA et al., 2010). O método ARC contribui na solução deste problema despertando como, por exemplo, a concentração, a curiosidade e o interesse do aluno.

O restante deste documento está organizado da seguinte maneira: seção 2 apresenta os principais termos e conceitos para entendimento da ARC. Seguidamente, na Seção 3 são apresentados os aspectos metodológicos para realização do trabalho, como por exemplo, método de coleta e análise dos dados. Na seção 4, são apresentados os resultados e discussão desta pesquisa. Por fim, na seção 5, são apresentadas as considerações finais assim como propostas de trabalhos futuros.

## **2. FUNDAMENTAÇÃO TEÓRICA DA ARC**

A ARC se apresenta como uma importante ferramenta para a solução de problemas dos profissionais no âmbito da programação e também de fundamental importância para os alunos da área de TI, pois, possibilita que o aluno possa se basear em códigos e soluções de outras pessoas para que os mesmos possam incorporar essas informações a sua biblioteca de conhecimentos, assim, desenvolvendo suas próprias capacidades de lógica e técnicas de programação. Desde os primórdios os homens utilizam o método da observação para entender e aplicar novos conhecimentos, por exemplo, a história de Isaac Newton e a maçã, a qual é citada na biografia escrita por William Stukele, “Memórias da vida de Sir Isaac Newton”: “Depois do jantar, o clima é quente, fomos para o jardim beber chá, sob a sombra de algumas macieiras ele me disse que só estivera na mesma situação, como quando anteriormente, a noção de gravitação veio em sua mente. Foi ocasionada pela queda de uma maçã, enquanto estava sentado em atitude contemplativa. Por que a maçã sempre desce perpendicularmente ao chão? Pensou...”. Desta mesma forma funciona a ARC, onde o estudante por meio de observação entende a lógica por trás da solução desejada, então ele desenvolve réplicas para aprofundar e concretizar esses novos conhecimentos, podendo assim resolver futuros problemas com mais facilidade.

A ARC utiliza alguns dos conceitos da aprendizagem por observação como, por exemplo, a imitação que é entendida como a reprodução do comportamento exposto ou explicado por um modelo, que, sofreu grande impulso a partir dos escritos de Miller e Dollard (1941), que apresentaram uma avaliação de imitação no começo dos anos quarenta. Com o propósito de explicar o que é entendido como aprendizagem por observação. O conceito do pensamento computacional também faz parte da ARC, na visão de Wing (2006), o pensamento computacional é o processo de reconhecer aspectos da computação no mundo que nos rodeia e aplicar ferramentas e técnicas para entender e raciocinar sobre sistemas e processos naturais, sociais e artificiais. Já Segundo Selby e Woollard (2013), o pensamento computacional é um processo cognitivo ou de pensamento que envolve o raciocínio lógico pelo qual os problemas são resolvidos e os artefatos, procedimentos e sistemas são melhor compreendidos, tendo como abrangência a capacidade de pensar em: algoritmos, termos de decomposição, generalizações, identificando e fazer uso de padrões, abstrações, escolhendo as representações mais adequadas de acordo com cada contexto e termos de avaliação. “Além da leitura e escrita deve-se acrescentar o pensamento computacional às habilidades analíticas das crianças.” (WING, 2006).

### 3. PERCURSO METODOLÓGICO

Esta pesquisa foi desenhada com base no método científico. Esse método é composto por um conjunto de procedimentos que permitem operacionalizar o trabalho para que seja possível ter acesso às relações de causa e efeitos, ou seja, é por meio da observação dos fatos que é possível questionar os ocorridos e a partir desses achados problematizá-los por meio de pontos de vista. Assim sendo, o método científico contribuiu não apenas com as regras e procedimentos para a realização da pesquisa, mas também teve sua contribuição para apoiar o saber científico.

Dessa forma, a problemática da pesquisa parte da curiosidade em conhecer a relevância da análise e replicação de códigos no ensino aprendizagem de linguagens de programação por meio da consulta de questionário. Note-se que para chegar a tal objetivo foi realizada uma pesquisa bibliográfica, que foi útil tanto para embasar os conceitos da pesquisa como para fundamentar o questionário e análises de dados. A análise, organização e interpretação desses dados foram feitas mediante principalmente por técnicas estatísticas.

A análise de dados reúne várias etapas para extrair significação aos resultados obtidos. No que se refere a essas diferentes etapas optou-se por organizar em três fases: 1) pré-análise; 2) exploração dos resultados; e 3) tratamento dos resultados, inferência e interpretação.

A pré-análise é responsável pela organização do material a ser analisado de modo sistemático. Em resumo, trata-se da organização dos dados por meio de quatro etapas: (a) escolha dos dados, que consiste na demarcação do que será analisado; (b) extração dos resultados, que diz respeito a coleta de dados e organização dos dados em documentos; (c) formulação das hipóteses e dos objetivos; (d) elaboração de indicadores.

A exploração dos resultados é a segunda fase e consiste na decomposição e padronização destes resultados. A organização é feita procurando categorizar e quantificar valores, possibilitando interpretações e inferências. Esta é a uma tarefa analítica.

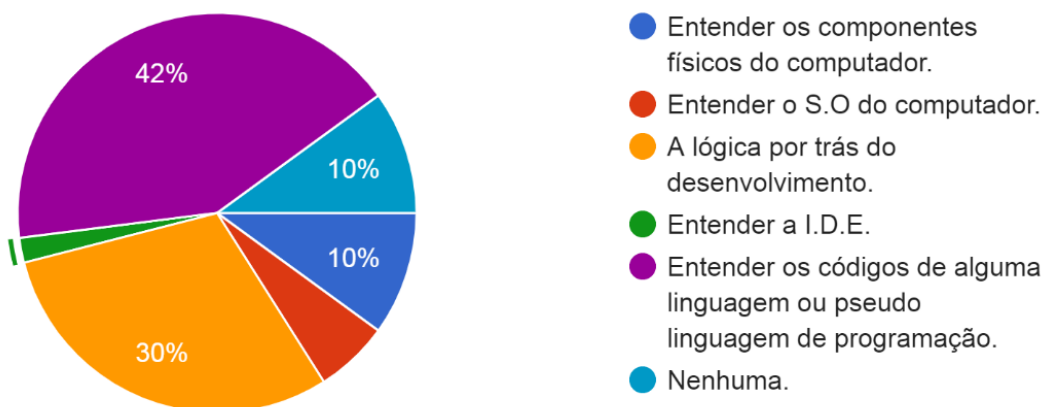
No que diz respeito ao tratamento dos resultados, inferência e interpretação são destinadas ao tratamento dos resultados; ocorre nela o resumo e o destaque das informações para análise, levando as interpretações inferenciais. A seguir são apresentados os resultados e discussão.

### 4. RESULTADOS E DISCUSSÃO

Em relação aos estudos sobre o desenvolvimento de softwares, na figura 1 a maior dificuldade encontrada pelo público alvo foi entender os códigos de alguma linguagem ou pseudocódigo de linguagem de programação (42%), seguida pela lógica por trás do desenvolvimento (30%).Essas dificuldades podem estar relacionadas a entender o raciocínio lógico e os comandos de programação. Segundo Esteves et al. (2007) os alunos necessitam de aprender a resolver problemas e elaborar algoritmos, o que dificulta a aprendizagem da programação, o que resulta em um elevado índice de insucesso às disciplinas de programação.A dificuldade menos votada foi entender o ambiente de desenvolvimento integrado ou “IntegratedDevelopmentEnvironment”(IDE). Alguns problemas iniciais como erros de

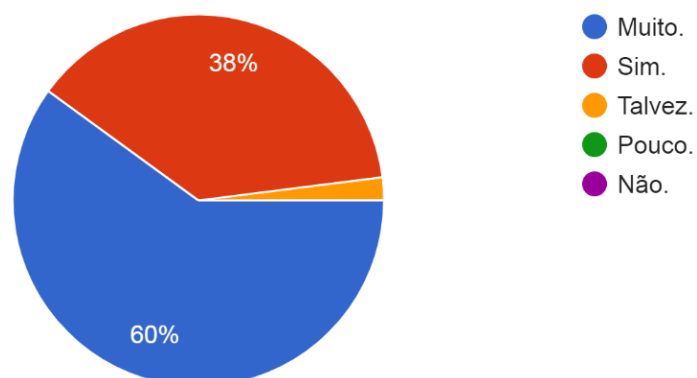
sintaxe ou mesmo de lógica, podem ser minimizados com a ajuda de “IDEs” que ofereçam um bom editor integrado ao interpretador/compilador da linguagem e a uma ferramenta de depuração (RUSSI; CHARÃO, 2011).

Figura 1– No início dos seus estudos sobre o desenvolvimento de softwares qual foi sua maior dificuldade?



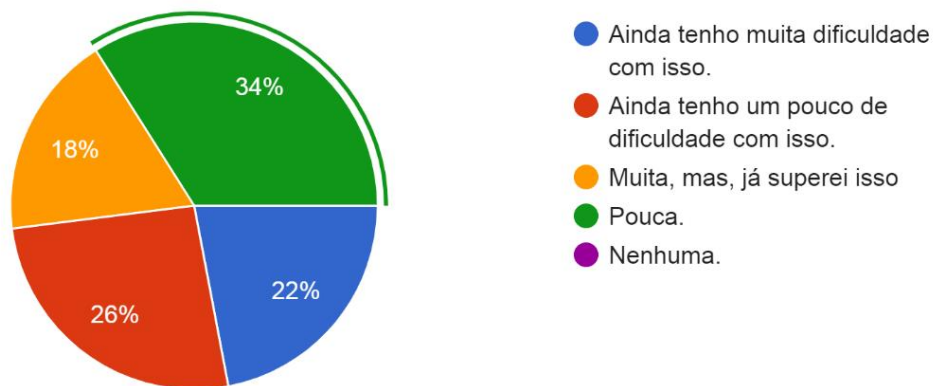
A maioria dos participantes (60%) responderam que é muito importante desenvolver seu pensamento lógico como estudante ou profissional da área de TI(Figura 2). A possibilidade de desenvolver sua própria lógica de programação é fundamental para o público da área, e com isso a importância de aprender esses conceitos básicos nos anos iniciais dos cursos da área de TI. A aprendizagem de conceitos e métodos para a construção de programas de computado não é trivial, pois requer o uso de habilidades de alto nível e boas doses de raciocínio abstrato (CASTRO et al., 2003).

Figura 2–Sendo você um estudante ou profissional da área de TI, você acha importante desenvolver seu pensamento lógico?



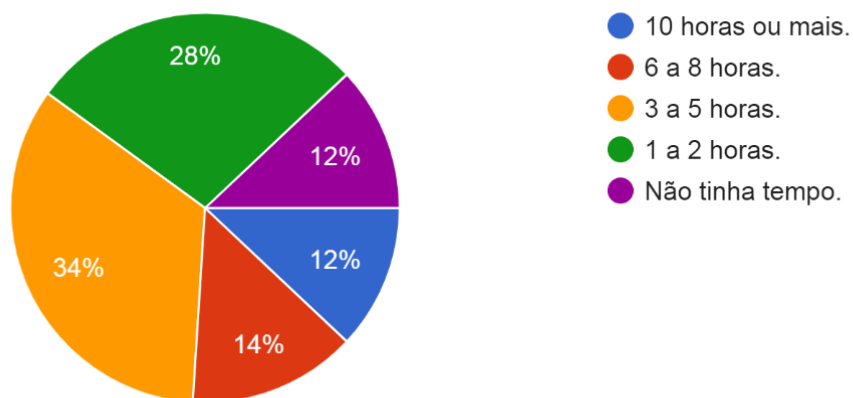
Em relação a importância do pensamento lógico no dia a dia de trabalho ou na faculdade, a maioria respondeu que é muito importante (58%). Apenas 4% acham pouco importante o pensamento lógico. Na figura 3 é possível visualizar que pouca dificuldade foi encontrada para começar a pensar logicamente e conseguir resolver problemas e desenvolver seus códigos em 34% dos pesquisados, seguido por 26% que ainda tem pouca dificuldade com isso. Programação é uma tarefa desafiadora e, para amenizar as dificuldades no seu aprendizado, têm sido adotadas diversas abordagens metodológicas para seu ensino em cursos de graduação (CASTRO et al., 2003). Ao longo dos cursos das áreas de TI, são oferecidas disciplinas que tendem a estimular o pensamento lógico do aluno, o que pode facilitar o processo de aprendizagem. Vieira et al. (2010) citam que disciplinas introdutórias da área de Arquitetura e Organização de Computadores podem auxiliar neste processo. Neste sentido, os conhecimentos básicos para a compreensão dos conceitos apresentados nestas disciplinas, fornecem subsídios fundamentais para a aprendizagem e compreensão da lógica de programação (VIEIRA et al., 2010).

Figura 3—Ainda sobre o pensamento lógico, quanta dificuldade você teve para começar a pensar logicamente e conseguir resolver problemas e desenvolver códigos?



Para desenvolver o pensamento lógico com exercícios ou outros meios quando não estavam na faculdade, 34 % responderam que se dedicaram entre 3 e 5 horas, 12% estudaram 10 horas ou mais, e outros 12 % não tiveram tempo para estudar (Figura 4). Porém o estudo de lógica pode exigir uma maior dedicação dos alunos. Em suas pesquisas, Norvig (2002) demonstrou que se leva, aproximadamente, 10 anos para ser um especialista em qualquer área do conhecimento, o que inclui programação.

Figura 4—No início dos seus estudos, quantas horas por semana você se dedicou a desenvolver seu pensamento lógico quando não estava na faculdade?



O método mais utilizado para solucionar exercícios e desenvolver o pensamento lógico (Figura 5), foi a própria lógica do pesquisado (38%), seguida pela análise e réplica de outros códigos (28%), fluxograma (14%), e pseudocódigo (14%). Criar a própria lógica pode ser mais fácil para alguns estudantes, do que tentar entender a lógica de outras pessoas.

Figura 5—Que método você utilizou para solucionar exercícios e desenvolver seu pensamento lógico?

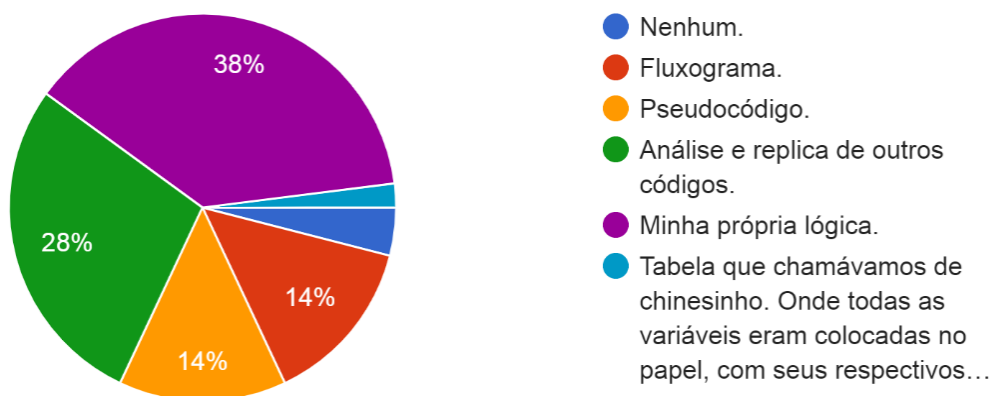
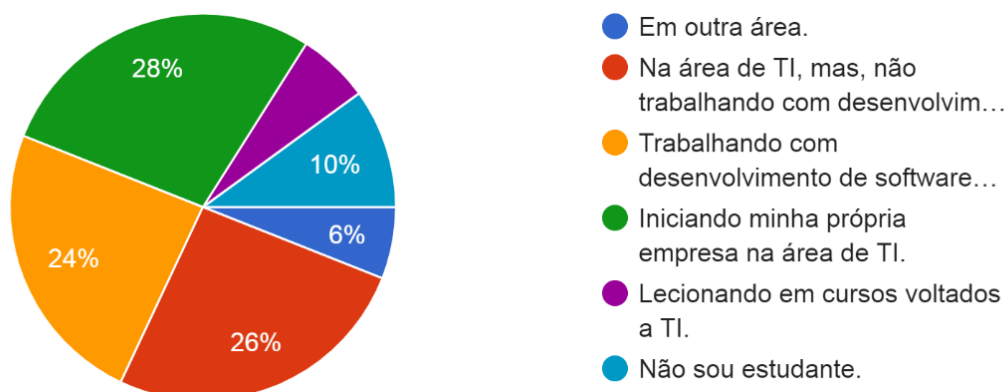


Figura 6—Para os estudantes, como você se vê daqui a alguns anos em sua carreira profissional?



Dos participantes, 10% não eram estudantes. Para os profissionais de TI, a maioria está satisfeito com a sua atual posição no mercado de trabalho (Figura 7). Em relação ao que os estudantes almejam em sua futura carreira profissional, 28% dos estudantes desejam iniciar sua própria empresa de TI, 26% pretendem continuar na área, mas não trabalhando com desenvolvimento, e 24% pretendem trabalhar na área de desenvolvimento de software. Cerca de 6% pretendem lecionar em cursos voltados para TI. O mercado para criação de novas empresas na área de TI vem crescendo. Os startups ou empresas nascentes têm se expandido no Brasil e no mundo nos últimos anos, principalmente na área de tecnologia e inovação, com altos valores agregados (NAGAMATSU et al., 2014). E com incentivos e surgimentos de novas idéias ficou mais fácil criar a própria empresa e crescer no ramo de TI por conta própria.

Figura 7—Para os que já são profissionais de TI, quão satisfeito você está na sua atual posição no mercado de trabalho?

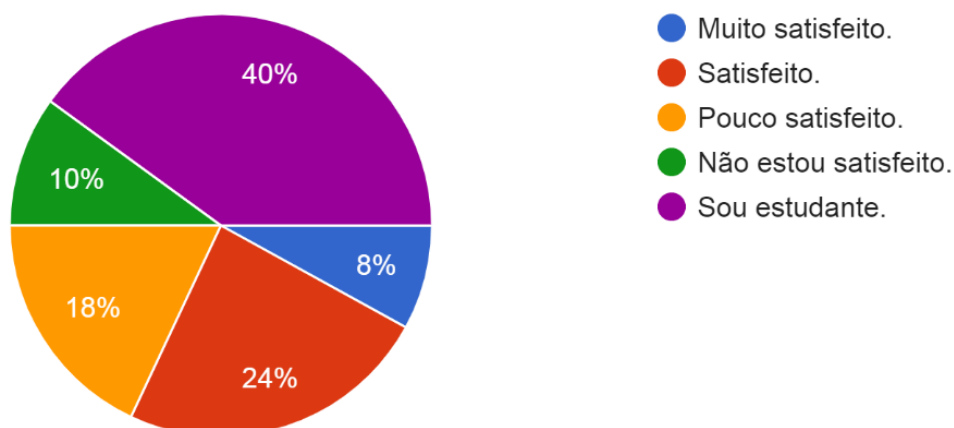
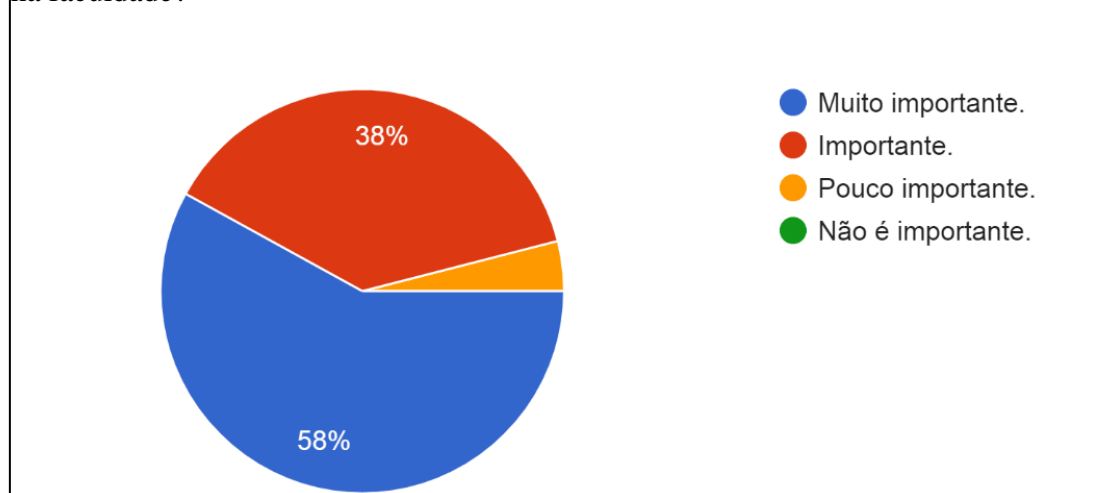


Figura 8—Quão importante é o pensamento lógico para o seu dia a dia no trabalho ou na faculdade?



## 5. CONSIDERAÇÕES FINAIS

Em vista dos argumentos apresentados neste trabalho acadêmico, pudemos perceber que a ARC é de suma importância para quem procura desenvolver sua lógica de programação, o seu raciocínio lógico e que ela pode ser aproveitada em qualquer linguagem de programação como também em toda a área de TI, pois, estimula a busca de conhecimento, o pensamento computacional e a criatividade. A observação e a replicação de códigos são os pilares da ARC, ela leva o aluno e o profissional de TI a desenvolver o seu pensamento crítico como também a sua técnica de codificação, expande a sua forma de entender e perceber a lógica, as inúmeras formas de utilizar a mesma e o pensamento computacional, assim expandindo sua biblioteca de conhecimentos cada vez mais, resultando no aprimoramento de suas habilidades. Na análise dos dados coletados em uma pesquisa realizada por amostragem, tendo esta mesma como amostra, pessoas da área de TI, pudemos identificar que a maior dificuldade encontrada por eles foi: “Entender os códigos de alguma linguagem ou pseudo linguagem de programação”, dificuldade está que poderia ser amenizada ou até mesmo resolvida através do método ARC, que não apenas mostraria a solução, mas, traria consigo muitos outros benefícios que já foram citados, tais como: desenvolvimento do pensamento computacional, da lógica aplicada ao código e tudo isso sendo absorvido pelo indivíduo através da prática e estudo diário por meio da ARC. Já os participantes que utilizavam a ARC demonstraram ter tido menos problemas no desenvolvimento da lógica de programação.

Todos esses pontos tornam evidente a importância da ARC no desenvolvimento da lógica de programação, mas, qualquer método de aprendizado só funcionará se estiver ligado a boas horas de esforço e interesse do próprio aluno. Encerramos este artigo esperando que sendo você estudante ou profissional de TI, possa ter entendido os benefícios da ARC e que possa incorporar esse método ao seu dia-a-dia de estudos e de trabalho.

Como trabalho futuro, um estudo semelhante pode ser conduzido com ênfase apenas na ARC e seus benefícios, tendo como amostra apenas estudante da área de TI. Para então podermos determinar de forma mais exata se os mesmos conhecem este método e se sabem dos benefícios de utilizá-lo.



## REFERÊNCIAS

CASTRO, Thais Helena Chaves et al. Utilizando programação funcional em disciplinas introdutórias de computação. **Anais do WEI**, 2003.

ESTEVES, Micaela et al. Contextualização da aprendizagem da programação: estudo exploratório no Second Life. **Proceedings of IADIS Ibero-Americana WWW/Internet**, p. 7-8, 2007.

NAGAMATSU, Fabiano Akiyoshi; BARBOSA, Janaina; REBECCHI, Adriana. Business model generation e as contribuições na abertura de startups. 2014.

NORVIG, P. Russel; INTELLIGENCE, S. Artificial. **A modern approach**. Prentice Hall, 2002.

RUSSI, Davi Felipe; CHARÃO, Andrea Schwertner. Ambientes de Desenvolvimento Integrado no Apoio ao Ensino da Linguagem de Programação Haskell. **RENOTE**, v. 9, n. 2, 2011.

VIEIRA, Paulo Vinícius; RAABE, André Luis Alice; ZEFERINO, Cesar Albenes. Bipide-ambiente de desenvolvimento integrado para a arquitetura dos processadores BIP. **Brazilian Journal of Computers in Education**, v. 18, n. 01, p. 32, 2010.

ANDRADE, Jairo Eduardo Borges. Aprendizagem por observação: perspectivas teóricas e contribuições para o planejamento instrucional - uma revisão. **Psicol. cienc. prof.** v.1 n.2 Brasília jul. 1981”[http://pepsic.bvsalud.org/scielo.php?script=sci\\_arttext&pid=S1414-98931981000200001#back2](http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1414-98931981000200001#back2)”.

FERREIRA, Cláudia; GONZAGA, Flávio; SANTOS, Rodrigo. Um Estudo sobre a Aprendizagem de Lógica de Programação Utilizando Programação por Demonstração.” **FAGOC, Faculdade Governador Ozanam Coelho, Brasil – Ubá, MG; UNIFAL, Universidade Federal de Alfenas, Brasil – Alfenas, MG; COPPE/UFRJ, Universidade Federal do Rio de Janeiro, Brasil – Rio de Janeiro, RJ**, 2015. “[http://www.bcc.unifal-mg.edu.br/lares/files/artigos/Um\\_estudo\\_sobre\\_a\\_aprendizagem\\_de\\_logica\\_de\\_programacao\\_utilizando\\_pbd.pdf](http://www.bcc.unifal-mg.edu.br/lares/files/artigos/Um_estudo_sobre_a_aprendizagem_de_logica_de_programacao_utilizando_pbd.pdf)”