

# CODEX - Learning Object to Support Loop Structure Teaching

Bruno Gonçalves  
Departamento de Ciência da Computação  
Universidade de Brasília  
Brasília, Brasil  
brunoag8@gmail.com

Breno Teixeira  
Departamento de Ciência da Computação  
Universidade de Brasília  
Brasília, Brasil  
brenotx@gmail.com

Leticia Lopes Leite  
Departamento de Ciência da Computação  
Universidade de Brasília  
Brasília, Brasil  
llete@unb.br

Marcia Cristina Moraes  
School of Education  
Colorado State University  
Fort Collins, USA  
marciacristina.moraes@gmail.com

**Abstract**— According to the Code.org there are currently 528,174 open computing jobs in U.S. and only 49,291 computer science students graduated into the workforce, last year. In order to try to reverse this scenario, several countries are working on a way to include computer science concepts, like computational thinking, in their curriculums for K-12 education. In addition, many studies and researches are being done related to practices and tools that can be applied in order to retain students that applied for computer science courses. In order to contribute to these initiatives, this paper presents CODEX, an interactive and ludic learning object, that aims to help students that do not need to have a background in programming to learn one of the most important structure in a programming language, the loop. CODEX was designed to be used both in class and in extra-class activity.

**Keywords**—Learning Objects, teaching programming, repetition.

## I. INTRODUÇÃO

De acordo com a organização Code.org, atualmente existem 528.174 vagas disponíveis nos Estados Unidos da América (EUA) e somente 49.291 estudantes foram graduados em cursos relacionados à Ciência da Computação no último ano (Code.org, 2018). Giraffa et al. (2015) apontam que a baixa procura por cursos das áreas de Engenharia e Computação, bem como o alto índice de evasão em tais cursos, são alguns dos motivos que têm contribuído para esse cenário. A fim de reverter esse cenário, vários países têm trabalhado em formas de incluir conceitos de Ciência da Computação, como o pensamento computacional, em seus currículos desde a Educação Básica até o Ensino Médio.

Em 2016, os EUA lançaram a iniciativa *Computer Science for All* que tem como objetivo empoderar estudantes americanos com habilidades de Ciência da Computação necessárias em uma economia digital (Smith, 2016). Em dezembro de 2016 a União Européia publicou um relatório apresentando como os diferentes países do bloco têm integrado o pensamento computacional na educação compulsória, por meio da programação, educação em Ciência da Computação ou outras disciplinas como matemática. Em 2017, a Comissão de Educação da Sociedade Brasileira de Computação (SBC) aprovou o relatório intitulado " Referenciais de Formação em

Computação: Educação Básica" que foi apresentado no XXXVII Congresso da Sociedade Brasileira de Computação (CSBC 2017) durante as Assembléias do Workshop de Educação em Informática (WEI) e da SBC (Raabe et al., 2017).

Diversos estudos e pesquisas têm apontado os muitos desafios relacionados ao ensino de programação (Giraffa et al., 2015), bem como têm proposto práticas pedagógicas e ferramentas que auxiliem na retenção de estudantes em diferentes cursos de computação. Dentre os desafios, destacamos a capacidade dos estudantes em assimilar conceitos fundamentais que servem como alicerce para a aprendizagem da programação. A fim de contribuir para a superação deste desafio, este artigo apresenta o CODEX, um objeto de aprendizagem lúdico e interativo que visa auxiliar estudantes, que não necessitam ter conhecimento prévio de programação, a aprender um dos mais importantes conceitos em linguagem de programação, o laço de repetição.

## II. ENSINO DE PROGRAMAÇÃO

Nos cursos que se destinam à formação de profissionais da área de Computação, o debate sobre as potencialidades, os avanços e as dificuldades relacionadas ao ensino estão bastante presente. Uma grande dificuldade encontra-se no primeiro contato do estudante com a linguagem computacional, que geralmente é feito em disciplinas relacionadas à área da Computação. Essa iniciação se apresenta como um desafio, tanto para o professor quanto para o aluno.

Conforme Souza *et al.* (2016), os estudantes têm bloqueios para entender determinados conceitos de programação, tais como ponteiros, recursão, declaração de variáveis, dentre outros. E, alguns estudantes apesar de entenderem, não conseguem aplicá-los com facilidade durante a construção de programas, tornando-se fundamental que os professores façam uso de práticas pedagógicas que facilitem a compreensão dos conceitos e a construção do conhecimento. De acordo com Pereira Júnior e Rapkiewicz (2016) e Souza *et al.* (2016), o ensino e a aprendizagem de programação são considerados tarefas complexas e, como consequência, os cursos de programação frequentemente têm altas taxas de reprovação e desistência.

O debate sobre o ensino da programação é amplo e envolve pesquisas que englobam desde a Educação Básica até a Educação Superior. Lima Júnior *et al.* (2015) apontam que uma das metas dos cursos de Computação está definida em torno da capacidade do estudante apresentar soluções para diversas classes de problemas encontrados no cotidiano das pessoas, das organizações e de muitos outros elementos. Dessa forma, em caráter introdutório, o autor salienta a necessidade de se fornecer aos estudantes as bases necessárias para o desenvolvimento da lógica de programação e representar o raciocínio envolvido por meio de algoritmos nexos e corretos. Atualmente, ainda são poucas as instituições que se preocupam em trabalhar as noções de programação na Educação Básica e em adotar em seu projeto pedagógico essa temática. Porém, estudos afirmam que o raciocínio lógico essencial à programação apresenta reflexos positivos no desenvolvimento de estudantes (Raabe e Silva, 2005).

De acordo com Souza (2016), historicamente o ensino de programação tem sido considerado de difícil entendimento para os alunos pelos seguintes motivos: falta de preparo dos estudantes, ausência de uma didática adequada e de ferramentas computacionais que ajudem os envolvidos (professores e alunos) a superarem os problemas que se apresentam no processo de ensino e aprendizagem. O autor ainda afirma que, geralmente, os cursos de programação iniciam-se com um pseudocódigo, uma linguagem simples e natural que não exige conhecimento de nenhuma sintaxe de linguagem de programação. Um ponto positivo dessa abordagem tradicional é a possibilidade de usar apenas lápis e papel para se codificar um algoritmo, sem a necessidade de computadores. Já, um aspecto negativo é que, para a maioria dos estudantes, esta abordagem é muito abstrata, pois não conseguem associar os comandos que escrevem com as execuções e respostas do programa.

O trabalho de Lima Júnior *et al.* (2015) apresenta um mapeamento de pontos críticos que geram os maiores problemas de aprendizagem. Ele foi realizado visando a definição de medidas corretivas que podem contribuir para melhorar o aprendizado da disciplina de Algoritmos e Programação de Computadores. Destacam-se seguintes pontos críticos: raciocínio lógico, capacidade de abstração, leitura e interpretação de texto e conhecimentos matemáticos. E, dentre os conteúdos em que os alunos encontram mais dificuldades estão: matrizes, vetores, estruturas de repetição e estruturas condicionais.

A partir das pesquisas apresentadas, percebe-se que a abstração inicial da programação, no que se refere a conceitos de programação e raciocínio lógico, é um determinante no desenvolvimento do aprendizado e que o raciocínio requisitado muitas vezes foge à realidade dos estudantes. Diante deste cenário e baseados na pesquisa apresentada por Souza *et al.* (2016), que identifica alternativas para solucionar os problemas oriundos do ensino de programação, consideramos que um aplicativo gamificado pode apoiar significativamente o ensino e a aprendizagem de programação. Estes jogos podem ser utilizados para treinar pessoas por meio de um ambiente virtual que imita a realidade, de forma a mapear conceitos de programação a partir de uma situação cotidiana. Atualmente, uma das possibilidades de implementação dos referidos jogos

educativos, ocorre a partir de Objetos de Aprendizagem (Oas\_, detalhados na próxima seção.

### III. OBJETOS DE APRENDIZAGEM

Para Koper (2003), um Objeto de Aprendizagem é definido como qualquer recurso digital, reproduzível e referenciável, utilizado em atividades de aprendizagem ou de apoio à aprendizagem, disponível para que outras pessoas o utilizem. De forma simplificada, Objeto de Aprendizagem é um recurso digital que tem a intencionalidade educativa e pode ser utilizado para complementar e aperfeiçoar os estudos, a fim de atingir a aprendizagem desejada, pois amplia as possibilidades de acesso ao conhecimento. A intencionalidade é um fator importante para os OAs, uma vez que estes são recursos digitais com fins educacionais bem definidos. Eles podem assumir diversos formatos, desde imagens e arquivos de texto, até animações e simulações, oportunizando os diferentes tipos de aprendizagem.

Na área pedagógica, a utilização de Objetos de Aprendizagem justifica-se pela lacuna individual que cada estudante apresenta durante o seu desenvolvimento educacional e pelas necessidades de formação inicial e continuada (Tarouco, 2014). Ao desenvolver um OA, um educador deve ter clareza que pode proporcionar aos estudantes formas diversificadas de acesso ao conhecimento, como também pode proporcionar o aprimoramento e a complementação do aprendizado.

### IV. TRABALHOS RELACIONADOS

Vários pesquisadores têm apostado na utilização de Objetos de Aprendizagem como ferramenta para auxiliar o ensino de programação. Sampaio e Souza (2008) apresentam o OA “Lógica Fácil” que tem como objetivo auxiliar os alunos na aquisição de conhecimentos relacionados às três estruturas de controle, a saber, sequencial, condicional e de repetição. O OA possui quatro módulos: teoria, animações, desafio e exercícios. No módulo teoria, são apresentados conteúdos relacionados às estruturas de controle. No módulo de animação, os alunos podem observar passo-a-passo a execução de animações relacionadas às estruturas de controle. No módulo de desafio, o aluno pode escrever um pseudocódigo relacionado a um problema proposto e visualizar sua execução. Por último, no módulo de exercícios são apresentados enunciados de problemas para que o aluno pratique os conceitos adquiridos. Não é possível para o aluno resolver os problemas dentro do OA, o aluno deve utilizar outra ferramenta para escrever seus algoritmos.

Cechinel *et al.* (2008) apresentam OAs desenvolvidos como vídeos explicativos (tutoriais) para auxiliar alunos da disciplina de Algoritmos e Programação. Os vídeos consistem de gravações do *desktop* do computador durante a programação de algum algoritmo ou estrutura de programação na linguagem C, incluindo explicações de como o problema está sendo solucionado.

Saraiva e Netto (2010) apresentam a ferramenta “Monitor” que integra um conjunto de objetos de aprendizagem de quatro tipos: conceitos, animações, propostas de exercícios e quiz. De

acordo com os autores, as animações são separadas em quatro partes. A primeira parte apresenta um código de um algoritmo escrito em pseudocódigo; a segunda parte apresenta uma simulação de valores armazenados e alterados em memória pelo computador de acordo com a execução do algoritmo; a terceira parte simula um console de execução, mostrando a saída gerada pelo algoritmo; e a quarta e última parte apresenta os botões de controle para a visualização da animação. Os exercícios devem ser realizados pelos alunos fora da ferramenta.

Martins *et al.* (2014) apresentam brevemente cinco objetos de aprendizagem para auxiliar a disciplina de Algoritmos e Programação. O primeiro objeto é um jogo que tem como objetivo auxiliar no ensino do conceito de variáveis. O segundo objeto é uma vídeo-aula que apresenta o conceito de operadores lógicos. O terceiro, quarto e quinto são jogos que visam ensinar respectivamente, comando *while* com condição alimentada pelo usuário, estrutura condicional composta e operadores relacionais.

Henrique e Rebouças (2015) apresentam uma revisão sistemática de literatura com o objetivo de identificar OAs que poderiam ser utilizados por professores e alunos para o ensino e a aprendizagem de conceitos relacionados ao paradigma de linguagem de programação orientado a objetos (POO). Os autores indicam que não encontraram nenhum trabalho que mencionasse diretamente o termo “Objeto de Aprendizagem”, mas que algumas ferramentas encontradas poderiam ser categorizadas como OAs e que poderiam ser utilizadas para o ensino de POO.

A proposta desenvolvida pelo Objeto de Aprendizagem CODEX, apresentado neste artigo, se difere dos trabalhos relacionados nos seguintes aspectos: ao contrário dos trabalhos de Sampaio e Souza (2008) e Saraiva e Neto (2010) que envolvem animações de algoritmos e Cechinel *et al.* (2008) que apresentam OAs como tutoriais, o CODEX é um OA gamificado na forma de um tabuleiro, que apresenta desafios aos alunos e permite que os mesmos construam suas soluções utilizando comandos de movimentação. A aprendizagem dos conceitos não é realizada de forma teórica, mas sim por meio do exercício do raciocínio lógico. Apesar de Martins *et al.* (2014) apresentarem OAs como jogos, o CODEX se diferencia destes trabalhos pois apresenta níveis de dificuldade e realiza o monitoramento dos acertos e erros dos estudantes, podendo apresentar para o professor estas informações por meio de estatísticas. Além disso, o CODEX foi desenvolvido em JavaScript, o que amplia seu potencial de uso (abarcando dispositivos móveis).

## V. CODEX – APRENDENDO LAÇOS DE REPETIÇÃO A PARTIR DE OBJETOS DE APRENDIZAGEM

Diante do cenário referente ao ensino de computação e da possibilidade de utilizar os OAs como alternativa pedagógica, foi desenvolvido o CODEX, uma proposta de apoio ao processo de ensino e aprendizagem de programação. O CODEX é um aplicativo gamificado que se utiliza de comandos básicos de orientação para trabalhar os conceitos relacionados aos laços de repetição da programação.

Tendo em vista que o *design* do OA é um recurso capaz de conduzir e facilitar a interação do usuário com o aplicativo, buscou-se na implementação do CODEX desenvolver uma interface que motivasse o estudante para o aprendizado. Dessa forma, o trabalho com uma equipe multidisciplinar tornou-se essencial e, o *layout* do OA foi criado por *designers* especializados no desenvolvimento de interfaces, fator que se acredita tenha agregado qualidade ao projeto.

Na produção do CODEX foi utilizado React e Javascript. O React é um arcabouço usado pelas maiores empresas de *front-end* do mercado (Netflix, Facebook, Instagram), ou *client-side*, e permite o controle do fluxo de renderização e da possibilidade de modularização. Trata-se de uma biblioteca declarativa, eficiente e flexível do JavaScript que é utilizada para construir interfaces de usuários. Ele também permite que desenvolvedores criem aplicações na internet que utilizam dados que podem sofrer alterações com o tempo sem a necessidade de recarregar a página. Seu objetivo é ser rápido, simples e escalonável.

### A. Arquitetura do CODEX

O CODEX utiliza uma arquitetura de aplicações chamada Flux que foi desenvolvida pelo Facebook e é usada para criar aplicativos web. A Flux complementa a apresentação de componentes criados pelo arcabouço React, oferecendo uma camada de dados onde é utilizado um modelo de fluxo unidirecional. As aplicações baseadas em Flux têm quatro componentes principais: *dispatcher*, *stores*, *views* e *actions*.

No fluxo da aplicação CODEX todas as funcionalidades geralmente são encontradas dentro de uma *store* da arquitetura Flux (Fig. 1). A *store* terá o trabalho de informar ao *dispatcher* quais eventos e ações que está aguardando. Quando um evento acontece, o *dispatcher* envia a carga útil para a *store* que está registrada para essa ação específica. Cabe então à *store* a função de atualizar a *view* que será apresentada para o usuário, que por sua vez desencadeia uma ação. A *view* propaga uma *action* através de um *dispatcher* central e isso será enviado para várias *stores*. Essas *stores* contêm uma aplicação de lógica comercial e atualiza todas as visualizações. Isso demonstra que o Flux segue um fluxo de dados unidirecional. A *action*, o *dispatcher*, a *store* e a *view* são nós independentes com entradas e saídas específicas. Os dados fluem através do *dispatcher* e, o *hub* central por sua vez gerencia todos os dados. O *dispatcher* atua como um registro com retorno de chamada ao qual as *stores* respondem. As *stores* permitem emitir uma alteração que será escolhida pelo *controller-views*.



Fig. 1. Fluxograma do Flux

## B. Funcionamento do OA

O funcionamento do CODEX baseia-se em fazer uma bola percorrer um caminho em um tabuleiro que tem seu início sinalizado com um círculo rosa e, o final, com um círculo verde do tabuleiro. Para isso, utilizam-se comandos de movimentação para cima, para baixo, para a esquerda e para a direita, além do comando que determina um laço de repetição. Para concluir o objetivo, o usuário deverá selecionar os comandos que a bola deverá realizar para percorrer o caminho definido no tabuleiro.

A interface proposta para o aplicativo busca trabalhar os conceitos de comando, de sequência e repetição, a partir de uma situação concreta (caminho a ser percorrido pela bola). Desta forma, para definir os passos a serem seguidos pela bola o usuário deve refletir sobre a sequência de instruções a ser realizada, assim como pode visualizar situações em que uma ou mais instruções serão realizadas repetidas vezes (laço de repetição).

Visando subsidiar o professor de dados que possam colaborar para definir sua prática docente no ensino de programação, o CODEX armazena os dados de interação dos usuários, tais como: número de instruções utilizadas para solucionar um problema, número de tentativas de cada usuário, número de respostas corretas e incorretas por nível. Possibilitando ao professor analisar e interpretar a trajetória de cada estudante, identificando dificuldades e potencialidades relacionadas ao conteúdo e, assim, tornando possível definir estratégias que possam contribuir para a aprendizagem do conteúdo.

## C. Utilização do CODEX

Após efetuar o *login*, o usuário será redirecionado para a tela que apresenta os níveis do jogo, sendo possível visualizar os níveis disponíveis. Caso seja o primeiro acesso, apenas o tutorial e o nível 1 estarão desbloqueados.

Os níveis foram definidos baseados na complexidade de solução de cada um dos problemas apresentados ao usuário. Dessa forma, o mesmo irá construindo e consolidando o raciocínio para viabilizar sua chegada à etapa seguinte. O símbolo do cadeado representa que o usuário ainda não conseguiu desbloquear um determinado nível e, para isso, precisará concluir o nível imediatamente anterior para ser possível sua visualização.

O nível denominado Tutorial é de acesso livre em qualquer momento, pois tem por objetivo possibilitar a apropriação da interface do jogo pelo usuário. Este nível contém instruções sobre o objetivo do jogo, as características de cada nível e as instruções sobre comandos e botões do CODEX (Fig. 2).

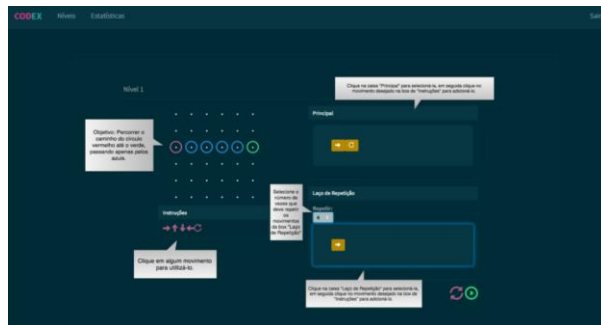


Fig. 2. Tutorial do CODEX

Os demais níveis apresentam a interface dividida em duas colunas, conforme a Fig. 3. Nestes, a coluna da esquerda apresenta o tabuleiro do CODEX e os comandos de movimentação. Já, na coluna da direita, o usuário encontrará a caixa de comandos principais, a caixa onde serão inseridos os comandos referentes ao laço de repetição, o contador de vezes que o laço de repetição será executado, o botão de limpar a tela e o botão de executar a lista de comandos definidos para solução do nível.

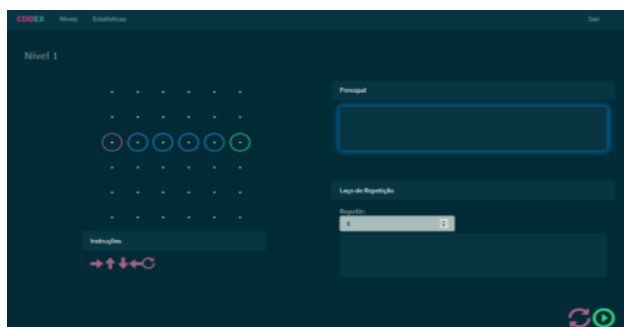


Fig. 3. Interface dos níveis do CODEX

Em se tratando da movimentação sobre o tabuleiro, foram criados comandos baseados em orientações básicas (cima, baixo, esquerda e direita), pois indicam movimentos de fácil compreensão e assimilação. Estes deslocamentos permitem a utilização de um tabuleiro 2D com uma visão de topo, facilitando a jogabilidade. Ao selecionar os comandos que solucionarão o desafio, o usuário deverá clicar no botão de execução para verificar se os comandos escolhidos estão corretos. A execução da lista de comandos definida pelo usuário é sinalizada pelo sistema com um rastro vermelho ou verde (Fig. 4), permitindo ao usuário visualizar o trajeto realizado a partir das suas instruções. Os comandos que representam um caminho correto serão sinalizados em verde, enquanto os incorretos em vermelho.

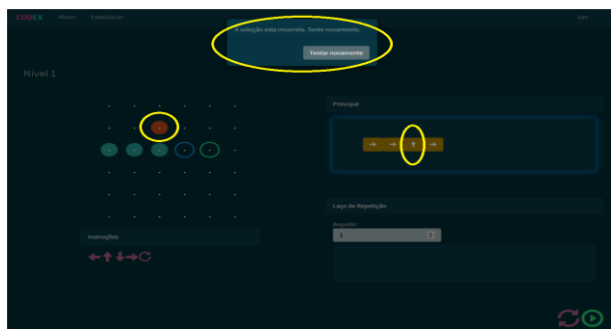


Fig. 4. Sinalização do comando correto (verde) e o incorreto (vermelho)

A possibilidade de visualização da trajetória definida permite ao estudante fazer uma reflexão acerca de sua resposta e transpor para uma situação mais concreta a execução de cada comando. Desta forma, propõe-se um processo de reflexão sobre o erro, visando a construção do conhecimento.

#### D. Visualização de Estatísticas

O registro de estatísticas relacionadas às respostas dos usuários representa uma importante funcionalidade do CODEX. O acesso às estatísticas ocorre por meio do menu disponível na aba superior do aplicativo e sua visualização se dá a partir de duas seções de gráficos (Fig. 5).



Fig. 5. Estatísticas registradas pelo CODEX

O gráfico anterior agrupa o quantitativo de tentativas corretas e incorretas de cada nível, permitindo ao docente analisar e avaliar o rendimento de seus alunos em cada nível e elaborar alternativas para apoiar seu aprendizado. Os gráficos inferiores estão divididos por nível. Nestes gráficos é possível quantificar os usuários que solucionaram cada nível com o mesmo número de instruções, possibilitando inferir e analisar as soluções apresentadas pela quantidade de movimentos e determinar se foi utilizado algum comando na caixa de laço de repetição.

## VI. CONCLUSÃO

O CODEX foi elaborado baseado nas dificuldades vivenciadas por estudantes de cursos de Computação de forma que, em classe ou extraclasse, o aluno possa revisar, compreender e manter o conhecimento previamente estudado, ou até mesmo, iniciar o processo de ensino e aprendizagem de forma diferenciada. Dado que o software trabalha com o conceito de laços de repetição desvinculado de uma linguagem de programação, o estudante é capaz de abstrair e compreender

o funcionamento de uma repetição. Dessa forma, será capaz de utilizar o conhecimento adquirido em qualquer situação e linguagem que venha a utilizar.

O uso de tecnologias pode permitir um ambiente mais propício ao processo de ensino e aprendizagem e viabilizar ao estudante situações que não são possíveis em ambientes reais. Nesse sentido, entende-se que o objetivo principal deste trabalho foi alcançado, pois o CODEX oferece uma apoio ao ensino de programação. A elaboração e a utilização de recursos digitais que desenvolvem conteúdos de forma dinâmica, atrativa e lúdica podem proporcionar práticas pedagógicas que vão ao encontro das necessidades ou dificuldades dos estudantes e dos professores.

Com o intuito de aprimorar e aumentar a contribuição do software projetado, foi considerado que ainda é possível acrescentar novos recursos e ferramentas com foco no desenvolvimento dos alunos. Para isso, como trabalhos futuros, são sugeridas as seguintes implementações:

- módulo de acesso ao professor para que ele tenha autonomia de criar novas fases e desafios, permitindo assim explorar pontos e conceitos específicos conforme as características de um ou mais alunos;
- implementação de *learning analytics* visando identificar como os alunos se comportam nos ambientes virtuais de aprendizagem, para assim reorientar os caminhos de aprendizagem;
- autenticação via rede social que facilitará o acesso e o registro e permitirá ao aluno compartilhar seu desenvolvimento no CODEX com outros usuários, além de tornar sua experiência mais interativa e lúdica.

## REFERÊNCIAS

- [1] CODE.ORG. Promote Computer Science. Disponível em: <https://code.org/promote>. Acessado em: Abril, 2018.
- [2] CHECINEL, C.; SILVEIRA, A. V.; SILVEIRA, R. S.; NUNES, E. M.; MOREIRA, A. A.; COGO, G. B.; BETEMPS, C. M.; TAVARES, R. N. “Desenvolvimento de Objetos de Aprendizagem para o Apoio à Disciplina de Algoritmos e Programação”. Em XIX Simpósio Brasileiro de Informática na Educação, 2008. Disponível em: [http://sbie2008.virtual.ufc.br/CD\\_ROM\\_COMPLETO/workshops/works hop%202/Desenvolvimento%20de%20Objetos%20de%20Aprendizagem%20para%20o%20Apoio.pdf](http://sbie2008.virtual.ufc.br/CD_ROM_COMPLETO/workshops/works hop%202/Desenvolvimento%20de%20Objetos%20de%20Aprendizagem%20para%20o%20Apoio.pdf). Acessado em: Maio, 2018.
- [3] GIRAFFA, L. M. M.; MÜLLER, L.; MORAES, M. C. “Ensino Programação apoiada por um ambiente virtual e exercícios associados a cotidiano dos alunos: compartilhando alternativas e lições aprendidas.” Anais do I Workshop de Ensino em Pensamento Computacional, Algoritmos e Programação, 2015. Disponível em: <http://www.br-ie.org/pub/index.php/wcbie/article/view/6303>. Acessado em: Abril, 2018.
- [4] HENRIQUE, M. S.; REBOUÇAS, A. D. D. S. “Objetos de Aprendizagem para auxiliar o ensino de conceitos do Paradigma de Programação Orientada a Objetos”. Revista Novas Tecnologias na Educação, v. 13, n.2, 2015.
- [5] KOPER, R. “Combining reusable learning resources and services to pedagogical purposeful units of learning”. Journal of Interactive Media in Education, ISSN: 1365-893x, 2003.
- [6] LIMA JÚNIOR, José Augusto Teixeira et al. “Dificuldades no processo de aprendizagem de algoritmos: uma análise dos resultados na disciplina

- de AL1 do curso de sistemas de informação da Faeterj–Campus Paracambi”. Cadernos UniFOA, (n.27), 2015.
- [7] MARTINS, G.; SCHMITD, A. E.; AMARAL, A. “Objetos de Aprendizagem Digitais voltados para o Ensino-Aprendizagem de Algoritmos e Programação”. Mostra Nacional de Iniciação Científica e Tecnológica Interdisciplinar, 2014. Disponível em: <http://eventos.ifc.edu.br/wp-content/uploads/sites/5/2014/08/OBJETOS-DE-APRENDIZAGEM-DIGITAIS-VOLTADOS-PARA-O-ENSINO-APRENDIZADO-DE-ALGORITMOS-DE-PROGRAMACAO.pdf>. Acessado em: Maio, 2018.
- [8] PEREIRA JÚNIOR, J. C. R.; RAPKIEWICZ, C. E. “O processo de ensino-aprendizagem de fundamentos de programação: uma visão crítica da pesquisa no Brasil”. Anais do XII Workshop sobre Educação em Computação (SBC), 2004.
- [9] RAABE, A. L. A.; SILVA, J. M. C. “Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos”. Em XIII Workshop de Educação em Computação (WEI’2005). São Leopoldo, RS, Brasil, 2005.
- [10] RAABE, A. L. A.; ZORZO, A. F.; FRANGO, I.; RIBEIRO, L.; GRANVILLE, L. Z.; SALGADO, L.; DA CRUZ, K. M. J.; BIGOLIN, N.; CAVALHEIRO, S. A. C.; FORTES, S. Referenciais de Formação em Computação: Educação Básica. 2017. Disponível em: <http://www.sbc.org.br/files/ComputacaoEducacaoBasica-versaofinal-julho2017.pdf>. Acessado em: Abril, 2018.
- [11] SAMPAIO, R. L.; SOUZA, A. C. S. “Uma experiência com objetos digitais de aprendizagem no ensino da Lógica de Programação.” Em IX Congresso Iberoamericano de Informática na Educação. Disponível em: [http://www.ufrgs.br/niee/eventos/RIBIE/2008/pdf/experiencia\\_objetos\\_br.pdf](http://www.ufrgs.br/niee/eventos/RIBIE/2008/pdf/experiencia_objetos_br.pdf). Acessado em: Maio, 2018.
- [12] SARAIVA, I. B.; NETTO, C.M. “Monitor: um conjunto de objetos de aprendizagem para apoio ao ensino de programação de computadores”. Em XVIII Workshop sobre Educação em Informática, 2010. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/wei/2010/0019.pdf>. Acessado em: Maio, 2018.
- [13] SMITH, Megan. Computer Science for All. 2016. Disponível em: <https://obamawhitehouse.archives.gov/blog/2016/01/30/computer-science-all>. Acessado em: Abril, 2018.
- [14] SOUZA, C. M. “VisuAlg - ferramenta de apoio ao ensino de programação”. Revista Eletrônica TECCEN, v.2(n.2), 2016.
- [15] SOUZA, D. et al. “Problemas e dificuldades no ensino e na aprendizagem de programação: Um mapeamento sistemático”. Revista Brasileira de Informática na Educação, v.24 (n.1), 2016.
- [16] TAROUÇO, Liane Margarida Rockenbach et al. “Objetos de Aprendizagem: teoria e prática”. 2014. ISBN 978-85-7727-643-1.