

# BrC: Proposta de uma Biblioteca em Português para Ensino de Programação em Linguagem C

Luciana Rita Guedes<sup>1</sup>, Aleksander Sade Paterno<sup>2</sup>

<sup>1</sup>Departamento de Ciência da Computação – <sup>2</sup>Departamento de Engenharia Elétrica - Universidade do Estado de Santa Catarina (UDESC) - Centro de Ciências Tecnológicas  
Rua Prof. Paulo Malschitzki, 200 – 89.219-710 – Joinville – SC – Brasil

[luciana.guedes,aleksander.paterno]@udesc.br

**Abstract.** *This paper presents the proposal of a C language compatible library that replaces reserved words of this programming language with equivalent words in Portuguese and can be applied in different integrated development environments (IDE). In addition, the basic input and output functions have their syntax simplified. The aim is to facilitate the learning of native Portuguese language programming beginners and to allow a smooth transition to the use of the C language. The library is easy to use and publicly available and no additional software is required for its application, besides the environment already used. The library was successfully used in beginner classes with undergraduate students from different courses for six semesters.*

**Resumo.** *Este artigo apresenta a proposta de uma biblioteca compatível com a linguagem C que substitui palavras reservadas desta linguagem por vocábulos equivalentes em português e pode ser aplicada em diferentes ambientes de desenvolvimento. Adicionalmente, as funções básicas de entrada e saída têm a sua sintaxe simplificada. O objetivo é facilitar a aprendizagem de iniciantes em programação nativos da língua portuguesa e permitir uma transição suave para o uso da linguagem C. A biblioteca é de fácil utilização e está publicamente disponível, não sendo necessária a instalação de nenhum software adicional para sua aplicação, além do próprio ambiente já utilizado. A biblioteca foi utilizada com êxito em turmas iniciantes com estudantes universitários de diferentes cursos durante seis semestres.*

## 1. Introdução

Estudantes iniciantes de programação costumam apresentar dificuldades no entendimento da lógica do computador, independentemente da linguagem de programação escolhida. O aprendizado de programação de computadores é considerado uma tarefa não simples e não trivial [Vahldick et al. 2015, Scaico et al. 2012]. Diversas propostas para melhorar o ensino de programação têm sido elaboradas e testadas nos últimos anos. Algumas enfatizam o uso de ferramentas [Scratch, Codecademy, Code.org 2018][Resnick et al. 2009], outras tem foco na metodologia [de Jesus, Raabe 2009][Denton et al. 2005]. Do ponto de vista da linguagem de programação escolhida, é fato que algumas apresentam estruturas sintáticas mais complexas que outras. A linguagem C (ANSI) possui uma sintaxe mais complexa, com mais elementos e maior rigor estrutural que outras linguagens conhecidas como Pascal e Python, por exemplo.

No entanto, a linguagem C costuma ser a escolha de muitos cursos superiores como linguagem inicial, por suas características de aplicações e robustez.

Muitos estudos têm sido realizados, em todo o mundo, especialmente nos últimos anos com objetivo de aprimorar o ensino de algoritmos e programação de computadores, tanto no aspecto metodológico como no desenvolvimento e utilização de ferramentas [Valhdick 2014, Zanchet et al. 2017, Laporte; Zaman 2017].

No cenário brasileiro, esta também é uma temática recorrente nos principais eventos e periódicos da área de Computação. No caso do Brasil, outra questão considerada é que, embora fosse altamente desejável que os ingressantes de cursos superiores já tivessem um domínio da língua inglesa, ao menos em termos de comunicação básica, a realidade encontrada nas universidades não está de acordo com este requisito [Anido 2015, Noschang 2014]. Um estudo mais recente sobre o índice de proficiência em inglês [EPI 2017] indica que o Brasil teve um pequeno decréscimo entre 2016 e 2017, caindo da 40<sup>a</sup> para a 41<sup>a</sup> posição, entre 80 países pesquisados. O mesmo estudo indica, ainda, que estudantes universitários de Engenharia e Ciências (exatas) são os que têm os piores índices de proficiência. Em contraste com estes dados, o ensino de programação de computadores costuma ser aplicado justamente nestes cursos.

Para amenizar as dificuldades com a língua inglesa, alguns professores optam pelo uso de pseudocódigo como Portugol ou outras estratégias como desenvolvimento de fluxogramas ou outros diagramas similares. Algumas destas abordagens exigem que o estudante crie seus algoritmos em papel sem visualizar os resultados de execução de seus códigos. Outros casos usam ferramentas cuja sintaxe difere da linguagem de programação que será utilizada, duplicando o esforço de aprendizagem do estudante [Noschang 2014].

Neste artigo será apresentada uma biblioteca que permite a escrita do código-fonte usando a mesma sintaxe da linguagem de programação C e o mesmo ambiente de desenvolvimento já escolhido pelo professor, porém usando palavras em língua portuguesa. A seção 2 apresenta trabalhos relacionados e a seção 3 traz o detalhamento sobre o desenvolvimento da biblioteca. Por fim, os resultados alcançados com o experimento são apresentados na seção 4.

## **2. Trabalhos correlatos**

O ensino de programação para iniciantes tem sido amplamente discutido no meio científico. Alguns trabalhos de revisão de literatura têm comprovado a quantidade de publicações desta área. Vihavainen et al. (2014) apresentam uma revisão sistemática da literatura (RSL) feita nas bases de dados da ACM e da IEEE, com foco nos índices de reprovação em cursos de programação para iniciantes. De modo a fazer uma análise comparativa com estes resultados, Ramos et al. (2015) também realizaram uma RSL, porém com foco regional, considerando apenas publicações no Brasil, confrontando os dados com aquele primeiro estudo. O primeiro trabalho resultou em mais de 1000 artigos e, no segundo caso, a busca retornou 394 artigos sobre o tema.

Com enfoque mais voltado ao ensino de programação para estudantes da educação básica, Silva (2017) obteve um total de 914 artigos em sua revisão bibliográfica, também restrita a eventos e periódicos brasileiros, considerando apenas a trabalhos publicados num período de cinco anos. Com relação a iniciativas de uso de

ferramentas que auxiliam no ensino, o trabalho de revisão da literatura de Aureliano e Tedesco (2012), com foco em publicações feitas no Brasil, concluiu que cerca de 65% dos estudos estão voltados a estas ferramentas. Entre as ferramentas de apoio ao ensino de programação para iniciantes no Brasil, pode-se destacar o trabalho de Leite et al. (2013) que apresenta um estudo de caso do VisuAlg. A ferramenta permite a escrita de pseudocódigos em português usando a sintaxe da linguagem Pascal. Petry e Rosatelli (2006) apresentam a ferramenta AlgoLC, que também permite a construção de algoritmos em língua portuguesa, com sintaxe que também assemelha-se à linguagem Pascal. Blatt et al. (2017) realizaram um mapeamento sistemático da literatura em busca de metodologias e ferramentas utilizadas no ensino algoritmos e programação e destacaram o uso do Scratch em mais de 40% dos artigos pesquisados. O Scratch [Maloney et al. 2008] é uma ferramenta visual que utiliza uma técnica denominada programação por blocos. Tem também uma abordagem lúdica que usa personagens e desenvolvimento de jogos como recursos de ensino e aprendizagem e está disponível em português. Noschang et al. (2014) apresentaram o Portugol Studio, um ambiente integrado de desenvolvimento que utiliza comandos escritos em língua portuguesa para o desenvolvimento de algoritmos computacionais. O trabalho de Anido (2015) apresenta o ambiente SACI, desenvolvido para o ensino de algoritmos voltado aos competidores da Olimpíada Brasileira de Informática. O SACI tem toda interface em língua portuguesa, mas mantém o código-fonte na linguagem de programação escolhida (JavaScript ou Python).

Muitos outros estudos apresentam propostas de ferramentas, muitas delas em língua portuguesa, demonstrando um esforço da comunidade científica na busca de soluções que atendam as especificidades dos estudantes nativos desta língua.

### 3. Metodologia de criação da biblioteca BrC

Para criação da biblioteca BrC foi feito, inicialmente, o levantamento das palavras reservadas da linguagem C que, segundo o padrão ANSI [Kernighan e Ritchie 1987] são um total de 32 palavras, conforme mostradas no Quadro 1.

**Quadro 1. Palavras reservadas da linguagem C (padrão ANSI). Fonte: [Kernighan e Ritchie 1987]**

1. auto	9. double	17. int	25. struct
2. break	10. else	18. long	26. switch
3. case	11. enum	19. register	27. typedef
4. char	12. extern	20. return	28. union
5. const	13. float	21. short	29. unsigned
6. continue	14. for	22. signed	30. void
7. default	15. goto	23. sizeof	31. volatile
8. do	16. if	24. static	32. while

Em seguida, foi feita uma tradução de cada uma destas palavras para equivalentes no idioma português. Por questão de organização, as 32 palavras reservadas foram divididas em quatro blocos distintos: 1) tipos de variáveis; 2) comandos de sistema; 3) comandos de seleção e 4) comandos de iteração. Ressalta-se que esta divisão não era obrigatória, sendo apenas uma decisão de implementação, sem nenhum impacto no resultado final. Após isso, foi criado um arquivo de extensão `.h` (header ou biblioteca) que, com o uso da diretiva de compilação `#define` permitiu “redefinir” as palavras para que fossem interpretadas pelo compilador na sua versão em português no lugar do original em inglês. Esta biblioteca foi nomeada `BrC`, como referência à linguagem C escrita na língua do Brasil, o idioma português (que poderia ser chamada de *Brazilian C*, ou *C “brasileiro”*).

Além das palavras reservadas da linguagem C, outros elementos foram incorporados à biblioteca. Inicialmente, no cabeçalho desta biblioteca, foram adicionadas diretivas do tipo `#include` para incluir as bibliotecas mais comumente utilizadas que são `stdio.h` e `stdlib.h`. Esta alteração foi feita para simplificar o código do estudante iniciante de modo que ele só tenha que incluir a própria biblioteca `brc.h` em seu programa, sem preocupar-se com outras bibliotecas básicas.

Outra alteração implementada foi a modificação das funções básicas de entrada e saída (leitura e escrita de dados). Aqui cabem algumas considerações, pois a observação do processo de aprendizagem de linguagens de programação, especialmente das funções de entrada e saída, tem demonstrado que os estudantes costumam ter dificuldades de compreensão sobre o significado destas funções. A função de entrada de dados via teclado representa, ao mesmo tempo, uma pausa na execução do código para aguardar a interação do usuário e também um processo de armazenamento na memória, que guarda o dado digitado. Não bastasse esta complexidade inerente à função, a sintaxe de `scanf`, na linguagem C, possui dois parâmetros, um dos quais com uso de ponteiros, como em `scanf(“%d”,&x)`. A biblioteca do `BrC` possui palavras para representar a entrada e saída de dados relativas a `scanf` e `printf` que são, respectivamente, `leiaC` e `escrevaC`. Neste caso, a sintaxe é rigorosamente idêntica às funções originais em C. No entanto, com o objetivo de simplificar o entendimento da entrada e saída de dados, optou-se por tomar emprestados os comandos da linguagem C++, `cin` e `cout`. Isto tornou possível criar as funções `leia` e `escreva` que possuem sintaxe simplificada e facilitam o entendimento. Isto evita que o iniciante em programação tenha que preocupar-se com os detalhes da sintaxe quando ainda está compreendendo o processo envolvido na função, seja ela de entrada ou saída de dados.

Finalmente, foram incorporadas mais duas funções à lista já existente: a função `main` foi redefinida com a palavra “principal” e a função `system` foi redefinida com a palavra “sistema”. Mais uma vez, foi apenas uma decisão de implementação, apenas para evitar o uso de palavras em inglês no código fonte do estudante, em consonância com a ideia do `BrC`.

Após estes procedimentos, a biblioteca `BrC` ficou conforme aparece no Quadro 2, que apresenta o arquivo `brc.h` integralmente, da forma como foi criado e está disponível publicamente no endereço <<https://goo.gl/jB8hTh>>.

Para usar a biblioteca `brc.h`, é necessário apenas copiá-la para a pasta onde encontram-se as outras bibliotecas já utilizadas pelo compilador. Por exemplo, no caso do uso do `gcc`, há uma pasta denominada `include` onde já existem as demais bibliotecas-

padrão usadas pelo compilador. Basta encontrar esta pasta e copiar o arquivo **brc.h** para dentro dela. No sistema operacional Windows, a biblioteca foi usada tanto no ambiente de desenvolvimento Dev-C++ como no CodeBlocks, pois ambos utilizam o gcc como compilador. No sistema operacional Linux, a biblioteca foi usada com o ambiente Geany.

Após a inserção da biblioteca na pasta correta, pode-se criar os programas usando-se o ambiente de desenvolvimento escolhido e substituindo as palavras da linguagem C pelas equivalentes na biblioteca brc.h. O Quadro 3 apresenta o exemplo de um programa em C e seu equivalente em BrC.

**Quadro 2. Conteúdo da biblioteca brc.h**

<pre>//Cabeçalhos #include &lt;stdlib.h&gt; #include &lt;stdio.h&gt; #include &lt;stdarg.h&gt; #include &lt;iostream&gt; using namespace std;  //Tipos de variáveis #define local auto #define caractere char #define constante const #define dobro double #define enumerador enum #define externo extern #define real float #define inteiro int #define longo long #define registrador register #define curto short #define sinalizado signed #define tamanhode sizeof #define estatico static #define estrutura struct #define tipodef typedef #define naosinalizado unsigned #define uniao union #define vazio void #define volatil volatile</pre>	<pre>//Comandos de sistema #define retorne return #define quebre break #define continua continue #define padrao default  //Comandos de seleção #define se if #define senao else #define escolha switch #define caso case  //Comandos de iteração #define faca do #define enquanto while #define para for #define vapura goto  //Funções de entrada e saída #define leiaC scanf #define escrevaC printf #define escreva(V valor) cout &lt;&lt; VALOR #define leia(V valor) cin &gt;&gt; VALOR  //Outras funções #define principal main #define sistema system</pre>
---	--

É importante observar que toda sintaxe da linguagem permanece inalterada (uso de chaves, ponto-e-vírgula, parêntesis, etc.), bem como os operadores relacionais (`==`, `>`, `<`, `>=`, `<=`, etc.). Os operadores lógicos (`&&`, `||`, `!`) também foram mantidos sem alteração. O objetivo é familiarizar o estudante com a sintaxe da linguagem, trocando apenas as palavras escritas em inglês.

Deste modo, a biblioteca criada representa uma tentativa de mudança de modalidade simbólica para compreensão de comandos e seu suporte sensorial às modalidades de comunicação que os alunos já conhecem, de acordo com a experiência dos docentes.

**Quadro 3. Exemplo de programa em BrC e seu equivalente em C.**

```
// programa escrito em BrC
#include <brc.h>

principal(){

    real a,b;
    inteiro c;

    faca{
        escreva("Digite um valor:");
        leia(a);
        escreva("Digite outro valor:");
        leia(b);

        se(a==b){
            escreva("Valores iguais.\n");
        }
        senao{
            escreva("Valores diferentes.\n");
        }

        escreva("Digite 1 para repetir, ");
        escreva("outro valor para sair:");
        leia(c);
    } enquanto(c==1);
    retorne 0;
}
```

```
// programa equivalente escrito em C
#include <stdio.h>

main(){

    float a,b;
    int c;

    do{
        printf("Digite um valor:");
        scanf("%f",&a);
        printf("Digite outro valor:");
        scanf("%f",&b);

        if(a==b){
            printf("Valores iguais.\n");
        }
        else{
            printf("Valores diferentes.\n");
        }

        printf("Digite 1 para repetir, ");
        printf("outro valor para sair:");
        scanf("%i",&c);
    } while(c==1);
    return 0;
}
```

Outra observação diz respeito ao uso das funções **printf** e **scanf**, que aparecem na versão em C do Quadro 3. Conforme já mencionado, para simplificar o entendimento do estudante, na biblioteca `brc.h` os comandos **escreva** e **leia** equivalem, respectivamente, a **cout** e **cin**, emprestados da linguagem C++. No entanto, numa aula de linguagem C, os comandos usados para entrada e saída de dados costumam ser aqueles mencionados inicialmente (`scanf` e `printf`). Por este motivo, eles foram descritos como equivalentes aos comandos **escreva** e **leia** no exemplo apresentado.

#### 4. Experimento e Resultados

O uso da biblioteca foi aplicado em diferentes turmas durante seis semestres, entre 2014 e 2017. As turmas eram mistas, compostas de estudantes de diferentes cursos de Engenharia e de Licenciatura. Os estudantes de engenharia eram de cursos como Civil, Elétrica e de Produção e o licenciandos eram de Física ou Matemática. Os estudantes eram, em sua grande maioria, iniciantes, ou seja, não tinham experiência alguma com linguagem de programação ou algoritmos computacionais. Isto foi verificado em entrevista realizada no primeiro dia de aula, como prática de apresentação conduzida e registrada pelo docente em todas as turmas. A disciplina de Algoritmos e Linguagem de Programação é ministrada com o objetivo do aprendizado da linguagem C para aplicação em resoluções de problemas em diversas áreas do conhecimento. A opção pela linguagem C é uma decisão dos colegiados dos cursos em acordo com o departamento de Computação, que oferece a disciplina na Universidade. As turmas tinham, em média, 20 estudantes e as aulas eram dadas em laboratório de informática com um equipamento para cada estudante. A disciplina possui uma carga horária total de 72 horas/aula e é distribuída em 4 aulas semanais, sendo 2 aulas teóricas e 2 práticas.

O uso da biblioteca BrC foi realizado a partir da segunda semana de aula, quando são iniciados os primeiros algoritmos computacionais, após as aulas introdutórias de apresentação da disciplina e funcionamento básico do hardware. O

grande diferencial ocorre nestas primeiras aulas, pois os estudantes já podem observar seus programas em execução no ambiente de desenvolvimento escolhido pelo professor. Eles foram incentivados a copiarem a biblioteca para seus computadores pessoais para executarem as tarefas extra-classe que servem de fixação dos conteúdos.

Embora existam várias outras iniciativas de uso de ferramentas de pseudocódigo ou similares [Petry e Rosatelli 2006, Leite et al. 2013, Noschang et al. 2014] que permitam a observação do programa em execução, o uso do BrC permite que o estudante escreva seu código utilizando a sintaxe da linguagem desde os primeiros algoritmos (uso de ponto-e-vírgula, chaves, parêntesis, etc.) e já inicie no ambiente de desenvolvimento onde ele usará a linguagem C posteriormente.

Nas turmas onde o BrC foi aplicado, o uso prosseguiu até aproximadamente metade da carga horária da disciplina, quando os estudantes já haviam passado pelas estruturas básicas de programação como sequência, seleção e iteração. Após esta fase onde os comandos básicos e, principalmente, a lógica de programação básica já estava internalizada, foi feita a transição para a linguagem C. Esta transição também caracteriza uma vantagem do BrC pois trata-se basicamente da tradução das palavras para a linguagem C. A única diferença ocorre na apresentação dos comandos *scanf* e *printf*, que têm sua sintaxe apresentada pela primeira vez, já que a biblioteca utiliza uma simplificação dos comandos de entrada e saída. Vale ressaltar que isto ocorre quando as dificuldades iniciais já foram ultrapassadas e há uma vivência prática já adquirida do uso destes comandos com a biblioteca do BrC.

Não foram realizadas métricas de desempenho por nota ou medidas de satisfação dos estudantes nas turmas onde a biblioteca foi aplicada. No entanto, os resultados observados foram: 1) possibilidade de execução do código desde as aulas iniciais: em contraponto com os métodos que usam algoritmos feitos com base exclusivamente em lápis e papel, a observação da execução do programa desde as primeiras aulas traz ao estudante uma experiência de aprendizagem mais concreta; 2) o uso do mesmo ambiente de desenvolvimento durante toda a disciplina: em todas as turmas aplicadas, o ambiente de desenvolvimento (IDE) escolhido para o BrC foi usado igualmente após a mudança para o uso da linguagem C, em contraste com as práticas que utilizam um ambiente na fase de introdução aos algoritmos e outro ainda na fase em que a linguagem C passa a ser integralmente utilizada pelos alunos e 3) a transição suave para a linguagem de programação C: a atividade docente de apresentação dos comandos da linguagem C anteriormente necessitava de duas aulas de 50 minutos para ser concluída e, após a prática de uso do BrC, passou a tomar apenas um quarto (1/4) deste tempo, ou seja, aproximadamente 25 a 30 minutos, viabilizando mais tempo em aula para os estudantes fazerem exercícios e tirarem eventuais dúvidas diretamente com o professor.

#### **4. Discussão e Conclusão**

Este trabalho apresentou o desenvolvimento e aplicação de uma biblioteca compatível com a linguagem C que permite substituir palavras reservadas da linguagem para equivalentes em língua portuguesa. A biblioteca foi construída de modo bem simples e pode ser usada com os principais ambientes de desenvolvimento já existentes para a linguagem C.

A aplicação nas turmas apresentou características observadas como positivas sob o ponto de vista do professor, quando permitiu diminuir o tempo de aula expositiva em

alguns momentos e, sob o ponto de vista dos estudantes, trouxe algumas características que podem facilitar o processo de aprendizagem e de familiarização com a linguagem de programação C.

O uso do BrC é uma estratégia inspirada na necessidade de o discente ter alguma forma de suporte menos abstrato para auxiliá-lo no desenvolvimento da capacidade de abstração requerida pela linguagem C ordinária. Dessa forma, o BrC mantém o conteúdo da disciplina num nível inicial mais acessível ao aluno até que ocorra a transição quando não se utilizará mais a biblioteca de tradução.

Como os experimentos realizados não fizeram um estudo comparativo de desempenho dos estudantes com o uso da biblioteca em relação a outras metodologias tradicionais, sugere-se que trabalhos futuros possam buscar este direcionamento de pesquisa.

Por fim, a proposta aqui apresentada não pretende ser indicada como solução final para os inúmeros problemas do ensino de programação da linguagem C, mas sim indicar uma alternativa simples que pode ser facilmente replicada em contextos similares.

## **Agradecimentos**

Registram-se agradecimentos ao estudante Carlos Henrique da Costa, monitor da disciplina de Algoritmos, e ao Prof. Dr. Rui Jorge Tramontin Junior, professor do Departamento de Ciência da Computação, pela contribuição dada na realização deste estudo. Este trabalho está vinculado ao projeto de pesquisa "Novas abordagens para o ensino de lógica de programação: estudo, análise e uso das estratégias didático-pedagógicas e dos recursos tecnológicos" do Grupo de Pesquisa em Informática na Educação (GPIE) da UDESC.

## **Referências**

- Anido, R. (2015). "SACI – ainda outro ambiente para o ensino de programação". In: *Anais do Workshop sobre Educação em Computação (WEI)* .
- Aureliano, V. C. O., & Tedesco, P. C. D. A. R. (2012). "Ensino-aprendizagem de Programação para Iniciantes: uma Revisão Sistemática da Literatura focada no SBIE e WIE". In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 23, No. 1).
- Blatt, L., Becker, V., & Ferreira, A. (2017). "Mapeamento Sistemático sobre Metodologias e Ferramentas de apoio para o Ensino de Programação". In *Anais do Workshop de Informática na Escola* (Vol. 23, No. 1, p. 815).
- CODECADEMY (2018). Codecademy: aprenda a programar. Website. Disponível em: <<http://www.codecademy.com/>>. Acesso em: 16/03/2018.
- CODE.ORG (2018). Code.org: comece a aprender. Website. Disponível em: <<http://www.code.org/>>. Acesso em: 16/03/2018.
- Denton, L. F., McKinney, D., & Doran, M. V. (2005, October). A melding of educational strategies to enhance the introductory programming course. In *Frontiers in Education, 2005. FIE'05. Proceedings 35th Annual Conference* (pp. F4G-7). IEEE.



- EPI English Proficiency Index (2017). "Índice de Proficiência em Inglês da EF para escolas". Relatório anual da *Education First (EF) English Proficiency Index*. Disponível em <<https://www.ef.com/~media/centralefcom/epi/downloads/epi-s/2017/ef-epi-s-2017-portuguese.pdf>>. Acesso em 02/04/2018.
- Hinterholz, O. (2009). "Tepequém: uma nova Ferramenta para o Ensino de Algoritmos nos Cursos Superiores em Computação". In *XVII-Anais do Workshop sobre Educação em Informática* (Vol. 20, p. 21).
- de Jesus, E. A., & Raabe, A. L. A. (2009, November). Interpretações da TAXONOMIA de Bloom no contexto da Programação Introdutória. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 1, No. 1).
- Kernighan, B. W., & Ritchie, D. M. (1987). "*C a Linguagem de Programação*". Campus.
- Laporte, L., & Zaman, B. (2017). "A comparative analysis of programming games, looking through the lens of an instructional design model and a game attributes taxonomy". *Entertainment Computing*.
- Leite, V. M., Senefonte, H. C., & Seabra, R. D (2013). "VisuAlg: Estudo de Caso e Análise de Compilador destinado ao ensino de Programação". In *Nuevas Ideas en Informática Educativa TISE 2013*. p. 637-640.
- Maloney, J., Peppler, K., Kafai, B., Y., Resnick, M., , and Rusk, N. (2008). "Programming by choice: urban youth learning programming with Scratch". *ACM SIGCSE Bulletin*, 40(1):367–371.
- Noschang, L. F., Pelz, F., & Raabe, A. (2014). "Portugol Studio: Uma IDE para iniciantes em programação". *Anais do CSBC/WEI*, 535-545.
- Petry, P. G., & Rosatelli, M. C. (2006). Ensino e Aprendizagem de Algoritmos com o AlgoLC. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 1, No. 1, pp. 408-417).
- Ramos, V., Wazlawick, R., Galimberti, M., Freitas, M., & Mariani, A. C. (2015). "A Comparação da Realidade Mundial do Ensino de Programação para Iniciantes com a Realidade Nacional: Revisão sistemática da literatura em eventos brasileiros". In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)* (Vol. 26, No. 1, p. 318).
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., & Kafai, Y. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11), 60-67.
- Saico, P., Lopes, D., Silva, M. D. A., Silva, J. D., Neto, S. V. M., & Falcão, E. D. (2012). "Implementação de um Jogo Sério para o Ensino de Programação para Alunos do Ensino Médio Baseado em m-learning". In *Anais do XX Workshop sobre Educação em Computação*. Curitiba: PR.
- SCRATCH (2018). Scratch: crie histórias, jogos e animações. Website. Disponível em: <<http://www.scratch.mit.edu/>>. Acesso em: 16/03/2018.

- Silva, J. C. D. (2017). Ensino de Programação para alunos do Ensino Básico: Um levantamento das pesquisas realizadas no Brasil. Trabalho de Conclusão de Curso. Universidade Federal da Paraíba.
- Vahldick, A., Mendes, A. J., & Marcelino, M. J. (2014). "A review of games designed to improve introductory computer programming competencies". In *Frontiers in Education Conference (FIE)*, 2014 IEEE (pp. 1-7). IEEE.
- Vahldick, A., Mendes, A. J., Marcelino, M. J., Hogenn, M., & Schoeffel, P. (2015). Testando a diversão em um jogo sério para o aprendizado introdutório de programação. 23º WEI-Workhop Sobre Educação Em Computação.
- Vihavainen, A., Airaksinen, J., & Watson, C. (2014). "A systematic review of approaches for teaching introductory programming and their influence on success". In *Proceedings of the tenth annual conference on International computing education research* (pp. 19-26). ACM.
- Zanchett, G. A., Vahldick, A., & Raabe, A. (2017). "Games for Programming as an Approach for First Programming Experiences". *International Journal on Computational Thinking (IJCThink)*, 1(1), 39.