

Desenvolvimento de Aplicativo para Ensino de Estruturas de Repetição na Disciplina de Algoritmos

Cleverson S. dos Anjos, Mariana Cequinel, Rodrigo Duda, Tiago Gerke, Valter Luís E. Junior

Instituto Federal do Paraná – Câmpus Irati (IFPR)
Rua Pedro Koppe, 100 - 84500-000 - Irati – PR – Brasil

{cleverson.anjos, rodrigo.duda, tiago.gerke,
valter.junior}@ifpr.edu.br, marianacequinell15@gmail.com

Abstract. *In this paper it is discussed the development of a tool to support the learning of loops within the Algorithms subject that aims to facilitate the visualization of elements that together determine the functioning of this structure. The RepitaApp app was developed after analysis of various loop structures, the elements that compose it and how they relate were identified. Through answers given by the user to questions asked by the app, it generates such structures. In preliminary tests with students the app's potential became evident regarding the demonstration of the elements necessary for the operation of the structure. It is expected that the use of RepitaApp minimizes the difficulties of students in this subject.*

Resumo. *Neste trabalho é abordado o desenvolvimento de uma ferramenta para o apoio à aprendizagem da matéria de estruturas de repetição dentro da disciplina de Algoritmos visando facilitar a visualização dos elementos que, em conjunto, determinam o funcionamento dessa estrutura. Após análise de diversas estruturas, foram identificados os elementos que a compõe e como eles se relacionam, em seguida, desenvolveu-se um aplicativo para dispositivos móveis, o RepitaApp, que, através da resposta a perguntas, gera tais estruturas. Em testes preliminares realizados com alunos evidenciou-se o potencial do aplicativo na demonstração dos elementos necessários para o funcionamento da estrutura. Espera-se que a utilização do RepitaApp minimize as dificuldades dos alunos nessa disciplina.*

1. Introdução

Nos cursos na área de Ciência da Computação, as disciplinas de Algoritmos e Programação são as que apresentam maior dificuldade para os alunos, com taxa de reprovação entre 30% e 40% [Beauboef 2005]. Os professores dessa disciplina frequentemente são surpreendidos pela falta de capacidade dos alunos de entenderem e resolverem problemas usando a lógica computacional [Iepsen 2013]. Dentro da disciplina de Algoritmos, o conteúdo de estruturas de repetição é um dos que impõe um maior desafio para a aprendizagem do aluno, uma vez que para que uma estrutura de repetição funcione vários elementos do algoritmo precisam funcionar de forma coordenada [Lahtinen 2005].

Considerando esse cenário, e a ausência de ferramentas para esse propósito, buscou-se uma forma de mostrar ao aluno os elementos necessários para a correta execução de uma estrutura de repetição e como esses elementos trabalham em conjunto.

De forma a tornar esse aprendizado interativo, colocando o aluno como agente principal no processo de aprendizagem, optou-se pelo desenvolvimento de um

aplicativo para dispositivos móveis. Para a implementação do aplicativo, elegeu-se a plataforma MIT App Inventor, que emprega a programação visual em blocos.

2. Fundamentação Teórica

2.1. Sobre a disciplina de Algoritmos

A disciplina de Algoritmos é ministrada como introdutória em praticamente todos os cursos da área de ciência da computação. Embora introdutória, tal disciplina é uma das que impõe maior dificuldade para o aluno [Beauboef 2005]. Um dos principais problemas encontrados é “identificar os pré-requisitos necessários para o desenvolvimento das competências de construção de algoritmos e programação, de forma a poder trabalhá-los” [Junior et al, 2005, p. 2.353].

Nessa disciplina, são ensinadas as principais estruturas necessárias para o desenvolvimento de programas de computador, dentre as quais se encontra a de repetição, cujo objetivo é, como se pode inferir do nome, repetir uma parcela específica de instruções um número pré-determinado ou não de vezes. As três estruturas de repetição básicas encontradas na vasta maioria das linguagens modernas são: enquanto (*while*), faça-enquanto (*do while*) e para (*for*). Temos na Figura 1 exemplos das três estruturas de repetição, usando a pseudolinguagem Portugal, que calculam o valor da tabuada de um número fornecido pelo usuário.

```
1 programa
2 {
3     funcao inicio()
4     {
5         inteiro i, j
6         i = 1
7         escreva("Informe um valor")
8         leia(j)
9
10        enquanto(i<=10){
11            escreva(i+" * "+j+" = "+i*j)
12            i++
13        }
14        i = 1
15
16        faca{
17            escreva(i+" * "+j+" = "+i*j)
18            i++
19        }enquanto(i<=10)
20
21        para(i=1; i<=10;i++){
22            escreva(i+" * "+j+" = "+i*j)
23        }
24    }
25 }
```

Figura 9. Exemplos das estruturas de repetição na pseudolinguagem Portugal.

Nas linhas 10 a 13, 16 a 19 e 20 a 22 da Figura 1, são mostrados exemplos das três estruturas de repetição básicas enquanto, faça-enquanto e para, respectivamente. Embora o conteúdo das estruturas, as instruções que serão repetidas, sejam essencialmente os mesmos, o funcionamento delas em si apresenta divergências:

- Enquanto: apresenta um teste lógico que determina se o conteúdo da estrutura deverá ser executado ou não. No caso da Figura 1, somente enquanto o valor da variável *i* for menor ou igual à 10 a repetição continuará;

- Repita: apresenta um comportamento semelhante ao “enquanto”, entretanto, apresenta uma diferença estrutural, o teste lógico só é verificado após a execução do conteúdo do laço, ou seja, ao passo que o “enquanto”, pode ser executado zero ou mais vezes, o “repita” será executado pelo menos uma vez;
- Para: concentra na estrutura todos os elementos necessários para seu funcionamento, a variável de controle, o teste lógico e a variação da variável de controle a cada iteração.

2.2. Sobre a utilização de dispositivos móveis na educação

Segundo pesquisa realizada pela Nielsen Ibope, no terceiro trimestre de 2015, existiam no Brasil 76,1 milhões de *smartphones* com acesso à internet (condição necessária para a instalação de aplicativos). Um aumento de 24 milhões em 1 ano [Ibope, 2016]. É inegável o interesse dos estudantes pelos seus dispositivos móveis. O New Horizon Report, por exemplo, reporta que escolas pelo mundo estão acolhendo e usando esses dispositivos a seu favor, com iniciativas do tipo B.Y.O.D – *bring your own device* (traga seu próprio aparelho), onde os estudantes são encorajados a levarem seus dispositivos para a sala de aula, onde serão incluídos no processo de aprendizagem [Becker, 2016].

Muito tem-se estudando sobre os efeitos do uso de dispositivos móveis nesse processo [Traxler 2007, 2009; Sharples 2010; Motiwalla 2007], inclusive de forma específica na disciplina de Algoritmos. Barcelos (2009) afirma que:

“A utilização dos dispositivos móveis no ensino de algoritmos tem a possibilidade de oferecer aos estudantes além da mobilidade e da aprendizagem móvel, diversas outras considerações: i) compreensão dos conceitos de algoritmos envolvendo informação audiovisual; (...)” [Barcellos 2009, p. 36]

Vemos assim que o aspecto audiovisual dos dispositivos móveis pode ajudar na melhor compreensão das estruturas dos algoritmos, uma vez que seu aspecto abstrato constitui uma das maiores dificuldades para os alunos. Além disso, a mobilidade desses dispositivos permite que eles facilmente sejam utilizados em sala de aula, servindo assim de apoio para os professores.

3. Metodologia

O desenvolvimento do aplicativo foi realizado em 6 etapas:

1. Levantamento das estruturas de repetição: foram identificadas as estruturas de repetição “enquanto”, “repita” e “para”. Comum à maioria das linguagens de programação.
2. Identificação dos elementos que determinam a execução da estrutura de repetição: foram identificados três elementos que devem funcionar de forma coordenada para a correta execução do laço, a variável de controle, o teste lógico e o incremento/decremento da variável a cada iteração do laço.
3. Definição das linguagens a serem utilizadas: foram utilizadas as linguagens Java, C/C++ e PHP.
4. Estruturação do aplicativo: para induzir o usuário a pensar no papel e no funcionamento de cada um dos elementos descritos no item 2, foram criadas perguntas, cujas respostas serão utilizadas na montagem do laço.
5. Implementação do aplicativo: para o desenvolvimento do aplicativo, foi utilizada a ferramenta MIT App Inventor, criada pela Google e atualmente gerenciada

pelo MIT (Massachusetts Institute of Technology). O App Inventor utiliza linguagem visual em blocos para o desenvolvimento de aplicativos para *smartphones* que utilizam o sistema operacional Android.

6. Testes: o aplicativo está sendo testado por um monitor da disciplina de Algoritmos em uma instituição de ensino.

4. Resultados e Discussão

Como resultado, desenvolveu-se o aplicativo RepitaApp que, através de perguntas respondidas pelo usuário, constrói os três principais laços de repetição nas linguagens de programação Java, C/C++ e PHP, além da pseudolinguagem Portugol.

É importante ressaltar que o objetivo do aplicativo é auxiliar o aluno na construção dos laços, e não os diferenciar ou indicar qual deve ser usado em determinadas situações.

O aplicativo possui apenas uma tela deslizante, cujo conteúdo é apresentado na Figura 2.

A opção por concentrar todas as informações em uma única tela objetiva facilitar a visualização de como as respostas afetam na construção da estrutura de repetição.

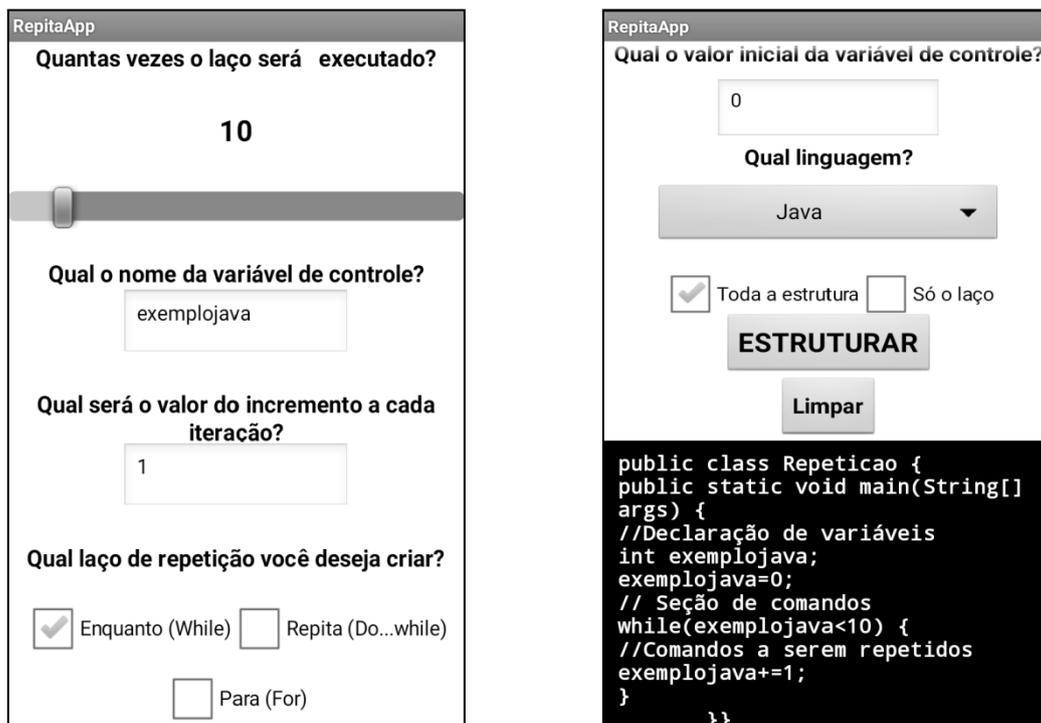


Figura 10 - Partes superior e inferior da tela do RepitaApp.

Por intermédio das perguntas, procura-se estimular a reflexão do aluno acerca da necessidade e propósito de cada um dos elementos que devem atuar de forma coordenada para que a estrutura de repetição surta o efeito desejado. As perguntas a serem respondidas, bem como sua finalidade, são:

- **Quantas vezes o laço será executado?** Nessa opção o usuário escolhe quantas iterações deverão ser executadas. Tal propriedade é dada pela coordenação do valor inicial da variável de controle, a condição lógica e o incremento da variável de controle.
- **Qual o nome da variável de controle?** Aqui o usuário escolhe um nome para sua variável que controlará quantas vezes o conteúdo do laço será repetido.

- **Qual será o valor do incremento a cada iteração?** Nessa opção o usuário deverá escolher o quanto o valor da variável de controle será incrementado a cada execução do laço.
- **Qual laço de repetição você deseja criar?** Escolhe-se aqui se o laço criado será o laço enquanto (*while*), repita (*do while*), ou para (*for*).
- **Qual o valor inicial da variável de controle?** Nesse campo o usuário define qual valor inicial será usado para a variável de controle.
- **Qual a linguagem?** Aqui é feita a escolha da linguagem, que pode ser Java, C/C++ ou PHP, apresentadas pelo aplicativo como demonstrado na Figura 3. O usuário também pode escolher gerar o laço em pseudocódigo. Tais linguagens foram escolhidas tendo em vista as principais linguagens utilizadas em cursos na área de informática.
- Por último, o usuário deverá assinalar se deverá ser gerado apenas o laço ou toda a estrutura necessária para que ele seja executado. Caso escolha ver apenas a estrutura de repetição, o aluno poderá visualizar como foram empregadas as informações por ele passadas na construção do laço, focando assim no conteúdo a ser aprendido. No segundo caso, o aluno poderá copiar o código com os elementos adicionais necessários para sua execução como a inserção do mesmo em um método principal e a importação de bibliotecas e executá-lo sem nenhuma modificação em seu computador, podendo então modificá-lo, interagindo com o código criado.

Exemplos de estruturas de repetição geradas pelo algoritmo encontram-se na Figura 4.



Figura 11. Linguagens disponíveis no Repita App

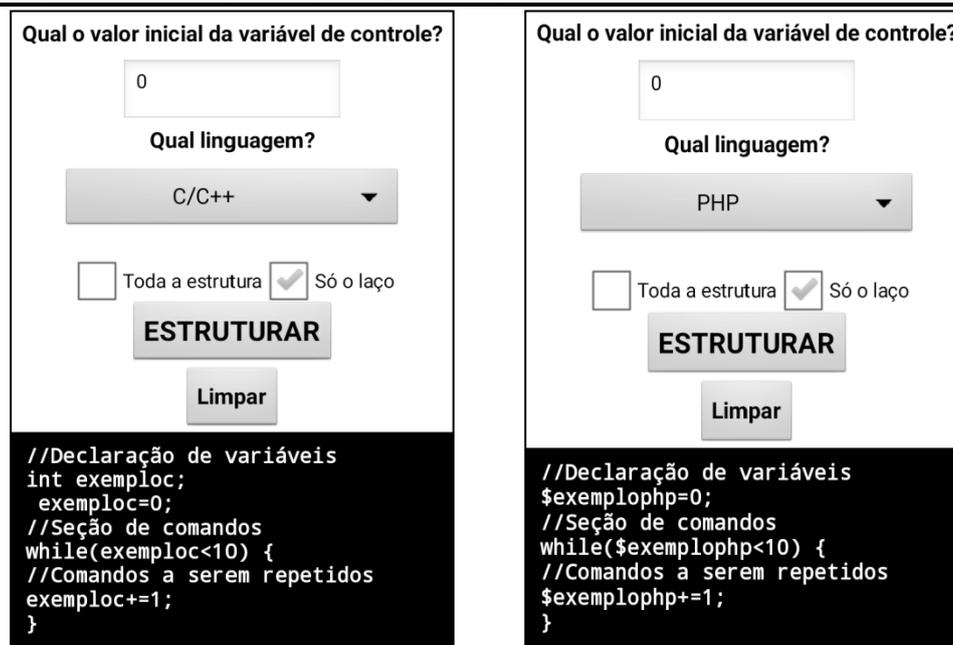


Figura 12. Exemplos de estrutura de repetição geradas nas linguagens C/C++ e PHP.

Após o desenvolvimento da primeira versão, foram realizados testes por um monitor da disciplina de Algoritmos. Devido ao *feedback* fornecido durante esses testes, pode-se corrigir alguns erros nos códigos gerados e evidenciar a eficiência do aplicativo, em especial no que diz respeito à percepção, por parte do aluno, da necessidade e funcionamento das partes que integram a estrutura de repetição.

5. Conclusão

O conteúdo de estruturas de repetição é um dos que impõe maior dificuldade para os alunos, dentro da disciplina de Algoritmos. Compreender a necessidade e funcionamento integrado dos elementos que promovem o correto funcionamento do laço constitui a raiz dessa dificuldade. Visando facilitar a visualização da integração desses elementos, foi desenvolvido o RepitaApp.

O desenvolvimento do RepitaApp foi facilitado pela ferramenta AppInventor, fazendo assim com que, após a idealização do aplicativo, o desenvolvimento fosse realizado de forma rápida e sem grandes dificuldades. Atualmente o aplicativo está sendo testado por um monitor da disciplina de Algoritmos em uma instituição de ensino que conta com os cursos Técnico em Informática Integrado ao Ensino Médio e Tecnologia em Análise e Desenvolvimento de Sistemas.

Após o encerramento dos testes com o monitor, o aplicativo começara a ser utilizado nas aulas de Algoritmos da referida instituição.

Futuramente, pretende-se adicionar a entrada de dados através de comandos de voz e a leitura do código gerado pelo aplicativo, de forma que ele possa ser utilizado por pessoas com necessidades especiais quanto à visão.

Referências Bibliográficas

- Becker, A., Samantha et al. NMC/CoSN Horizon Report: 2016 K. 2016.
- Barcelos, R., Tarouco, L., & Bercht, M. (2009). O uso de mobile learning no ensino de algoritmos. *RENOTE*, 7(3), 327-337.

- Beaubouef, T., & Mason, J. (2005). Why the high attrition rate for computer science students: some thoughts and observations. *ACM SIGCSE Bulletin*, 37(2), 103-106.
- Ibope, E. N. (2016). Pesquisa Ibope Nielsen Online. 2016.
- Iepsen, E. F. (2013). Ensino de algoritmos: detecção do estado afetivo de frustração para apoio ao processo de aprendizagem.
- Júnior, J. C. R. P., Rapkiewicz, C. E., Delgado, C., & Xexeo, J. A. M. (2005). Ensino de algoritmos e programação: uma experiência no nível médio. In *XIII Workshop de Educação em Computação (WEI'2005)*. São Leopoldo, RS, Brasil.
- Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005, June). A study of the difficulties of novice programmers. In *ACM SIGCSE Bulletin* (Vol. 37, No. 3, pp. 14-18). ACM.
- Motiwalla, L. F. (2007). Mobile learning: A framework and evaluation. *Computers & education*, 49(3), 581-596.
- Sharples, M., Taylor, J., & Vavoula, G. (2010). A theory of learning for the mobile age. In *Medienbildung in neuen Kulturräumen* (pp. 87-99). VS Verlag für Sozialwissenschaften.
- Traxler, J. (2007). Defining, Discussing and Evaluating Mobile Learning: The moving finger writes and having writ.... *The International Review of Research in Open and Distributed Learning*, 8(2).
- Traxler, J. (2009). Current state of mobile learning1. *Mobile learning*, 9.