

## Lord of Code: uma ferramenta de apoio ao ensino de programação

Marcella S. Café de Souza, Felipe A. de Melo Costa, Victor Luan Silva, Daniela Costa Terra

Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais (IFMG)  
Rua Afonso Sardinha, nº 90, Ouro Branco, MG – CEP: 36.420-000

{marcellacafe, dspairsray, victorluansilva}@gmail.com,  
daniela.terra@ifmg.edu.br

**Abstract.** *The Lord of Code (LoC) tool aims to help students to learn computer programming through a game where interpretation and resolution of code problems are required. This approach try to apply gamification to solve java code errors. A first experiment is developed from an incomplete game environment including questions about data structures and operational system's algorithms. The full proposal is based on tagging questions by subject and prerequisites. This mechanism is used to construct an alternative flow of questions to the incorrect answers.*

**Resumo.** *A ferramenta Lord of Code (LoC) visa auxiliar na aprendizagem de programação por meio de um jogo que estimula a interpretação e resolução de problemas em códigos escritos em Java. A abordagem pretende inserir o aspecto lúdico dos jogos a desafios envolvendo lógica de programação. Um experimento inicial é realizado a partir de esboço incompleto do ambiente de questões envolvendo algoritmos estudados nas disciplinas de Estrutura de Dados e Sistemas Operacionais. A proposta completa pressupõe a classificação de questões em relação ao assunto abordado e a conteúdos que sejam pré-requisitos a sua solução. Esta classificação é considerada no fluxo alternativo de questões a serem respondidas em caso de resposta incorreta.*

### 1. Introdução

Os cursos que envolvem a Computação no nível profissionalizante ou de ensino superior incluem disciplinas relacionadas a algoritmos e programação de computadores as quais promovem o desenvolvimento de habilidades essenciais para o bom rendimento do estudante e do regresso desses cursos. Todavia, o que se tem observado é que são essas as disciplinas responsáveis pelos maiores índices de reprovação e desistência. Segundo Rocha (1993), a média de desistência é de 60 %. Para Barbosa et al. (2011) isso acaba contribuindo para que o aluno desista do curso já que se trata de formação básica para a área.

Autores como Jesus e Brito (2009) dizem que as disciplinas de algoritmos e linguagem de programação exigem certas “[...] habilidades e competências como: raciocínio lógico, resolução de problemas e a capacidade de abstração da solução em uma representação formal e/ou em uma linguagem computacional”. Segundo os autores Santos e Costa (2006) a programação “[...] envolve, sobretudo entender problemas e descrever formas de resolução, de maneira imparcial, para que então sejam codificadas

em uma linguagem de programação”. Para o estudante conseguir absorver bem os conteúdos é preciso desenvolver habilidades a partir da prática de programação, sendo fundamental a interpretação do problema para análise das possíveis soluções.

O desenvolvimento de ferramentas voltadas para o auxílio da aprendizagem de linguagens de programação pode contribuir para um melhor aproveitamento das disciplinas que dependem deste conteúdo.

O trabalho aqui apresentado é desdobramento de uma proposta de trabalho interdisciplinar desenvolvida para o Curso de Licenciatura em Computação do Instituto Federal de Minas Gerais, *Campus Ouro Branco*. A proposta buscou aplicar estudos da psicologia da aprendizagem, em especial o de Paulo Freire (1996), para conceber uma ferramenta de auxílio ao ensino de programação orientada a objetos. O trabalho propõe o desenvolvimento de um jogo, denominado *Lord of Code (LOC)* para auxiliar na *expertise* da linguagem Java e na solução de problemas em algoritmos computacionais. O intuito é que o jogador faça suas análises de um código Java e indique qual resultado será retornado pelo algoritmo ou identifique qual trecho apresenta erro. As regras do jogo consistem em o jogador digitar a resposta correta, se ele acertar as questões então marcará pontos e acumulará ajudas que poderão ser usadas para as questões seguintes.

## 2. Metodologia

Algumas ferramentas, como a *Code Hunt*, estão disponíveis na web para auxiliar na aprendizagem da linguagem Java. A ferramenta criada pela Microsoft é um jogo em que são apresentados trechos de código com erros que precisam ser solucionados para o jogador prosseguir (TILLMANN et al., 2014). Outro exemplo é o da ferramenta *CodeWrite* que insere questões onde o estudante precisa implementar uma solução em código, segundo seus conhecimentos na linguagem (DENNY et al., 2011).

O trabalho aqui apresentado, o LOC, utiliza preceitos e pesquisas da psicologia e da educação para desenvolver uma ferramenta em formato de jogo capaz de auxiliar o estudante a desenvolver habilidades de programação necessárias a disciplinas diversas da Computação.

### 2.1. Esboço inicial como Projeto Interdisciplinar

Utilizou-se para o desenvolvimento do jogo o HTML5, o CSS3, MySql e PHP. O jogo incluiu questões simples de algoritmos utilizados nos estudos de estruturas de dados e sistemas operacionais, inclusive trechos com problemas em códigos para com múltiplas *threads*. A ferramenta pretende disponibilizar provas de certificação oficiais para programadores Java, como o *Sun Certified Java Programmer (SCJP)* [MUGHAL e RASMUSSEM 2008]. Também nesse caso, além da sintaxe básica do Java serão abordados problemas envolvendo questões de outras disciplinas da computação que estão interligadas com a programação.

A Figura 1 exibe a tela principal do jogador. Após descrever no espaço “Digite a resposta...” e clicar em “Enviar” o usuário irá obter 100 pontos e acumular 2 ajudas, em caso de resposta correta. O botão em formato de seta permite ao jogador avançar para a próxima questão. Ao clicar no botão “Ajuda” uma dica de solução será apresentada ao jogador. Se a dica obtida levar o jogador à resposta certa então o número de ajudas é decrementado. O intuito é propor um desafio, motivando a autonomia do estudante para descobrir a solução do problema, conforme defendido por Paulo Freire (1996) que destaca a importância do instinto pesquisador e ativo na busca pelo conhecimento. A

cada vez que o jogador errar a resposta o número de tentativas será decrementado. Quando “Tentativas” chegar a zero o jogo termina. A pontuação do jogador é exibida no campo “Pontuação”. O ranking dos melhores jogadores cadastrados é exibido quando o jogo termina. É possível finalizar o jogo a qualquer instante clicando em “Deslogar”.

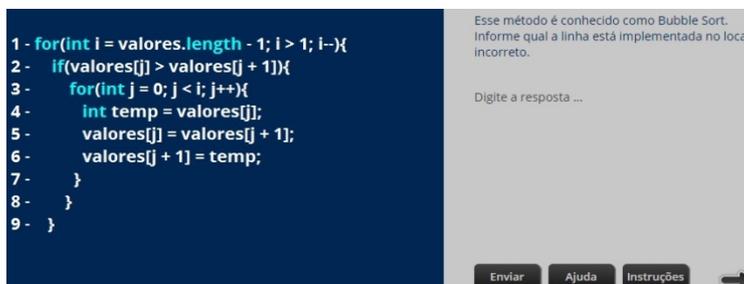


Figure 1. Tela do jogo LoC

Um teste inicial foi realizado com estudantes do IFMG, Campus Ouro Branco, que já haviam cursado as disciplinas de algoritmos e sistemas operacionais. *Links* de acesso ao artefato e ao questionário de opinião sobre o uso da ferramenta foram enviados por e-mail. O resultado obtido revela um grau de dificuldade das questões entre “regular” e “difícil”. Quanto ao nível de auxílio da ferramenta opinaram estar entre “bom” e “poderia ser melhor”. Quando perguntados sobre a qualidade do *design* de *interface* classificaram como “muito bom” e “bom”. Por ser um artefato ainda em desenvolvimento, não foram realizados teste mais elaborados.

## 2.2. Proposta Completa do LoC

Com base na necessidade de se realizar avaliações diagnósticas e correção das deficiências de aprendizado detectadas a ferramenta deverá evoluir e incluir a categorização dos usuários como estudantes ou professores. Os últimos poderão fazer uso do banco de questões públicas do LoC ou incluir novas questões ao seu perfil. O papel do usuário professor é elaborar o fluxo de perguntas que comporão o jogo após classificar cada uma em relação ao assunto abordado e aos conteúdos que sejam pré-requisitos do assunto principal. O pressuposto é que o conteúdo e seus pré-requisitos sejam compreendidos para a resolução correta de uma questão.

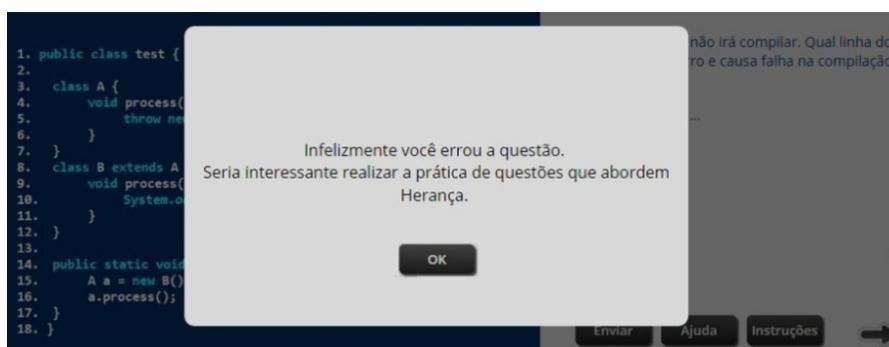
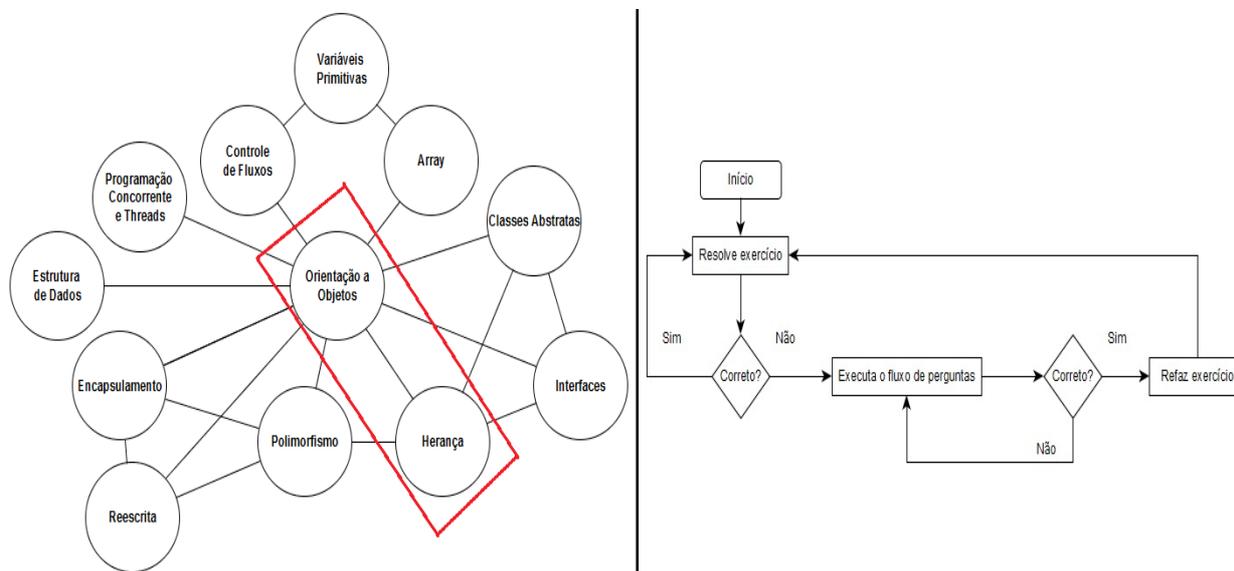


Figure 2. Tela informativa de resposta incorreta

Esta classificação em conteúdos envolvidos é considerada para a montagem de um fluxo alternativo de perguntas adicionais a serem respondidas caso determinada resposta esteja incorreta (Figura 2). Na montagem do teste o usuário deverá definir a ordem de precedência para cada um dos conteúdos envolvidos na solução de cada

questão. As perguntas do fluxo principal que envolverem pré-requisitos incluirão uma chave que referenciará um fluxo alternativo de perguntas elaboradas exatamente para trabalhar os pré-requisitos especificados (Figura 3).



**Figure 3. Classificação e interdependência de conteúdos da POO (à esquerda).  
Fluxograma de resolução do teste (à direita)**

O grafo exibido na Figura 3 destaca a relação entre os conceitos da programação orientada a objetos tais como, o polimorfismo, a herança, a reescrita, *etc.* Como em todo campo de estudos, na orientação a objetos também se observa uma teia conceitual em que alguns assuntos devem ser compreendidos primeiro já que são a base conceitual para o aprendizado de outros. Dessa maneira quando o usuário estudante, convidado pelo professor a executar um teste, errar a questão sobre o assunto “polimorfismo” ele será avisado (Figura 2) que deverá resolver um fluxo adicional de perguntas envolvendo os conceitos ou requisitos associados. Após resolver o fluxo alternativo o usuário terá nova chance de responder à questão que havia errado. Caso contrário, se sua resposta estiver correta ele prosseguirá para questões restantes do fluxo principal. Ao projetista do teste, o usuário professor, será dada a possibilidade de aninhar fluxos alternativos a questões que já sejam parte de outros fluxos alternativos. A execução de um fluxo de teste gerará um registro associado ao usuário estudante respondente. Os registros de execução poderão ser analisados pelo projetista para detectar com precisão as deficiências de aprendizagem apresentadas, individualmente.

### 3. Considerações finais

Melhorias nas metodologias de ensino-aprendizagem são necessárias para um melhor aproveitamento das disciplinas dos cursos da área diretamente relacionadas ao projeto de soluções computacionais.

Este texto propõe uma ferramenta que auxilie no ensino de lógica e sintaxe de programação ao oferecer ambiente de aprendizagem capaz de despertar o interesse e envolvimento do estudante. Conforme proposto na seção 2.2, com a elaboração de um fluxo de perguntas devidamente classificadas por assunto e conteúdos envolvidos espera-se que o estudante possa ser motivado a identificar onde estão suas falhas e com autonomia poder superá-las.

Além de fluxos de questões elaboradas pelo usuário professor a grupos de estudantes específicos, considera-se a inclusão de testes públicos disponíveis para acesso por qualquer usuário. Um exemplo é a disponibilização de provas de certificação oficiais para programadores, como o conhecido teste SCJP para Java. A denominação da ferramenta LoC, de *Lord of Code*, em Português, “senhor dos códigos”, surge da possibilidade dada ao estudante para que este mesmo possa desenvolver a *expertise* necessária em programação.

#### 4. Referências bibliográficas

- Boulic, R. and Renault, O. (1991) “3D Hierarchies for Animation”, In: *New Trends in Animation and Visualization*, Edited by Nadia Magnenat-Thalmann and Daniel Thalmann, John Wiley & Sons Ltd., England.
- Barbosa, Leônidas S., Teresa CB Fernandes, and André MC Campos. "Takkou: uma ferramenta proposta ao ensino de algoritmos." *XVIII Workshop sobre Educação em Computação (WEI 2011)*. 2011.
- De Jesus, Andreia, and Gláucia Silva Brito. "Concepção de ensino-aprendizagem de algoritmos e programação de computadores: a prática docente." *Varia Scientia* 9.16 (2009).
- Denny, Paul et al. CodeWrite: supporting student-driven practice of java. In: *Proceedings of the 42nd ACM technical symposium on Computer science education*. ACM, 2011. p. 471-476.
- Freire, Paulo. "Pedagogia da Autonomia. 24ª edição." *Paz e Terra* (1996).
- Mughal, Khalid, and Rolf Rasmussen. *A programmer's guide to Java SCJP certification: a comprehensive primer*. Addison-Wesley Professional, 2008.
- Rocha, H. (1993) “Representações Computacionais Auxiliares ao Entendimento de Conceitos de Programação”, In: “Computadores e Conhecimento: Repensando a Educação”. Livro organizado por Valente, J. A. Editora Unicamp.
- Tillmann, Nikolai et al. Code hunt: Searching for secret code for fun. In: *Proceedings of the 7th International Workshop on Search-Based Software Testing*. ACM, 2014. p. 23-26.