

Methodology based on flipped classroom and problem solving related to students' habits: a proposition for teaching programming for beginners

Original Title: Metodologia baseada em sala de aula invertida e resolução de problemas relacionado ao cotidiano dos estudantes: uma proposta para ensinar programação para iniciantes

Lucia Maria Martins Giraffa^{1,2}, Luana Müller²

¹ Escola de Humanidades – PPGEdU – PUCRS – Brasil

² Faculdade de Informática – PUCRS – Brasil

ARTICLE INFO

Article history:

Received 19 June 2016

Accepted 25 October 2016

Available online 1 August 2017

Keywords:

Teaching Programming for
Beginners
Flipped Classroom
Problem Solving Methodology

ISSN: 2594-5602

DOI:

10.14210/ijctthink.v1.n1.p52

ABSTRACT

INTRODUCTION: This paper presents the lessons learned from a methodology created to support the teaching of programming for beginners based on Flipped Classroom and Problem Solving approaches using a virtual classroom and exercises linked with student leisure interests. **OBJECTIVES:** To help students stay on schedule and avoid discipline drop out. **METHODOLOGY:** we created a virtual classroom with different resources with videos, code examples, and exercises statements that allowed us to approach the interests of the students through situations anchored in their leisure habits. **RESULTS:** options and possibilities to integrate multiple spaces and tools on the Internet, examples of exercises related to the daily life of the students

RESUMO.

INTRODUÇÃO: Este artigo apresenta uma metodologia para apoiar o ensino de programação para iniciantes baseada na abordagem de Sala de Aula Invertida e na resolução de problemas. **OBJETIVOS:** Auxiliar na permanência dos estudantes iniciantes em programação e evitar o abandono na disciplina. **METODOLOGIA:** Buscou-se criar alternativas para ampliar o espaço de encontros presenciais com a criação de uma sala de aula virtual e uso de enunciados de exercícios que nos permitissem aproximar dos interesses dos estudantes por meio de situações ancoradas nos seus hábitos de lazer. **RESULTADOS:** Discute-se a questão dos desafios e possibilidades da integração de múltiplos espaços e ferramentas existentes na Internet, a questão da motivação para realizar as tarefas a partir da oferta de exercícios que estejam relacionados ao cotidiano dos estudantes.

1. Introdução

A preocupação com a evasão escolar é um problema que afeta todos os níveis de escolarização e os cursos na área de Computação não fogem a esta regra. Programas da UNESCO¹ buscam incentivar docentes e instituições a buscarem alternativas para mitigar a desistência e suas consequências, as quais impactam não somente as instituições de ensino, mas também a sociedade como um todo. Dentre as múltiplas diretrizes incentivadas pelo relatório destaca-se uma que motiva a escrita deste registro: socializar e difundir experiências e ações relacionadas a metodologias que auxiliem a premência do estudante e conseqüente continuidade do seu estudo.

A relevância e atualidade do tema abordado neste artigo podem ser mensuradas pelo fato dos estudos investigativos relacionados ao ensino de programação não serem recentes e há tempos preocupam os docentes da área de Computação justamente por não haver uma solução definitiva para o problema. A preocupação com o tema agravou-se nos últimos anos em face da baixa procura das carreiras envolvendo engenharia e programação. A Sociedade Brasileira de Computação (SBC²), as sociedades americanas

¹ <http://www.unesco.org/new/pt/brasil/education/>

² <http://www.sbc.org.br/>

ACM³, IEEE⁴ e diferentes entidades europeias⁵ publicam relatórios e boletins em seus sites alertando para o problema e deflagrando uma ampla reflexão na comunidade docente e discente. Eventos da área de Computação tais como *Technical Symposium on Computer Science Education-SIGCSE*⁶, *European Counterpart*⁷, *Innovation and Technology in Computer Science Education (ITiCSE)*, *Computer Science Education: Innovation and Technology-CSEIT*⁸ são fóruns interessantes para buscar elementos e experiências relacionados ao desafio de ensinar programação e a questão da baixa procura pelas carreiras de Engenharia e Computação. ACM e IEEE também publicaram recomendações relacionadas aos currículos e ensino na área de Computação⁹ buscando garantir padrões de qualidade e indicando alternativas para captação de novos estudantes e fixação dos mesmos nas carreiras.

Os fatores que levam os jovens a optarem por outras carreiras não serão discutidos neste artigo. O foco é discutir e apresentar uma proposta de ensino que motive o estudante a aprender e auxilie o discente a persistir na carreira escolhida. Acredita-se que além das questões relacionadas às competências e habilidades prévias que o estudante deve possuir para estudar Computação, uma parcela significativa da evasão pode ser controlada pela atuação docente.

A forma com que o espaço presencial e virtual se organiza influi na maneira com que os estudantes se relacionam entre si e com seus professores. Um ambiente que incentive a leitura, a troca de experiências e o diálogo pode auxiliar o estudante a manter-se no curso. O uso de espaços virtuais pode auxiliar a interação do grupo e permitir a criação de uma comunidade virtual de aprendizagem.

O ensino de programação é associado ao ensino de Algoritmos. Ou seja, para aprender a programar o estudante deve aprender a pensar algorítmicamente, uma vez que a organização lógica/estrutural do programa deve ser possível de ser executada pelo computador. O computador possui uma estrutura interna muito simples se comparada a nossa estrutura de pensamento. Nós humanos temos a capacidade de processar informações incompletas, confusas, redundantes, mal organizadas e mesmo assim conseguimos resolver problemas, até quando eles são mal formulados. Nossa impressionante capacidade de estabelecer associações e interligações nos permite criar alternativas de soluções aos mais diversos problemas. Porém, para programar é necessária uma abordagem mental diferente e o entendimento de como funciona esta ferramenta: o computador. As operações são básicas e simples se comparadas as que realizamos quando pensamos. E a organização física e lógica deste recurso é baseada em construtores relacionados à Matemática.

Existem diferentes paradigmas para se programar computadores. Assim como as demais áreas do conhecimento, o ensino de programação evoluiu em função dos avanços tecnológicos e das necessidades da sociedade. Durante muitas décadas o ensino de programação para iniciantes foi realizado usando o paradigma da programação estruturada com linguagens de programação tais como Pascal e C. Por esta razão a grande maioria dos estudos envolvendo o ensino de programação usa como base este paradigma. A primeira autora deste artigo leciona Algoritmos e Programação para iniciantes desde 1987 onde temos vivenciado as dificuldades e as limitações

³ <http://www.acm.org/>

⁴ <http://www.ieee.org/index.html>

⁵ <http://www.informatik.uni-trier.de/~ley/organizations.html>

⁶ <http://sigcse2016.sigcse.org/>

⁷ <http://www.informatics-europe.org/component/phocadownload/category/10-reports.html?download=42:european-computer-science-takes-its-fate-in-its-own-hands>

⁸ <http://www.cseducation.org/index.html>

⁹ <http://www.acm.org/education/curricula-recommendations>

encontradas pelos estudantes ao chegarem à universidade. Nestes mais de 20 anos atuando no ensino de programação é visível o impacto da perda de qualidade da formação básica que os estudantes estão recebendo no Brasil. Evidente que o lócus de nossa pesquisa e observação é a cidade de Porto Alegre, Rio Grande do Sul. Entretanto, as políticas e orientações curriculares no Brasil recebem forte influência do governo central (federal). Além disso, a crise na formação de professores, a baixa procura pelas carreiras docentes, especialmente na área de Matemática e Física, geram reflexos na formação dos estudantes e impactos negativos na universidade. Nos anos de 2012 a 2014 conduzimos pesquisas para identificar as causas do abandono e desistência nas disciplinas de 1º nível nos cursos de programação (Giraffa & Moraes, 2012; Giraffa & Mora, 2013; Giraffa, Moraes & Uden, 2014) e encontramos que, apesar das deficiências relacionadas aos pré-requisitos de Matemática, os estudantes apontam que os fatores principais são: o não entendimento dos enunciados, exercícios dissociados de problemas “reais” e pouca disponibilidade extraclasse para estudos são obstáculos para sua aprendizagem. A questão não pode ser colocada de forma reducionista uma vez que são muitos os fatores intervenientes para que o abandono e a evasão da disciplina ocorram. No entanto, a pesquisa mostrou um cenário um tanto diferente daquele costumeiramente apontado em pesquisas na literatura. Como no caso do trabalho desenvolvido pelos ingleses Bennedsen e Caspersen (2007) foi o usado como impulsionador da investigação que originou a proposta metodológica aqui relatada. O trabalho dos autores apresenta o resultado de uma pesquisa realizada no ano de 2007 que objetivava avaliar indicadores que poderiam ser “padrões” nas instituições de todo o mundo relacionados às taxas de evasão em cursos de introdução à programação. O artigo descreve o encaminhamento da pesquisa e seus resultados. O número de instituições que responderam ao instrumento (questionário) foram 67 universidades dos cinco continentes. Os autores esperavam um número expressivo de instituições respondentes do instrumento e ficaram bastante frustradas com o retorno. Mas até este retorno foi objeto de reflexão e influenciou os autores a tecerem conclusões menos firmes e generalizadoras, mas não menos importantes para os docentes da área de Computação. Esta pesquisa deu início a toda uma trajetória investigativa destes autores relacionada ao ensino de programação para iniciantes durante a qual foram investigados os aspectos relacionados às habilidades e competências necessárias para se aprender a programar orientado a Objetos¹⁰. Quando se buscou confrontar estes estudos com similares no Brasil (Delgado et al, 2005; Raabe, 2005; Raabe, 2006; Raabe, 2007) observaram-se indicadores semelhantes. Ou seja, a falta de competências e habilidades relacionadas a conteúdos de Matemática, limitações no uso da memória para lembrar o que já foi aprendido, capacidade de ler enunciados e entendê-los de forma clara e analítica são fatores que levam os estudantes a não conseguirem entender os conceitos basilares da programação.

Tendo estes elementos identificados na literatura e que combinavam com a observação e informações relacionadas à nossa prática organizamos uma proposta/abordagem metodológica para trabalhar com os estudantes de maneira a envolvê-los mais no processo e permitir que trabalhassem de forma mais participativa na construção do seu conhecimento. A proposta metodológica sugerida pelas autoras baseia-se em dois eixos norteadores:

1. *Problem Solving* (Resolução de Problemas) a qual é entendida por:
“[...] solução de problemas baseia-se na apresentação de situações abertas e sugestivas que exijam dos estudantes uma atitude ativa ou um esforço para

¹⁰ <http://www.daimi.au.dk/~mec/publications/index.html>

buscar suas próprias respostas, seu próprio conhecimento. O ensino baseado na solução de problemas pressupõe promover nos estudantes o domínio de procedimentos, assim como a utilização dos conhecimentos disponíveis, para dar resposta a situações variáveis e diferentes”.

(POZO e ECHEVERRÍA, 1988)

2. *Flipped Classroom* (Sala de Aula Invertida) a qual é entendida por:

“[...] o que tradicionalmente é feito em sala de aula, agora é executado em casa; e o que tradicionalmente é feito como trabalho de casa, agora é realizado em sala de aula”.

(BERGMANN e SAMS,2016)

A adoção destas duas abordagens relacionadas a metodologias ativas combinadas com os hábitos de lazer dos estudantes nos permitiram estabelecer um cenário de experimentação associado ao novo currículo criado para o curso de bacharelado em Ciência da Computação na Faculdade de Informática (FACIN) da PUCRS¹¹. O curso encontra-se no 3º semestre de implementação e o estudo de caso baseia-se nos resultados observados nas turmas da disciplina de Fundamentos de Programação neste período. Utilizamos como instrumentos de coleta de dados as observações acerca do comportamento dos estudantes, depoimentos espontâneos, depoimentos induzidos em perguntas no fórum após a entrega das notas finais e análise dos resultados das avaliações das 4 turmas. Dada a similaridade de conteúdos entre esta disciplina e sua antecessora (Algoritmos e Programação I- Alpro I) comparamos os resultados, índice de evasão e notas finais, aos dados históricos desta disciplina nos últimos 6 anos. A disciplina de Alpro-I possuía alto índice de evasão. Os estudantes desistiam, em número significativo (45% a 70%) antes da conclusão do semestre. Em pesquisa realizada por (Giraffa, Moraes, 2012; Giraffa, Mora 2013) discutimos estes números e suas implicações. Um dos fatores mais relevantes para evasão ou cancelamento estava relacionado a metodologias utilizadas pelos professores. Observamos que os estudantes que persistiam na sua grande maioria obtinham aprovação.

A questão metodológica é tida na literatura tanto de Ensino de Programação, quanto de Educação como um elemento fundamental para permanência do estudante. Trabalhos como Tello (2007) e Tinto (1997) já apontavam estas questões. Tinto é considerado como referência nos estudos sobre evasão, e o autor salienta que a integração acadêmica, seja no âmbito pessoal ou institucional pode ter relação com o abandono e/ou permanência. O autor ressalta a importância de as IES criarem programas, principalmente para estudantes do primeiro ano, que permitam a interação entre os estudantes e dos estudantes com os professores e demais atores da Educação, tanto em assuntos acadêmicos como sociais. Ou seja, para estudantes que chegam a universidade, especialmente no caso dos nossos estudantes, recém-saídos do Ensino Médio, a adoção de estratégias que aproximem seus hábitos de lazer (aspecto social) (sociais) permite criar um ambiente mais próximo ao que estão acostumados e sem perda de qualidade nas tarefas propostas. Muitas vezes quando são mencionados os enredos criados para os exercícios e provas a reação dos colegas é de que está havendo uma certa flexibilização ou perda de rigor na oferta de atividades tanto para prática, bem como para avaliação. Os exercícios que temos proposto ao contrário tem se mostrado com maior dificuldade do que àqueles tradicionalmente encontrados na literatura padrão.

¹¹ <http://www.pucrs.br/facin/curso/ciencia-da-computacao/>

Uma crítica que fazemos é que os enunciados em muitos livros didáticos são diretos e não permitem ao estudante realizar uma análise do texto identificando que informações são relevantes e as quais são “acessórias”, ou seja, aquelas que criam o cenário associado ao problema. A importância em exercitar a leitura crítica e análise do conteúdo é uma competência importante a ser desenvolvida. A exemplo disto citamos um clássico: “Elaborar um programa para ler 10 valores e armazenar num vetor. Escrever o vetor lido e o maior valor armazenado”. Temos substituído este enunciado por texto mais elaborados permitindo ao estudante refletir acerca da estrutura necessária para armazenar os dados fica subentendida e sua escolha faz parte da decisão do aluno a o propor a solução. Um exemplo disto é: “Os membros da família Targaryen uniram-se para realizar um torneio e ao final de 10 provas precisam saber quem acumulou mais pontos. Escreva a lista dos participantes e respectivos pontos e indique quem foi o vencedor. Se houver mais de um, em caso de empate com relação à pontuação, seu programa deverá indicar isto”. Com este exercício o estudante poderá fazer várias versões em função da maneira que percebe o problema e a estrutura que escolhe, decorrente disto terá maior ou menor complexidade em resolver sua proposta. Implicando no uso de vários conteúdos. Além disto, amplia a variedade soluções apresentadas pelos diferentes alunos.

Existem diferentes formas do professor abordar o processo de ensino e o de aprendizagem em relação aos seus estudantes. Segundo Becker (2001), essas formas podem ser sintetizadas em três pedagogias fundamentais: centrada no professor, a qual tende a valorizar relações hierárquicas e a anular a capacidade criativa do estudante, bastando que este repita o que já lhe é apresentado “pronto”; centrada no estudante, a qual em geral acaba por atribuir ao estudante qualidades que este não tem, como domínio do conhecimento sistematizado e domínio de didáticas; e centrada na relação estudante-professor, a qual tende a desabsolutizar a hegemonia prévia de ambas as partes, pois estas trazem suas “bagagens” e constroem o conhecimento em conjunto. Cada uma destas pedagogias pode ser colocada em prática através de metodologias de ensino específicas. A pedagogia centrada na relação estudante-professor pode adotar a metodologia dialética, que pressupõe ser o estudante um ser ativo no processo de aprendizagem e o professor o problematizador de situações e articulador do processo de conhecimento (Portal, 2003). Neste contexto, o professor não é mais apenas o organizador do processo de aprendizagem, este é o mediador das ações dos estudantes. Assim, o professor deve provocar e propiciar as atividades do estudante; permitindo ao mesmo realizar a ação de análise e reflexão crítica. O estudante, por sua vez, deixa de ser um mero assistente e tornar-se um agente de busca, seleção e construção do seu conhecimento, tornando-se corresponsável pelo seu aprendizado.

Neste sentido a abordagem de sala de aula invertida cumpre um papel importante na organização metodológica que desenvolvemos. São indicadas as leituras prévias serem realizadas de maneira que o estudante possa tomar ciência do conteúdo antes da atividade presencial. Ele/ela deve ler, buscar compreender os conceitos fundamentais, analisar os exemplos e fazer suas anotações para poder discutir na sala com o professor e só colegas. Nisto reside um entrave em termos de hábitos de estudos e vivências dos estudantes. Os estudantes chegam a universidade sem a autonomia e a proatividade necessárias. Cabe a nós incentiva-los e explicar como devem fazer para tirar proveito do estudo antecipado. Costumamos explicar como ler os capítulos dos livros, como pesquisar na Internet os assuntos, que sites soa confiáveis e relevantes, como selecionar as informações que podem auxiliá-los no estudo. Inclusive sugerimos que façam anotações identificando que conteúdos não entenderam, quais exemplos foram mais significantes e assim por diante. Os alunos precisam ser ensinados a fazer pesquisa, a ler

os textos da maneira adequada para que possam significar a tarefa da leitura previa. Sem este direcionamento raros farão a leitura e, se fizerem, não a farão com o resultado que desejamos. Inverter a forma tradicional de expositiva para leitura previa vai requerer do professor também inverter sua lógica de planejamento e intervenção junto aos discentes. “Inverter” para o estudante é também inverter para o docente. Lembra que nós, os professores, fomos educados e ensinados a ensinar na logica expositiva. O desafio se faz tanto aos discentes bem como para os docentes. Sugere-se que comecem estas leituras desde a 2ª aula de maneira que nos encontros virtuais os alunos estejam habituados a fazer isto.

2. Estruturação da Sala Virtual de Apoio

O esquema das aulas utilizadas nestas turmas de Fundamentos da Programação se baseia em um conjunto de materiais disponibilizados de maneira prévia para os estudantes, através do ambiente virtual de aprendizagem (Moodle¹²) da disciplina. Um ambiente virtual de aprendizado deve ser muito mais que um repositório de materiais. Se utilizado em sua totalidade de recursos pode tornar-se uma extensão da sala de aula, agregando e auxiliando o processo de ensino e aprendizagem. O ambiente virtual foi utilizado com essas turmas de forma a estender as atividades das salas de aula presencial e apoiar as aulas realizadas à distância. Existem atividades mediadas presencialmente pelo professor (aula no mesmo espaço físico) e outras atividades nas quais o professor interage com os estudantes de forma síncrona ou assíncrona. Este espaço virtual que consideramos como extensão da sala de aula presencial é o elemento basilar da relação com os estudantes. O espaço de interação virtual (SAV – Sala de Aula Virtual) aliado a uma escolha de exercícios (desafiadores e associados a problemas que os estudantes vão enfrentar como programadores) fez a diferença nos resultados das turmas. Para maiores detalhes sugere-se a leitura de Giraffa (2015a).

Os recursos utilizados na SAV para suporte às atividades não presenciais baseiam-se nos seguintes serviços:

1. Fórum de Avisos: espaço usado pelo professor para fazer a comunicação com os estudantes. Ele substitui a lista de e-mails tradicionalmente usada para divulgar notícias e avisos aos estudantes. Pode ser usado para divulgar eventos, oportunidades de cursos, leituras e outras informações relevantes;
2. Fórum das descobertas: espaço usado pelos estudantes para compartilharem suas descobertas e promover a interação entre eles;
3. Fórum dos monitores: espaço especial para os estudantes conversarem com a monitoria;
4. Fóruns de compartilhamento: espaços usados pelos estudantes para compartilharem as soluções propostas por eles para todos os exercícios desenvolvidos no decorrer do semestre;
5. Cronograma das Atividades: a FACIN possui um sistema de reservas que fornece suporte aos professores e coordenação de recursos computacionais que permite distribuir as aulas práticas que correm nos laboratórios. Desta forma o professor pode disponibilizar seu planejamento e ir alterando conforme o ritmo das aulas e os naturais ajustes que são feitos no cronograma em função da

¹² O ambiente pode ser outro qualquer desde que possua as mesmas funcionalidades disponíveis ao professor.

realidade da sua sala de aula. Este cronograma pode ser exportado em padrão HTML e é colocado na sala de aula virtual na forma de uma página ou arquivo.

6. Tópicos contendo os Arquivos: disponibilizados em formato PDF, contendo os materiais relacionados aos tópicos a serem abordados os materiais da disciplina.
7. Materiais complementares em formatos multimídia: uso de pequenos filmes gerados com uso de ferramenta de captura, com exercícios realizados no ambiente de programação, com a solução comentada passo-a-passo. Foram criadas videoaulas e disponibilizadas no YouTube com explicações detalhadas do uso de recurso técnicos tais como: configuração das variáveis de ambientes, tratamento de arquivos, uso de ferramentas extras ao ambiente de programação;
8. Mensagens: utilizadas para esclarecimento de dúvidas e comunicação privada do professor com seus estudantes e vice-versa;
9. Podcasts (arquivos de áudio em formato mp3): usados para complementar o trabalho desenvolvido nos encontros presenciais e fornecer feedback/orientação aos estudantes. Maiores detalhes do uso destes recursos foram relatados em (Giraffa e Faria, 2012);
10. Sala de entrega de Tarefas: usadas para envio de trabalhos relacionados a nota de exercícios os quais integram da nota final da disciplina.

A figura 1 apresenta um recorte da sala de aula criada no Moodle.

Figura 1: Sala Moodle da disciplina

A metodologia utilizada na condução da disciplina, como mencionado anteriormente, baseia-se na resolução de problemas organizados em ordem crescente de complexidade. Toda a questão da leitura referente à teoria e seu aprofundamento dos conteúdos fica por conta do estudante. Quando da adoção de algum livro, a sequência de leitura para este recurso é disponibilizada.

Cada aula inicia com um resumo dos principais pontos indicados para leitura e depois são desenvolvidas atividades práticas relacionadas com a lista de exercícios disponibilizada na página ou entregue durante a aula. Raramente utilizamos algum slide de apoio (tipo Power Point e similares) para esta contextualização. Usamos muito esquemas para promover esta contextualização e resgate de conceitos. A capacidade de resolver um problema e expressar a solução via um algoritmo, vai requerer que o estudante saiba analisar o problema que recebeu e seja capaz de situá-lo dentro de um contexto onde existem diversas classes de problema. Depois disto, o estudante deve ser capaz de perceber os componentes que constituem aquele problema, ter bem claro qual a solução esperada, identificar os dados disponíveis para serem computados e, finalmente, organizar uma estratégia de solução baseada nos seus pré-requisitos. Podendo, também, complementar os dados iniciais fornecidos pelo enunciado, caso seja necessário. Esta etapa de pré-análise do problema pode ser feita de forma mais informal, motivando o estudante a explorar de forma mais livre o enunciado do problema, fazer conjecturas a respeito de sua interpretação, verificar se os colegas possuem a mesma percepção do problema e outras. Logo, é importante que se crie hábitos de leitura crítica dos enunciados. Isto é, os estudantes invistam tempo em atividade extraclasse para realizarem leitura criteriosa do enunciado que receberam. Para tal, o professor deve conduzir esta atividade lendo o enunciado ou solicitando a um dos estudantes que o façam de forma sistemática sublinhando verbos que detonam as ações esperadas e sugere-se que separem as informações tendo por base o esquema E (entrada) P (processamento) S (saída).

Evita-se a resolução dos exercícios por parte do professor e busca-se uma postura menos diretiva por parte do docente. O problema será resolvido utilizando as contribuições dos estudantes. Neste caso, o professor funciona como elemento mediador que coloca as informações no quadro (ou em dicas nos fóruns da sala virtual) e as disponibiliza para todos os estudantes. Ou seja, o professor funciona como facilitador e não um guia que apresenta uma solução passo-a-passo. O professor deve incentivar os estudantes a colocarem suas soluções (corretas ou não) para serem compartilhadas com os colegas nos fóruns criados (fóruns de compartilhamento) para discutir os tópicos em estudo. Desta forma, a turma pode constatar que não existe um modelo único de solução. Cada pessoa é um ser único no sentido de processar as informações. Cada um pode perceber os problemas de forma diferente. Logo, cada um elabora soluções diferentes. E, todas estas soluções podem estar atendendo ao problema. O que vai ocorrer é que algumas são mais otimizadas do que outras. Não se pode exigir que em uma fase inicial de construção de conhecimento, o estudante, além de conseguir elaborar o algoritmo, desenvolva-o de forma otimizada. A etapa de otimizar (melhorar o algoritmo já elaborado) é subsequente ao processo de elaboração da primeira versão. Esta etapa é muito mais complexa e difícil para o estudante iniciante.

A fim de reforçar o processo de construção de diferentes soluções para um mesmo problema, disponibilizam-se na página da disciplina (link devidamente identificado) vários exemplos de soluções construídas por outros estudantes (até de turmas anteriores) e monitores. A monitoria desempenha um papel importante neste contexto. Os estudantes têm a oportunidade de conversar sobre suas dúvidas com outro colega mais experiente. Em muitas situações, outro colega com faixa etária mais próxima e mesmo curso (ou curso de mesma área do conhecimento) consegue promover a mediação melhor que o professor. A atividade da monitoria é supervisionada pelo professor. Eventualmente, delega-se ao monitor a organização das discussões que ocorrem na lista e no fórum. Esta prática aproxima o monitor da realidade da disciplina, fazendo-o sentir-se parte da turma. Neste caso usamos a estratégia da monitoria virtual.

Ou seja, foi criado um fórum especial para que os estudantes se comunicassem com os monitores extraclasse em horários variados. Cabe salientar que alguns dos monitores fizeram uso do Skype para orientar os estudantes. Obtivemos permissão especial para realizar este tipo de atividade uma vez que a cultura da universidade com relação à monitoria é a presencialidade. Monitores ficam no laboratório em horários previamente agendados. Observou-se que, na prática, isto não funciona devido a restrições de horários em ambas as partes. A opção da monitoria virtual foi muito bem aceita. Os monitores também foram estimulados a serem autores e muitas das videoaulas disponibilizadas no YouTube foram feitas por eles.

Além de incentivar os estudantes a realizarem uma leitura crítica dos enunciados propostos nos exercícios, estimula-se que realizem a leitura prévia dos materiais de aula e da leitura recomendada. Essa prática tem como objetivo contribuir para a realização de uma efetiva aula dialogada, onde os estudantes participam, questionando o professor e interagindo ao serem questionados. Em relação às avaliações realizadas, busca-se que o estudante demonstre o estágio em que se encontra no processo de aquisição e construção do conhecimento. Desta maneira, a avaliação consiste na realização de três provas presenciais e diversos trabalhos práticos realizados ao longo do semestre.

3. Os exercícios diferenciados

Além da organização do espaço virtual que funcionou como ponto de apoio à interlocução para os docentes, discentes e monitoria, um fator significativo para contribuir para motivação para estudar foram os exercícios com enunciados diferenciados.

Optamos por usar exercícios temáticos, abordando nos enunciados séries de televisão, filmes, super-heróis e produtos da moda como elementos para criar exercícios que trouxessem algum tipo de identificação com seus dia-a-dia. O principal objetivo destes exercícios, com este tipo de abordagem, é aproximar os hábitos de lazer e escolhas pessoais dos estudantes como elementos auxiliares na motivação dos estudantes. Observamos que as tarefas muitas vezes acabam sendo percebidas como uma espécie de jogo RPG¹³, onde os enunciados são percebidos pelos estudantes como se fossem roteiros a serem seguidos com desafios em diferentes graus de complexidade.

A 2 apresenta um recorte do enunciado que criamos para trabalhar testes e estrutura de repetição na qual foi utilizado como mote o sistema de pulseira configuráveis de berloques. O diferencial desta abordagem encontra-se no fato de ser algo “palpável”, presente no dia-a-dia dos estudantes ou de pessoas de suas famílias.

¹³ *Role-Playing Games* (jogo de interpretação de personagens). É um tipo de jogo no qual os jogadores desempenham o papel de um personagem em um cenário fictício.

Pontifícia Universidade Católica do Rio Grande do Sul
Faculdade de Informática
Fundamentos de Programação

Exercícios de Reforço - Repetição

A moda entre as mulheres é usar colares personalizados de prata com pingentes de prata. Vide imagem meramente ilustrativa.



Estes colares são configuráveis. Isto é, você compra o cordão e depois decide quantos e quais serão os pingentes a serem colocados.

O cordão é ofertado em 2 comprimentos:

- 50 cm custa R\$ 300,00;
- 70 cm custa R\$400,00.

Os pingentes se dividem em 2 categorias:

- Prata custam R\$70 reais cada e,
- Prata com pedras incrustadas custam R\$100 reais cada.

Os pingentes podem ser colocados pendurados num gancho que custa R\$200,00 reais ou numa caixa redonda que custa R\$350,00 reais. Vide foto acima.



A empresa Bichara deseja que você faça um programa que auxilie os clientes a montarem seu colar e possam simular seu custo final considerando as possibilidades de configuração ofertadas pela empresa.

Os clientes podem pagar à vista com 5% de desconto. Ou fazer em até 3 vezes no cartão pelo preço do orçamento. Caso deseje fazer em até 5 vezes o custo acresce de 10% ao preço orçado. Você deve

decidir que dados são fundamentais de serem pedidos ao cliente para que seu programa funcione como desejado.

O prazo para entrega desta encomenda é dia 2 de maio, 16:00 horas.

Características desta 1ª versão:

- A pessoa compra 1 colar apenas;
- Não existe limite de peças para comprar (considera-se que ela pode querer comprar mais adereços para poder variar e ir trocando). Ou seja, o tamanho do gancho ou caixa redonda não é limitante para a compra;
- A saída do programa apresenta o valor inicial da, a descrição dos itens comprados (tipo e quantidade) e o valor final após a escolha da condição de pagamento. No caso de prestações escreva a quantidade de prestações e o valor de cada prestação.

Existe uma sala de entrega de tarefas especialmente criada para isto denominada **Bichara system 1.0**

Seu código Java deve ser comentado destacando o que cada parte faz e incluir seus dados tais como nome, número de matrícula e turma e nome da professora.

O arquivo do projeto deve ser zipado e colocar como identificação do arquivo a 1ª letra do nome seguida do seu sobrenome.
Exemplo: xfulano-versao1.zip ou xfulano-versao1.rar

Figura 2: Enunciado para teste estruturas de seleção

Outro exemplo abordado, utilizou como tema a série Game of Thrones. O uso deste exemplo foi bem interessante, uma vez que a cada semana, após um novo episódio, o conteúdo exibido pelo programa era alvo de comentários entre os estudantes durante o dia seguinte a exibição do episódio (dia este, no qual a referida disciplina era lecionada). Desta forma, era disponibilizado um enunciado que fazia com que fatos do episódio corrente fossem transformados em exercícios. Um exemplo disto pode ser visto na Figura . Observando a figura o leitor pode observar que incluímos um e-mail fictício de Danyeres para turma encomendando programa. Esta estratégia de criar um e-mail que é enviado para professora e com as tarefas solicitadas gerou uma motivação extra.

Fundamentos de Programação
Exercícios com Iterações

Finalmente sabemos o dia em que o destino de Jon Snow será revelado. A HBO utilizou a conta oficial de "Game of Thrones" no Twitter para anunciar que o sexto ano da série estreia no dia 24 de abril nos EUA. A emissora ainda não revelou a data de estreia no Brasil, mas as últimas temporadas estrearam simultaneamente nos dois países (e em muitos outros).



Os Dragões de Daenerys Targaryen (Dany para os fãs)



São elementos-chave nesta trama fantástica (no sentido de irreal) ambientada nos 7 reinos onde conquistar o Trono de Ferro é a meta dos diversos herdeiros das famílias que disputam ambiciosamente o poder. Os dragões de Dany receberam seus nomes em homenagem a três pessoas:



Rhaegal em homenagem a Rhaegar Targaryen, irmão que Dany não chegou a conhecer, mas que ela admira pelas histórias que ouviu sobre ele. Rhaegal é um dragão verde, com detalhes em bronze nas escamas, asas e olhos, sua chama é laranja e vermelha.



Dragon em homenagem a Drogo, seu falecido marido e líder dos Dothrakis, Dragon é o maior dos três dragões, é feroz, agressivo e sua chama negra e vermelha tem maior alcance, ele é negro com detalhes em vermelho nos olhos, oitões e até seu sangue também é negro.



Viseerion em homenagem a Viseerion Targaryen, o outro irmão de Dany que cresceu junto a ela, mas foi morto pelo seu marido depois de ameaçar a morte grávida, Viseerion é o dragão branco com detalhes dourados até nos olhos, ele tem chamas laranjas e amarelas.

No episódio de início de 2ª temporada nos deparamos com a notícia de que os dragões estão crescendo e um deles, Dragon, está foragido. Dany preocupada com isto deu-se conta que os dragões... quando soltos... poderiam trazer problemas para o reino. Lembrando que a turma 128 de Fundamentos de Computação do Curso de Bacharelado em Ciências da Computação da FACIN possui hobby: programadores (as), ele envia um e-mail à Profe. Giraffe solicitando o seguinte programa:

```
*****
De: daenerys_targaryen@staport.westnet.us
Enviado em: terça-feira, 12 de abril de 2016 21:06
Para: Lucia Maria Martins Giraffe <giraffe@pucrs.br>
Assunto: encomenda de programa

Prezada Profa. Lucia,
Preciso que sua competente turma de alunos (as) faça um programa para
verificar a poder de fogo que meus dragões possuem para saber se devo ou não
disparar o alerta da "Ameaça de Dragão" no meu reino.
Formeço-lhe as seguintes informações:
1. Cada dragão nasceu com um peso diferente:
    * Rhaegal nasceu primeiro;
    * Viseerion nasceu em segundo e possui peso 20% a mais que Rhaegal
    * Dragon é o maior e possui peso 30% maior que Viseerion
2. Os dragões crescem proporcionais ao seu peso. Ou seja, quanto mais se alimentam
maior é seu peso. Os dragões inicialmente comem x quilos que corresponde a 3%
de alimento por dia considerando seu peso atual!
3. A cada 300kg o poder de cuspir fogo aumenta. Ou seja, até ter 300kg os dragões
são inofensivos. A partir de 300kg os dragões começam a queimar metros
quadrados. Na seguinte regra:
    * A cada 300kg ele pode queimar 10m². E isto ocorre incrementalmente e
cumulativamente. Ou seja, seu poder de cuspir fogo e cresce incrementalmente e
aumenta a área queimada.
```

Figura 3: Enunciados relacionados a série GoT

De forma geral, o principal diferencial dos exercícios está no fato de abordarem temáticas simples, porém usando enunciados rebuscados, tornando o exercício diferente e divertido para o estudante. Além disso, por ser um enunciado mais rebuscado, no qual acrescentávamos tais elementos lúdicos, não era possível aos estudantes encontrarem uma solução equivalente na internet.

Em outro exemplo da nossa abordagem, durante o ensino do conteúdo sobre algoritmos de repetição, os exercícios mais comuns encontrados nas literaturas envolvem somatórios e produtórios, com cálculos acumulativos, para os quais o resultado anterior, deve ser usado para o cálculo posterior. Em uma destas abordagens utilizamos enunciado em que, com cálculos acumulativos, os estudantes precisavam verificar os níveis de poder do Super-Homem, onde, a cada dia, seus poderes aumentavam em relação ao poder que o mesmo possuía no dia anterior. Focando nesta mesma temática, também abordamos exercícios com o tema “apocalipse zumbi”, no qual os estudantes deveriam criar um programa que ajudasse a calcular o tempo de infestação de uma doença que estava transformando os habitantes de uma cidade de zumbis. Ainda no mesmo tema, durante a época de Páscoa, os estudantes também receberam como missão construir um programa para calcular o consumo total de chocolate feito por um coelho “chocolatra”, onde a cada dia ele consumia um percentual de seu peso, e o consumo fazia com que seu peso aumentasse. Como pode ser visto, todos eles tratam do mesmo tipo de problema e de cálculo, e com somente uma reconstrução do enunciado é possível abordar o problema de múltiplas (e divertidas) maneiras.

O filme Vingadores também foi utilizado para criação de um enunciado como mostra a 4. O exercício foi elaborado de forma a abordar diversos conceitos, tais como, algoritmos de ordenação, orientação a objetos e manipulação de vetores. Durante os dias nos quais os estudantes deveriam solucionar o trabalho era visto um engajamento acima

da média. Todos os trabalhos chegariam a um mesmo resultado final, após a execução do algoritmo, o que tornou feio com que os alunos o encarassem como uma espécie de quebra-cabeças.

Manipulação de objetos e arrays

Ultron está de volta! E agora mais poderoso!



O vilão que assombrou os Vingadores voltou e está determinado a destruir seus inimigos e toda a humanidade!

Você foi contratado pela **S.H.I.E.L.D.** para ajudar a desenvolver um sistema que irá gerar informações cruciais para a batalha que pode salvar ou condenar a humanidade. Sabemos que cada poderoso (seja ele vilão ou herói) possui um nome e um nível de poder. Desta forma, construa uma classe chamada **Poderoso**, que carregue esses dois atributos (não se esqueça do construtor, **getter** e **setter**).

Construa uma classe chamada **CadastroHeróis**. Nesta classe implemente um array de objetos do tipo Poderoso. Não esqueça de criar um método **getter** que retorne o array.

Crie métodos para adicionar, pesquisar e remover heróis do array. Crie também métodos para mostrar todos os heróis cadastrados em ordem alfabética, todos os heróis cadastrados por ordem decrescente de poder, e o herói mais poderoso.

Crie também na classe **CadastroHeróis** o método **simularBatalha**. Este método deve receber por parâmetro um objeto Poderoso, para o vilão **Ultron**.

Então... VAMOS COMEÇAR A BATALHA!!!

Crie neste método um algoritmo que faça a simulação e imprima a ordem na qual os ataques a **Ultron** devem ser feitos para que ele seja vencido. Mas existem algumas regras:

- Se você atacar **Ultron** com uma força superior a força



Figura 4: Enunciado relacionados ao filme Vingadores

Cabe salientar que esta abordagem foi mantida nas questões da prova onde se buscou trazer o caráter lúdico aos enunciados. Uma das provas realizadas teve sua data coincidindo com a semana de estreia do filme Jurassic World. Procuramos assim, realizar uma prova com esta temática, onde os estudantes deveriam responder diversas questões sobre um programa para gerenciamento de um parque dos dinossauros, como pode ser visto na Figura .




Relatórios OBRIGATÓRIOS:

Relatório Quantidade de animais de cada tipo e categoria (2.0 pts):
O método deve retornar uma String na qual devem aparecer as informações referentes a quantidade de animais para cada tipo e categoria. Exemplo:
Carnívoros: PP: 52, MP: 20, GP: 12. Herbívoros: PP: 68, MP: 38, GP: 30.

Relatório Peso pesado (2.0 pts):
Escreva um método que receba por parâmetro o tipo e categoria de um dinossauro e retorne o dinossauro (objeto) mais pesado do tipo e a categoria.




Relatórios EXTRAS: Escolha 2 dos 3 para fazer.

Relatório Quantidade de carne para carnívoros (3.0 pts):
Cada carnívoro come um valor relacionado ao seu peso e de acordo com sua categoria.
Raças de pequeno porte comem cerca de 10% de seu peso por dia.
Raças de médio porte comem cerca de 15% de seu peso por dia.
Raças de grande porte comem cerca de 20% de seu peso por dia.
O método deve retornar à quantidade (em Kg) de carne que deve ser comprado no mês (considere o mês por 30 dias).

Relatório Dá tempo de fugir? (3.0 pts):
Esse relatório, devido aos últimos acidentes, é considerado de extrema importância.
Considera-se que, para corridas curtas em situação de extremo desespero, uma pessoa consiga correr em média a

Figura 5: Enunciado baseado no filme Jurassic World

Em outra avaliação, trouxemos o problema de alocação e gerenciamento de figurantes para a série Game Of Thrones, uma vez que a série necessita de um alto número de figurantes a cada episódio. Os exercícios utilizavam conceitos como orientação a objetos e composição de objetos, além de repetição e manipulação de vetores para suas resoluções. Neste caso, o caráter lúdico do enunciado não é tão forte como nos exercícios anteriores, porém, aborda um problema que pode ser enfrentado por um programador na vida real (alocação de funcionários, alocação de times e etc).

Os enunciados baseados em atividades e lazer trouxeram uma contribuição extra no que tange ao esforço para abstrair os elementos fundamentais para realização do exercício. Outro ponto importante a destacar é que o uso de exercícios deste gênero exige dos estudantes interpretação de textos, uma vez que precisam compreender o enunciado para, a partir daí, abstrair os conceitos de programação que devem ser utilizados para cada caso. Durante esta etapa da interpretação o professor realizava com os estudantes uma “desconstrução” do enunciado, tentando com isso auxiliá-los a preencher essa lacuna. Outro ponto cobrado dos estudantes era o uso de comentários e/ou documentação escrita dentro do próprio código, que além de auxiliá-los na construção de textos, os engajava em um processo de reflexão e revisão dos programas criados.

4. Considerações finais

A experiência com uso destes espaços virtuais como elementos de extensão e mediação das atividades da sala de aula presencial resultou em vários pontos positivos na condução da disciplina e no aproveitamento dos estudantes. O mais significativo em termos de indicador foi o comportamento dos estudantes com relação aos exercícios e o trabalho de acompanhamento. Destacamos alguns pontos mencionados pelos estudantes:

- A forma como o professor conduziu a disciplina incentivando-os a ler, pensar e realizar exercícios;
- O tipo de exercícios e atividades disponibilizados no ambiente e respectivos feedbacks;
- A organização do espaço virtual que lhes permitiu compartilhar com colegas soluções e dúvidas comuns;
- Um sistema de avaliação compatível com os trabalhos realizados em aula;
- O comportamento do professor com relação aos estudantes: respeitoso e incentivador.

Evidentemente, uma proposta metodológica deve ser refinada a cada semestre no que tange aos aspectos operacionais (relativos ao uso dos recursos de apoio), distribuição das informações sobre a disciplina e outros. Os pontos onde geralmente são feitas as maiores alterações são:

- Apresentação de um cronograma geral de atividades para todo o semestre. Para o estudante ter ideia do volume de trabalho (leituras, exercícios, pesquisas, tarefas práticas, etc.) a ser realizado;
- Disponibilização dos materiais de forma antecipada (no mínimo uma aula à frente);
- Adequação da tecnologia associada aos recursos para encontros síncronos e assíncronos. Algumas delas podem se mostrar inadequadas ou obsoletas. Outras podem ter problemas com os provedores privados utilizados pelos estudantes fora da universidade, como no caso dos vídeos;

Pode-se ressaltar como aspectos positivos desta metodologia o envolvimento dos estudantes nas atividades de aula presencial e sua participação dos estudantes enviando materiais e soluções para serem compartilhadas com os colegas. Em contrapartida, considerar como aspecto “negativo” da mesma o enorme volume de informações que o professor e o estudante têm de gerenciar, pode-se citar como exemplo a grande quantidade postagens no fórum e documentos que o professor precisa verificar semanalmente. Neste sentido adotou-se o uso de ferramentas de Learning Analytics (LA), disponíveis no Moodle versão 3.0.2. Maiores detalhes podem ser obtidos em Giraffa (2015).

A escolha dos exercícios e atividades deve ser diversificada e com enunciados que permitam a aplicação direta de um conceito e sim que permita o estabelecimento relações com o aprendido e a bagagem que o estudante possui, considerando suas limitações e expectativas.

Cada vez mais os estudantes chegam à universidade com deficiências na sua formação. Deficiências estas relacionadas à expressão em língua materna (escrita), interpretação de textos e enunciados, hábitos de estudo e pesquisa e, especialmente, com formação deficitária no que tange a conteúdos de Matemática. Não adianta apenas o professor constatar isto e dizer que os estudantes já vieram com os referidos déficits. O fato é que após passarem nos exames seletivos eles se tornam universitários e passam a ser responsabilidade da universidade e, por consequência, dos seus professores. Cabe aos docentes e gestores buscarem soluções para transpor estas lacunas no aprendizado progresso destes estudantes.

Auxiliá-los a se motivarem para transpor estas limitações e lacunas é uma tarefa importante a ser desempenhada pelo professor. Atualmente o uso da Internet e seus recursos são cada vez mais difundidos e faz parte do cotidiano dos jovens. Competir com as redes sociais e os jogos online são outros desafios do docente no século XXI que somente poderá ser superado com outro entendimento do que é ser professor neste novo contexto hipermediático e com pouca profundidade discussões (Giraffa e Marczak, 2012).

Agradecimentos

Agradecemos a Coordenação do Curso por acreditar no trabalho que propusemos e nos dar espaço para concretiza-lo, aos monitores da disciplina que nos acompanharam nestes semestres, especial agradecimento a Raphael Varela Ribeiro que elaborou vídeos de apoio, a colega Milene Silveira que entrou no espírito dos enunciados relacionados a filmes e series e criou uma questão de prova relacionada aos filmes Harry Potter, e a colega Marcia Moraes que nos acompanhou no primeiro semestre do curso.

Referências

- BECKER, F. (2001). Educação e construção do conhecimento. Porto Alegre: Artmed.
- BENNEDSEN, J. B., & CASPERSEN, M. E. (2007). Failure Rates in Introductory Programming, SIGCSE Bulletin inroads, 39 (2), 32-36.
- BENNEDSEN, J. B., & CASPERSEN, M. E. (2009, October). Recalling Programming Competence. Proceedings of the 9th International Conference on Computing Education Research, Koli, Finland, 09.
- DELGADO, C., XEXEO, J. A. M., SOUZA, I. F., RAPKIEWICZ, C. E., & PEREIRA JÚNIOR, J. C. (2005). Identificando competências associadas ao aprendizado de leitura e construção de algoritmos. Anais do XXV Congresso da SBC, São Leopoldo, RS, Brasil, 25.
- GIRAFFA, L. M. M., & MORAES, M. C. (2012). O desafio de ensinar a programar no primeiro nível em cursos de graduação: alternativas para conter a evasão. In Segunda Conferencia Latinoamericana sobre El Abandono en La Educación Superior II CLABES. Porto Alegre (Brazil). ALFAGUIA (CLABES) (Vol. 1, pp. 486-498).
- GIRAFFA, L. M. M., & FARIA, E. T. (2012). Using Podcasts to Better Qualify the Teaching Evaluation Process in Virtual Classes. Proceedings of 6th International Technology, Education and Development Conference, Valencia, Spain, 06.
- GIRAFFA, L. M. M., & MARCZAK, S. S. (2012). Being a digital teacher: myths, dilemma and challenges for 21st century teachers. Proceedings of 4th annual International Conference on Education and New Learning Technologies, Barcelona, Spain, 04.
- GIRAFFA, L. M. M., & MORA, M. D. C. (2013). Evasão na Disciplina de Algoritmo e Programação: Um Estudo a Partir dos Fatores Intervenientes na Perspectiva do Aluno. In Tercera Conferencia sobre el Abandono en la Educación Superior (III CLABES), México.
- GIRAFFA, L. M., MORAES, M. C., & UDEN, L. (2014). Teaching Object-Oriented Programming in First-Year Undergraduate Courses Supported By Virtual Classrooms. In The 2nd International Workshop on Learning Technology for Education in Cloud (pp. 15-26). Springer Netherlands.
- GIRAFFA, L. M. (2015). RECURSOS DE LEARNING ANALYTICS PARA COMPOR INDICADORES AUXILIARES NA AVALIAÇÃO DOS ESTUDANTES. Revista EmRede, 2(2), 32-43.
- GIRAFFA, L. M. M. (2015). Orientação virtual: expandindo o espaço de orientação em tempos de comunicação ubíqua. Educação Por Escrito, 6(2), 194-207.

PORTAL, L. L. F. (2003). Até que ponto a informática é uma ferramenta para intervenção educativa? *Mediação Tecnológica: construindo e inovando* (pp. 125-136). Porto Alegre: EDIPUCRS.

RAABE, A. L. A., & ILVA, J. (2005). Um ambiente para atendimento as dificuldades de aprendizagem de algoritmos. *Anais do XXV Congresso da Sociedade Brasileira de Computação*, São Leopoldo, RS, Brasil, 25.

RAABE, A. L. A., & GIRAFFA, L. M. M. (2006). Uma Arquitetura de Tutor para Promover Experiências de Aprendizagem Mediadas. *Anais do XVII Simpósio Brasileiro de Informática na Educação*, Brasília, DF, Brasil, 17.

RAABE, A. L. A., DAZZI, R. L. S., & SANTIAGO, R. (2007). Adquirindo experiência na construção de ferramentas de apoio a aprendizagem de algoritmos. *Anais do XVIII Simpósio Brasileiro de Informática na Educação*, São Paulo, SP, Brasil, 18.

TELLO, Steven F. An analysis of student persistence in online education. In: *International Journal of Information and Communication Technology Education*, 3(3), 47–62, 2007.

TINTO, V. Classrooms as communities: Exploring the educational character of student persistence. *The Journal of Higher Education*, v. 68, n. 6, pp. 599-623. 1997.