

Um minicurso *online* de Algoritmos como apoio às disciplinas iniciais da graduação: preparação, execução e resultados sobre a satisfação dos alunos

Barbara Moissa¹, Lucio Vasconcelos dos Santos², Ana Carolina Tomé Klock¹,
Isabela Gasparini^{1,2}, Avanilde Kemczinski^{1,2}

¹PPGCA e ²PPGECMT – ^{1,2}Universidade do Estado de Santa Catarina (UDESC)

{barbara.moissa, vasconcellos.lucio, actklock}@gmail.com,
{isabela.gasparini, avanilde.kemczinski}@udesc.br

Abstract. *The disciplines of Programming Logic and Algorithms are inherent to several courses of exact sciences, such as Computer Science, Technologies, Mathematics, Civil, Mechanical, Production Engineering, etc. However, learning such contents is a hard task. In order to develop a quality material to teach Algorithms, the ADDIE model was used. This paper describes the process of planning, execution and evaluation of a short course, followed by the result of the students' satisfaction evaluation.*

Resumo. *As disciplinas de Lógica de Programação e Algoritmos são inerentes a diversos cursos da área de exatas, como os relacionados à Ciência da Computação, Tecnologias, Matemática e Engenharias Civil, Mecânica, de Produção, etc. Entretanto, aprender tais conteúdos é uma tarefa difícil. Com o objetivo de elaborar um material de qualidade para ensinar Algoritmos, o modelo ADDIE foi utilizado. Este artigo descreve o processo de planejamento, realização e avaliação do minicurso, seguido do resultado da avaliação de satisfação dos alunos.*

1. Introdução

Atualmente a formação superior de diferentes cursos da área de exatas se preocupa em oferecer, mesmo que em disciplinas iniciais, disciplinas de Algoritmos e Lógica de Programação. De acordo com Delgado et al. (2005) e De Castro et al. (2003), neste primeiro contato com o ensino de Algoritmos e Programação, os estudantes encontram um grande obstáculo na aplicação de suas habilidades anteriores, gerando frustração e ocasionando reprovações, desistências da disciplina ou até abandono do curso. Segundo Castro et al. (2003), “o ensino de programação é desafiador porque a própria tarefa de programar não é fácil”.

Desta forma, é importante buscar novas estratégias para apoiar o estudante no processo de ensino-aprendizagem da lógica de programação. Forbellone e Eberspächer (2000) definem a lógica de programação como a ordenação dos pensamentos e dos processos de raciocínio na produção de soluções válidas e coerentes para problemas a serem resolvidos por meio de programas. Segundo os autores, o raciocínio é algo abstrato e intangível e, para expressá-lo, o ser humano precisa utilizar uma linguagem como, por exemplo, uma linguagem de programação. Entretanto, o uso de linguagens de programação dificulta o entendimento do raciocínio lógico devido à diversidade de

detalhes computacionais [Forbellone e Eberspächer, 2000]. Para representar o raciocínio da lógica de programação, utilizam-se os algoritmos, que são formalmente definidos como “uma sequência de passos que visam atingir um objetivo bem definido” [Forbellone e Eberspächer, 2000]. Os algoritmos podem ser representados de diversas formas, como por meio de pseudocódigo, fluxogramas e narrativas.

As disciplinas relacionadas a Algoritmos são consideradas essenciais na formação dos profissionais de Computação e são as disciplinas que apresentam um nível de dificuldade maior durante o aprendizado [Xavier et al., 2004]. Zanetti e Bonacin (2014) afirmam que a aprendizagem da programação é um processo difícil devido à necessidade de um alto nível de abstração e do entendimento de alguma linguagem de programação. Os alunos iniciantes não possuem habilidades e competências relacionadas ao raciocínio lógico e à resolução de problemas, tendo dificuldades na utilização do ambiente de desenvolvimento, na compreensão das estruturas de programação, no entendimento da sintaxe, na divisão das funcionalidades em procedimentos e na localização dos erros no programa [Gomes et al., 2008]; além disso, os métodos pedagógicos apresentados são inadequados [Lahtinen et al., 2005]. Apesar dos professores buscarem a aplicação de técnicas inovadoras, percebe-se o desinteresse pelo curso, a desmotivação para o estudo de programação, as reprovações sistemáticas, a apatia, a baixa autoestima, a desistência da disciplina e até o abandono do curso [Delgado et al., 2005].

Tendo em vista a importância de se desenvolver as competências necessárias para escrever Algoritmos de forma adequada e eficaz, é importante buscar soluções educacionais para atender a esta necessidade. Isto requer ferramentas e meios adequados para que os objetivos de aprendizagem sejam atingidos. Para isso, diversas estratégias e tecnologias de apoio podem ser adotadas, dentre elas, o *Design Instrucional* (DI). O DI envolve o planejamento, o desenvolvimento e a aplicação intencional e sistemática de situações didáticas específicas com o intuito de promover a aprendizagem [Filatro, 2008], sendo visto por muitos autores como a produção de soluções educacionais para atender necessidades de aprendizagem de certos grupos de pessoas [Kenski, 2015]. Quando utilizado em Ambientes Virtuais de Aprendizagem (AVAs), o DI é entendido como “o planejamento do ensino-aprendizagem, incluindo atividades, estratégias, sistemas de avaliação, métodos e materiais instrucionais” [Filatro e Piconez, 2004]. O processo de DI consiste em uma série de passos para a produção de materiais de alta qualidade [Crawford, 2004]. Entre os diversos modelos existentes para o desenvolvimento do DI, podem ser citados o modelo 4c-ID [Merriënboer et al., 2002], o modelo ASSURE (Heinich et al., 2002), o modelo de Dick e Carey (2001) e o modelo ADDIE [Mustaro et al., 2007], sendo esse último amplamente encontrado e utilizado na literatura.

Com base no exposto, foi elaborado um Minicurso de Algoritmos, de modo a oferecer suporte às disciplinas presenciais de Algoritmos e/ou Linguagens de Programação de oito cursos da Graduação do Centro de Ciências e Tecnologias (CCT) da Universidade do Estado de Santa Catarina (UDESC), a saber: Ciência da Computação, Tecnologia em Análise e Desenvolvimento de Sistemas, Matemática, Física, e Engenharias Civil, Mecânica, de Produção e Elétrica. Este minicurso foi desenvolvido com apoio do processo de DI e com base no modelo ADDIE. O objetivo deste artigo é mostrar que o DI auxilia na produção de soluções educacionais de

qualidade, bem como mostrar a influência deste processo na satisfação dos alunos que o utilizaram. Esse artigo foi estruturado da seguinte forma: a Seção 2 apresenta o modelo ADDIE, o ambiente AdaptWeb[®] e suas ferramentas de suporte; a Seção 3 descreve a concepção e execução do Minicurso; e a Seção 4 analisa a satisfação dos alunos em relação ao minicurso. Por fim, na Seção 5 são apresentadas as considerações finais do trabalho.

2. Fundamentação

Esta seção descreve o modelo ADDIE utilizado para elaborar o minicurso de Algoritmos, o ambiente AdaptWeb[®] e suas ferramentas de suporte ao aluno e ao professor.

2.1. Modelo ADDIE

O modelo ADDIE (acrônimo para *Analysis, Design, Development, Implementation and Evaluation*) é um modelo de DI [Mustaro et al., 2007]. Este modelo é dividido entre duas fases: a fase de concepção, que envolve as etapas de Análise (*Analysis*), Projeto (*Design*) e Desenvolvimento (*Development*); e a fase de execução, que envolve as etapas de Implementação (*Implementation*) e Avaliação (*Evaluation*) [Filatro, 2008].

Na etapa de **análise**, o problema educacional é compreendido e uma solução aproximada é projetada [Filatro, 2008]. Também são levantadas as necessidades educacionais, realizada a caracterização do público-alvo e definidos os objetivos instrucionais e restrições. Na etapa de **projeto** é realizado o planejamento da situação didática propriamente dita, podendo ser expressa por meio da Matriz de *Design* Instrucional (MDI), que contempla os tópicos, conteúdos, exercícios, mídias e o planejamento de aula [Filatro, 2008]. Os materiais didáticos desenvolvidos durante esta etapa são inseridos e adaptados no AVA durante a etapa de **desenvolvimento** [Filatro, 2008], juntamente com as últimas correções necessárias antes da etapa de implementação [Mayfield, 2011]. Na etapa de **implementação** ocorre à capacitação dos professores e alunos, além da situação didática propriamente dita, quando ocorre a aplicação da proposta de DI [Filatro, 2008]. Por último, a etapa de **avaliação** verifica se os objetivos instrucionais foram alcançados, podendo ser de forma diagnóstica (nível de conhecimento, habilidades e necessidades), somativa (*feedback* dos usuários) ou formativa (contínua e presente em cada etapa do processo) [Filatro, 2008].

2.2. AdaptWeb[®]

O AdaptWeb[®] (acrônimo para Ambiente de Ensino-Aprendizagem Adaptativo na *Web*) é um sistema hipermídia adaptativo que permite a adaptação da navegação, da apresentação e do conteúdo conforme o perfil do aluno [Gasparini et al., 2009a]. Ou seja, o ambiente armazena alguns dados do aluno (e.g., curso, conhecimento, preferências e histórico navegacional) e adequa o conteúdo, a apresentação e a navegação ao perfil de cada aluno.

Através do módulo de autoria deste ambiente, o professor pode organizar os materiais instrucionais em tópicos hierárquicos. Para cada um dos tópicos, o material pode ser incluso na forma de conceito (um para cada tópico), exemplos, exercícios e materiais complementares. Neste módulo, o professor pode organizar os materiais instrucionais uma única vez para uma disciplina ministrada em dois ou mais cursos, definindo quais materiais estão disponíveis para cada um deles. O professor também

pode criar avaliações e analisar a participação de seus alunos. O aluno, por sua vez, acessa a disciplina através do Ambiente da Disciplina, em que tem acesso ao Ambiente de Aula (i.e., onde se pode ver todo o material instrucional destinado àquela disciplina), ao Mural de Recados, ao Fórum de Discussão e à ferramenta de Análises de Aprendizagem.

2.3. Ferramentas de Suporte ao Aluno e ao Professor

O AdaptWeb[®] possui uma ferramenta de *Learning Analytics* integrada. Esta ferramenta coleta os dados de navegação dos alunos e os analisa através de técnicas estatísticas para fornecer informações de modo visual aos professores e aos alunos [Moissa et al., 2014]. O professor acessa essas informações através da ferramenta de Análise Interativa disponível no módulo de Autoria, cuja interface é exibida na Figura 1.

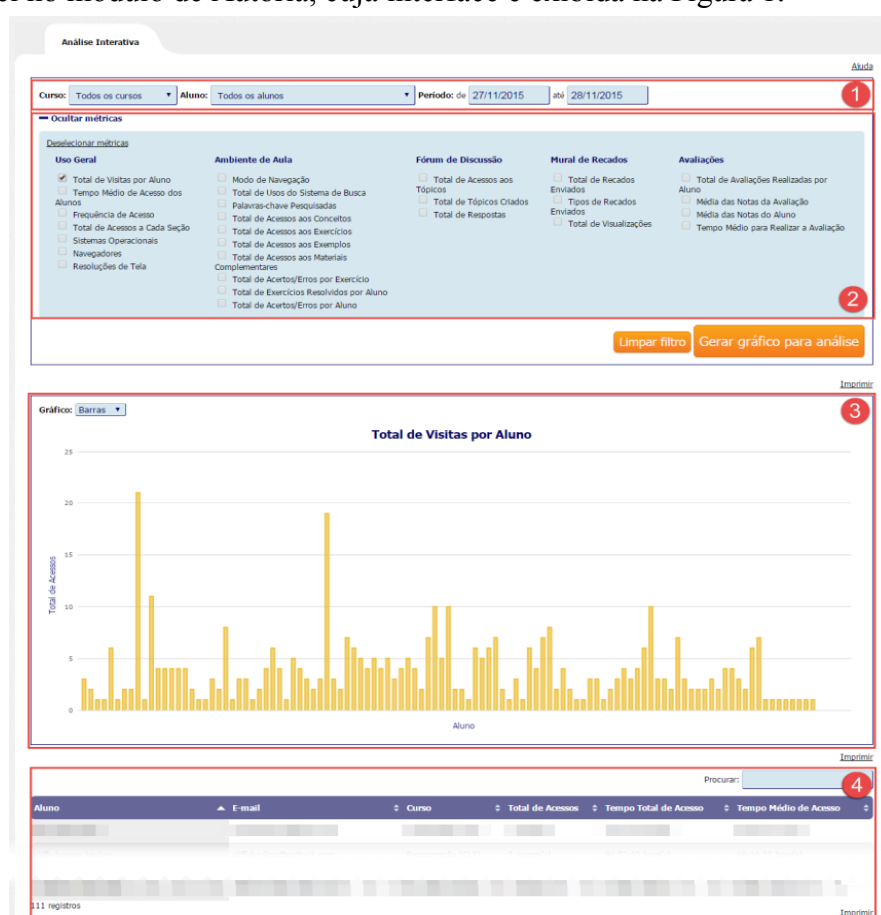


Figura 1. Parte da ferramenta Análise Interativa – Visão do Professor

A interface possui quatro áreas principais: 1) Filtro; 2) Seleção de Métricas; 3) Visualização; e 4) Informações Detalhadas. Na área 1, o professor filtra as informações por curso, aluno e período de tempo e, na área 2, o professor seleciona qual métrica deseja analisar. As métricas disponíveis atualmente na ferramenta estão agrupadas em cinco categorias: (1) **Uso Geral**: Informações sobre o uso do ambiente AdaptWeb[®] como, por exemplo, Total de Visitas por Aluno, Tempo Médio de Acesso dos Alunos e Frequência de Acesso; (2) **Ambiente Aula**: Informações referentes ao uso dos materiais instrucionais como, por exemplo, Total de Acessos aos Conceitos, Total de Acessos aos Exemplos e Total de Acessos aos Materiais Complementares; (3) **Fórum de Discussão**:

Informações referentes ao uso do Fórum de Discussão como, por exemplo, Total de Acessos aos Tópicos, Total de Tópicos Criados e Total de Respostas; (4) **Mural de Recados**: Informações referentes ao uso do Mural de Recados como, por exemplo, Total de Recados Enviados, Tipos de Recados Enviados e Total de Visualizações; e (5) **Avaliações**: Informações sobre a realização de avaliações da disciplina como, por exemplo, Total de Avaliações Realizadas por Aluno, Média das Notas da Avaliação e Média das Notas do Aluno.

Esta ferramenta auxilia o professor a acompanhar as atividades dos alunos durante o período em que sua disciplina é ministrada, auxiliando-o a identificar, por exemplo, quais alunos não estão acessando o ambiente, quais alunos estão com notas baixas, quais alunos não estão fazendo os exercícios, em quais exercícios e tópicos os alunos possuem mais dificuldades, etc., e assim, intervir no processo de ensino-aprendizagem de forma a corrigir os problemas identificados. Por sua vez, os alunos podem analisar informações referentes ao seu desempenho e progresso através da ferramenta Análises de Aprendizagem, disponível no Ambiente da Disciplina. Uma parte da interface desta ferramenta está ilustrada na Figura 2, onde são exibidas diversas informações referentes ao progresso do aluno. O aluno pode visualizar seu progresso na disciplina (i.e., total de materiais instrucionais acessados), seu progresso na disciplina por tipo de recurso (i.e., total de materiais instrucionais acessados - que podem ser do tipo conceito, exemplo, exercício e material complementar), seu progresso na disciplina por tópico (i.e., total de materiais instrucionais acessados em cada tópico principal) e seu progresso em detalhes (i.e., total de materiais instrucionais acessados por tipo em cada tópico).

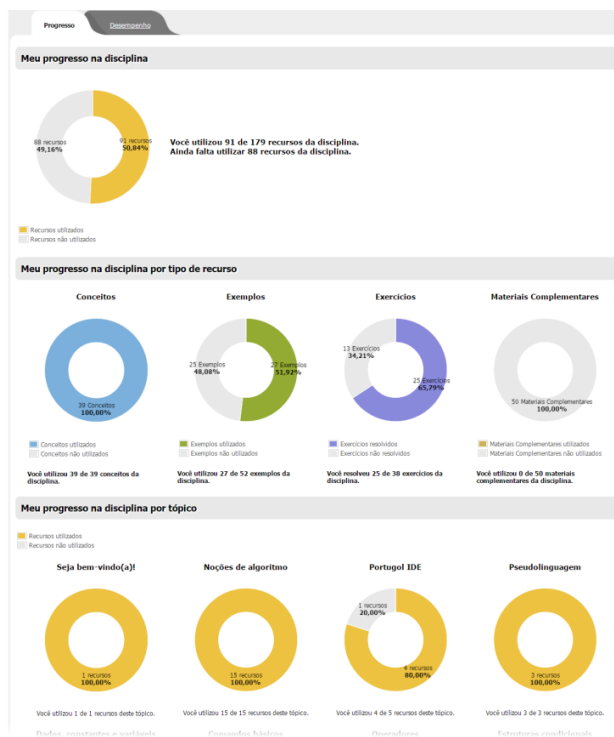


Figura 2. Parte da Ferramenta Análises de Aprendizagem – Visão do Aluno

O aluno também pode acompanhar seu desempenho visualizando informações como: seu desempenho nos exercícios (i.e., total de exercícios resolvidos e total de

exercícios resolvidos corretamente), os exercícios resolvidos pela turma, os exercícios resolvidos corretamente pela turma, o seu desempenho nas avaliações e o desempenho da turma nas avaliações. Através destas análises, o aluno pode verificar a quantidade de conceitos estudados, de exemplos acessados, de exercícios resolvidos e de materiais complementares estudados. O aluno também pode verificar a quantidade de exercícios que ele acertou e/ou errou, qual é o desempenho dos demais alunos da turma, como os demais alunos da turma estão estudando e, assim, identificar se está ou não atrasado em seus estudos e se seu desempenho está compatível com o desempenho do restante da turma.

3. Concepção e Execução do Minicurso

Como um dos problemas na aprendizagem de Algoritmos é a falta de um método pedagógico adequado [Lahtinen et al., 2005], o modelo ADDIE foi utilizado para definir os tópicos e materiais necessários em um Minicurso de Algoritmos de qualidade. O procedimento realizado é detalhado a seguir.

3.1. Análise

Em muitos cursos de graduação, tais como Matemática, Computação/Informática, Engenharia Mecânica, Engenharia de Produção, Física, etc., existem disciplinas com o propósito de ensinar a programação e suas aplicações. Porém, não são previstas as dificuldades prévias dos alunos em relação à abstração para a resolução de problemas e, muitas vezes, não há recursos tecnológicos ou não são aplicadas estratégias didáticas adequadas para atingir este propósito. Assim, o Design Instrucional como ferramenta de planejamento, execução e avaliação para o curso de Algoritmos, nos permitiu trabalhar os problemas educacionais supracitados e desenvolver as competências necessárias de programação para alunos de Ensino Superior.

Para resolver este problema sem interferir diretamente nas aulas presenciais das disciplinas de Algoritmos e Linguagem de Programação da universidade, optou-se por um Minicurso de Algoritmos totalmente *online* aplicado em paralelo à essas disciplinas. Este minicurso devia abordar todos os conceitos fundamentais para o desenvolvimento de um algoritmo, ter uma semana de duração e ser realizado durante o final do segundo semestre de 2015. Por se tratar de um minicurso para iniciantes, optou-se pela utilização de pseudolinguagem para abordar estes conteúdos, de forma que, ao final do minicurso, o aluno tivesse condições de prosseguir seu aprendizado na linguagem de programação de sua escolha. Outra vantagem da pseudolinguagem, é a eliminação da linguagem de programação que, de acordo com Forbellone e Ebërsbacher (2000) dificulta o entendimento do raciocínio lógico. Sem uma linguagem de programação, também elimina-se a barreira do idioma, já que a pseudolinguagem utiliza o idioma Português e não o Inglês (utilizado pelas linguagens de programação).

3.2. Projeto

Como parte dos conceitos fundamentais de Algoritmos, foram consideradas as definições de algoritmo e pseudolinguagem, comandos de entrada/saída de dados, tipos de dados, variáveis, constantes, operadores (lógicos, aritméticos e relacionais), estruturas de seleção, estruturas de decisão e laços de repetição. Considerando-se o uso da pseudolinguagem e a necessidade de visualização de funcionamento dos algoritmos, propõe-se o uso da ferramenta Portugol IDE, que permite implementar os principais

conceitos de programação de forma similar ao funcionamento de uma linguagem avançada [Portugol IDE, 2016]. Para realizar o planejamento adequado da situação didática, utilizou-se uma MDI. A Tabela 1 descreve uma parte desta MDI, que apresenta as informações relacionadas ao planejamento do minicurso. Para desenvolvê-la, foi considerada a estrutura do AVA AdaptWeb[®]. No qual é possível definir pré-requisitos para cada tópico; definir diferentes tipos de materiais, etc.

A coluna Objetivo define o que se pretende atingir, a coluna Conteúdo estrutura os tópicos necessários para atingir os objetivos listados. Também é possível definir os pré-requisitos de cada tópico, assim como definir os modos de entrega de cada tópico (i.e., tipos de material), apontadas as referências bibliográficas e estimado o tempo necessário para cada atividade, de forma a evitar que o minicurso exceda o tempo máximo estipulado.

Tabela 1. Trecho da MDI utilizada para organizar as informações do minicurso

Unidade Instrucional/Semana: 1. Introdução à Programação, Algoritmos e Portugol IDE.						
Data: 10/2015 Palavras-chave: algoritmo, pseudolinguagem, pseudocódigo, narrativa.						
Objetivo	Conteúdo	Pré-requisitos	Forma de entrega	Avaliação	Referências bibliográficas	Tempo
(...)						
Definir o que é um algoritmo, seus tipos e orientar como criá-los.	Conceito Nº 2 – Noções de algoritmo	Nenhum	Conteúdo	N/A	(Forbellone; Eberspächer, 2000)	3 a 5 min
Introduzir a ferramenta de pseudolinguagem, como instalar e começar a usar.	Conceito Nº 3 – Portugol IDE	Conceito nº 2	Conteúdo Exemplos	Exercícios	(Forbellone; Eberspächer, 2000)	3 a 5 min
Definir o que é pseudolinguagem e pseudocódigo.	Conceito Nº 4 – Pseudolinguagem	Conceito nº 3	Conteúdo Exemplos	Exercícios	(Forbellone; Eberspächer, 2000)	10 a 20 min
(...)						

3.3. Desenvolvimento

Para desenvolver o material do minicurso, foi estabelecido um grupo de três pessoas, com o apoio de uma professora de Algoritmos e Linguagem de Programação. Duas pessoas da equipe, ambos professores de Linguagem de Programação e mestrandos em Ensino e Tecnologia, ficaram responsáveis pela elaboração do conteúdo, exemplos e exercícios, enquanto a terceira pessoa, Mestre em Computação Aplicada, ficou responsável pelas questões técnicas (como a inclusão e formatação do conteúdo). Mesmo com a divisão de funções, a equipe trabalhou colaborativamente em todas as etapas. O conteúdo do minicurso foi desenvolvido seguindo a MDI descrita na etapa anterior.

Após a estruturação do conteúdo no ambiente AdaptWeb[®], foi realizado um teste piloto com alunos da graduação e pós-graduação que não conheciam o ambiente. O teste piloto tinha o intuito de avaliar os instrumentos do experimento: processo de matrícula no minicurso, Termo de Consentimento Livre e Esclarecido e o Questionário Satisfação e o próprio o ambiente. Os avaliadores seguiram um protocolo de teste e as opiniões dos participantes foram registradas. Algumas adequações foram realizadas no Questionário de Satisfação após o teste piloto.

3.4. Implementação

Inicialmente, alunos de turmas das disciplinas relacionadas a Algoritmos e Linguagem de Programação da Universidade do Estado de Santa Catarina foram convidados a participar do minicurso. A divulgação do minicurso, que contou com apoio dos professores destas disciplinas, foi realizada entre 21/11/2015 e 26/11/2015. O minicurso foi realizado entre os dias 27/11/2015 e 06/12/2015. Dos 114 inscritos, 88 participaram

do minicurso (acessaram pelo menos um conceito) e 71 concluíram o minicurso (acessaram pelo menos um conceito e realizaram a avaliação final). Antes de iniciarem o minicurso, todos os participantes concordaram com um Termo de Consentimento Livre e Esclarecido, onde foram explicados todos os benefícios, riscos, procedimentos, formas de participação e esclarecido que os participantes não serão identificados em nenhum momento.

3.5. Avaliação

Este trabalho utilizou tanto a avaliação formativa, quanto a avaliação somativa. A avaliação formativa é realizada por meio dos exercícios a cada tópico estudado pelo aluno, bem como sua participação no Fórum de Discussão, visto que o ambiente é preparado para tal [Gasparini et al., 2009b]. À medida que os alunos interagem no ambiente, eles são monitorados pela ferramenta de *Learning Analytics* (LA) (estes elementos nos fornecem subsídios para saber se o aluno está evoluindo em sua aprendizagem em relação aos objetivos definidos). Para avaliar se os objetivos de aprendizagem foram atingidos, foi realizada uma avaliação final (avaliação somativa) com 10 questões, sendo que estas questões abordavam todas as estruturas de programação apresentadas no minicurso. Os resultados mostram que a média dos alunos foi 8,3, sendo o desvio padrão 1,41, a menor nota 5 e a maior nota 10. Para verificar o grau de satisfação dos alunos com o minicurso executado, o quão fáceis de entender foram os materiais elaborados e o quão atrativos foram os materiais disponibilizados por meio do DI, um Questionário de Satisfação foi aplicado ao final do minicurso. As respostas foram analisadas e os resultados obtidos são detalhados na próxima seção.

4. Resultados sobre a Satisfação dos Alunos

O Questionário de Satisfação é composto de 8 questões, sendo que 6 questões utilizam a escala de Likert (com os seguintes níveis: discordo totalmente, discordo parcialmente, não discordo e nem concordo, concordo parcialmente e concordo totalmente e não sei opinar) e 2 questões são abertas. 62 alunos responderam a esse questionário. A Tabela 2 mostra um mapa de calor referente às 6 questões em escala de Likert. As áreas com a cor mais forte representam as respostas mais frequentes dos alunos e, as áreas com a cor mais fraca representam as respostas menos frequentes dos alunos. É importante ressaltar que os alunos também poderiam selecionar a resposta “Não sei opinar” e que essas foram desconsideradas na análise realizada.

Tabela 2. Resultados das questões em escala de Likert

	Discordo Totalmente	Discordo Parcialmente	Não Concordo e Nem Discordo	Concordo Parcialmente	Concordo Totalmente
O conteúdo foi claro e fácil de entender	0	0	1	13	46
A quantidade de exercícios foi satisfatória	0	1	4	16	39
A quantidade de exemplos foi satisfatória	0	2	6	14	37
A quantidade de materiais complementares foi satisfatória	0	2	7	9	36
A duração do minicurso (em 1 semana) foi adequada para o conteúdo apresentado	1	1	2	15	41
A avaliação de aprendizagem foi condizente com o conteúdo apresentado	0	1	5	8	45

Conforme pode ser observado na Tabela 2, o conteúdo instrucional desenvolvido de modo geral deixou os alunos satisfeitos. O ponto melhor avaliado foi a clareza do conteúdo. Na questão aberta “Cite os pontos positivos do conteúdo”, o

conteúdo foi avaliado como abrangente, sucinto, organizado e de fácil compreensão. Também foi considerado por alguns alunos como uma boa base para a programação. Já na questão aberta “Cite os pontos negativos do conteúdo”, foram mencionados como pontos negativos o curto período de duração do minicurso (uma semana), a falta de exercícios práticos (implementação), o baixo número de exercícios, conteúdo muito básico (deveria abranger mais conteúdos como vetores, matrizes, ponteiros, etc.) e a utilização apenas de pseudocódigo. Em relação aos exemplos, enquanto alguns alunos os acharam suficientes e bons, outros os acharam estes insuficientes e simples demais. Apesar da curta duração do minicurso ter sido citada como ponto negativo, grande parte dos alunos considerou a duração do minicurso adequada (conforme pode ser observado na Tabela 2). Também é possível reparar que as questões abertas confirmaram as respostas dadas às questões em escala de Likert.

5. Considerações Finais

O modelo de DI ADDIE foi adotado para auxiliar no desenvolvimento do material instrucional de um Minicurso de Algoritmos, planejado para alunos do primeiro período da graduação com duração de uma semana. Este minicurso foi realizado entre os dias 27 de novembro de 2015 e 7 de dezembro de 2015, no qual 88 alunos participaram. Ao final do minicurso, os alunos foram convidados a responder um Questionário de Satisfação para avaliar conteúdo instrucional desenvolvido. Os resultados desta avaliação mostram que os alunos consideram o conteúdo instrucional desenvolvido claro, de fácil compreensão e, de modo geral, adequado para o período de duração. Alguns pontos fracos do conteúdo instrucional elaborado foram a falta de exercícios práticos (i.e., exercícios de implementação), o baixo número de exercícios, o conteúdo básico (i.e., falta de assuntos como vetores, matrizes, ponteiros, etc.) e a utilização apenas de pseudocódigo. Com a análise dos pontos negativos citados, percebe-se que os alunos gostam de desenvolver algoritmos e vê-los em execução quando os estudam, além de serem desafiados com problemas que utilizam estruturas mais complexas como vetores e matrizes. Acredita-se que o uso apenas de pseudocódigo foi considerado como um ponto negativo pois os alunos já possuíam conhecimento sobre linguagens de programação, visto que o minicurso foi realizado no final do semestre.

Por meio dos resultados obtidos na análise do Questionário de Satisfação, percebe-se que o desenvolvimento de materiais pedagógicos com base no modelo de DI ADDIE ajudou a elaborar um minicurso e materiais de qualidade, visto que, é necessário entender o problema educacional (e.g., dificuldade de compreensão, falta de recursos tecnológicos) e as possibilidades de execução (e.g., não interferir com as aulas presenciais da universidade, tecnologias disponíveis) para então elaborar estratégias adequadas. Também nota-se a importância de avaliar se o objetivo educacional foi alcançado, visto que apesar de um resultado positivo ter sido obtido, também foram encontrados pontos em que o conteúdo desenvolvido pode ser melhorado. Como trabalhos futuros, pretende-se estender a duração e o conteúdo do minicurso para corrigir as falhas identificadas e realizá-lo no início do semestre, com alunos sem conhecimentos prévios de Algoritmos e/ou Linguagens de Programação. Espera-se que com as alterações o novo minicurso possa atender ainda mais as expectativas e necessidades dos alunos.

Agradecimentos

Este trabalho foi realizado durante uma bolsa de estudos do Programa de Estágio Sênior no Exterior - CAPES Proc. Nº BEX 2539 / 15-2 (Brasil).

Referências

- Crawford, C. (2004). Non-linear instructional design model: Eternal, synergistic design and development. *British Journal of Educational Technology*, v. 35, n. 4, p. 413–420.
- de Castro, T.H.C., de Castro Júnior, A.N., de Menezes, C.S., Boeres, M.C.S. e Rauber, M.C.P. (2003). Utilizando Programação Funcional em Disciplinas Introdutórias de Computação. Anais do WEI 2003.
- Delgado, C., Xexeo, J. A. M., Souza, I. F., Rapkiewicz, C. E. e Pereira Junior, J. C. (2005). Identificando competências associadas ao aprendizado de leitura e construção de algoritmos. Anais do XXV CSBC, p. 2371-2382.
- Dick, W., Carey, L. and Carey, J. O. (2001). *The systematic design of instruction*. Nova York: Longman.
- Filatro, A. (2008). Design instrucional na prática. São Paulo: Pearson Education do Brasil.
- Filatro, A., Piconez, S. (2004). Design instrucional contextualizado. São Paulo: Senac.
- Forbellone, A. L. V. e Eberspächer, H. F (2000). Lógica de Programação: A construção de Algoritmos e Estrutura de Dados. 2a ed. São Paulo: Makron Books.
- Gasparini, I., Oliveira, J. P. M. de, Pimenta, M. S., Lima, J. V. de, Kemczinski, A., Proença Jr., M. L. e Brunetto, M. A. de O. C. (2009a) AdaptWeb[®] - Evolução e Desafios. *Cadernos de Informática*, v. 4, n. 2, p. 47-56.
- Gasparini, I. ; Rosa, C. da, Kemczinski, A., Pimenta, M. S., Palazzo, M. de Oliveira, J. (2009) Ampliando as possibilidades de Avaliação Formativa: acompanhando a participação dos alunos no AdaptWeb. Anais do Workshop do SBIE.
- Gomes, A., Henriques, J. e Mendes, A. (2008). Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores. *Educação, Formação & Tecnologias*, v. 1, n. 1, p. 93-103.
- Heinich, R., Molenda, M., Russell, J. and Smaldino, S. (2001). Instructional media and technologies for learning. 7. ed. Upper Saddle River, NJ: Prentice Hall.
- Kenski, V. M. (2015). *Design Instrucional para cursos on-line*. São Paulo: SENAC.
- Lahtinen, E., Ala-Mutka, K. e Järvinen, H. (2005). A study of the difficulties of novice programmers. Proceedings of the 10th ITiCSE '05, p. 14-18.
- Mayfield, M. (2011). Creating training and development programs: using the ADDIE method. *Development and Learning in Organizations: An International Journal*, v. 25, n. 3, p. 19-22.
- Merriënboer, J. J. G., Clark, R. E. and Croock, M. B. M. (2002). Blueprints for complex learning: The 4C/ID-model. *Educational Technology Research and Development*, v. 50, n. 2, p. 39–64.
- Moissa, B., Carvalho, L. S. de e Gasparini, I. (2014). A Web Analytics and Visualization Tool to Understand Students' Behavior in an Adaptive E-Learning System. Proceedings of HCI International 2014, Part I. p. 312-321.
- Mustaro, P. e Silveira, I. (2007). Structure of storyboard for interactive learning objects development. In: Alex Koohang e Keith Harm (Org.) Learning Objects and instructional design. Informing Science Press. Capítulo 9, p. 253–279.
- Portugol IDE (2016). <http://www.dei.estt.ipt.pt/portugol/node/2>, [acesso: Fev 2016].
- Xavier, G. M. C., Garcia, D. F., Silva, G. F. E. e Santos, A. C. dos (2004). Estudo dos Fatores que Influenciam a Aprendizagem Introdutória de Programação. In: Escola Regional de Computação Bahia-Sergipe. Feira de Santana: Universidade Estadual de Feira de Santana.
- Zanetti, H. A. P e Bonacin, R. (2014). Uso de semiótica e análise de normas em práticas de ensino de Programação de Computadores utilizando Robótica Pedagógica. *RETC*, v. 14, n. 1, p. 25-36.