Análise dos padrões comportamentais de estudantes de um MOOC sobre Iniciação à Programação para a plataforma Arduíno

Analysis of the behavioral patterns of students of a MOOC on Introduction to Programming for the Arduino platform

Napoliana Silva de Souza[1], Gabriela Trindade Perry[1]

[1] Programa de Pós-Graduação em Informática na Educação (PPGIE), Universidade Federal do Rio do Sul, Porto Alegre, Brasil.

souzapoliana2@gmail.com, gabriela.perry@ufrgs.br

Recibido: 02/04/2020 | **Aceptado**: 22/07/2020

Cita sugerida: N. Silva de Souza and G. Trindade Perry, "Análise dos padrões comportamentais de estudantes de um MOOC sobre Iniciação à Programação para a plataforma Arduíno," *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, no. 27, pp. 36-45, 2020. doi: 10.24215/18509959.27.e4

Esta obra se distribuye bajo **Licencia Creative Commons CC-BY-NC 4.0**

Resumo

A dificuldade para aprender programação é um assunto fortemente discutido em contextos de cursos da área da Computação, sendo diversos os fatores que atuam como barreiras. Devido a essa preocupação, discussões sobre como implementar estratégias para superar esse desafio têm sido apresentadas, desencadeando novas ações que possam auxiliar o aprendizado de programação. Diante disso, os MOOCs têm se apresentado como ferramentas com potencial para apoiar o ensino presencial e para atender demandas mais específicas de estudantes que buscam aprender programação on-line e de forma autônoma. Nesse contexto, essa pesquisa relata um estudo de caso de um MOOC introdutório sobre programação Arduino, disponibilizado em uma plataforma brasileira de cursos on-line. Técnicas de mineração de dados foram empregadas com o objetivo de encontrar os padrões comportamentais de 2031 estudantes que participaram do curso. Os resultados apontaram números de ações semelhantes entre os participantes, porém dois grupos se destacam entre os que apresentaram médias mais altas de engajamento no curso: estudantes com mais de 30 anos, com nível superior e que gostam do tema do curso; bem como estudantes mais jovens com ensino médio e que adoram o tema do curso.

Palavras-chave: MOOC; Programação; Dificuldades de aprendizado; Mineração de dados; Padrões de comportamento; Engajamento.

Abstract

The difficulty to learn programming is is a topic that is strongly discussed in contexts of Computer courses, and there are several factors that act as barriers. Due to this concern, discussions on how to implement strategies to overcome this challenge have been presented, generating new actions that can help the programming learning. Thus, MOOCs have been presented as tools with the potential to support classroom teaching and to meet more specific demands of students who seek to learn programming online and autonomously. In this context, this research reports a case study of an introductory MOOC on Arduino programming, available on a Brazilian online course platform. Data mining techniques were employed aiming to find the behavioral patterns of 2031 students who participated in the course. The results showed similar numbers of actions among the participants; however two groups stand out among those who showed higher averages of engagement in the course: students over the age of 30, with higher education and who like the subject of the course; as well as younger high school students who love the course theme.

Keywords: MOOC; Programming; Learning difficulties; Data mining; Behavior patterns; Engagement.

1. Introdução

A aquisição de conhecimentos acerca de uma determinada linguagem de programação de computadores exige o desenvolvimento de habilidades lógicas e práticas, sendo uma condição fundamental para a resolução de problemas com graus de complexidade diferenciados, e que pode ser um desafio para muitos estudantes [1].

As linguagens de programação têm atraído novos públicos, professores e estudantes do ensino fundamental e médio são alguns dos perfis que têm demonstrado o interesse em conhecer e realizar atividades direcionadas ao estímulo do pensamento computacional com o objetivo de desenvolver habilidades de programação e para a obtenção de conceitos computacionais [2].

Durante o percurso de aprendizado de programação pode haver desafios, requerendo dos estudantes atitudes positivas, tais como: motivação, proatividade e prática; e as atitudes contrárias a essas características podem afetar negativamente a aprendizagem [2], por isso, há a necessidade de estabelecer métodos mais eficientes de ensino capazes de simplificar o aprendizado de programação, e incentivar o interesse do aluno [1]. Enfatiza-se que o simplificar o aprendizado de programação não significa abandonar o "rigor" dos códigos e das avaliações, mas recomenda-se ter alternativas educacionais diferenciadas em termos de conteúdos, métodos de avaliação e uso de ferramentas que propiciem melhorias na aprendizagem e no desempenho dos estudantes [3].

Os métodos tradicionais de ensino podem não ser suficientes durante o aprendizado de uma linguagem de programação, podendo provocar o desinteresse, desempenhos insatisfatórios, bem como desencadear altos índices de reprovação e frustração na aprendizagem, e devido a essa preocupação novos métodos pedagógicos e de avaliação da aprendizagem têm sido preparados para atender principalmente os estudantes com baixo desempenho ou com dificuldades de aprendizado [3]. Nesse cenário, os *Massive Open Online Courses* (MOOCs) podem atuar como ferramentas educacionais de uso do professor, e também como suporte para os estudantes iniciar o aprendizado de linguagens de programação por meio de conteúdos teóricos e práticos [4].

Disponibilizar exercícios que permitam aos estudantes implementar programas/gerar códigos e receber o feedback dos professores ainda são tarefas complexas em ambientes MOOCs, o que justifica muitos cursos optarem pelo uso de avaliações convencionais, tais como os questionários de múltipla escolha [5]. No entanto, cursos on-line com finalidades de ensino de programação podem ser explorados de diversas formas, com alternativas de uso de vídeos, e outros formatos de materiais que possam ser usados como suporte ao aprendizado autônomo; além de materiais teóricos, tarefas práticas são passíveis de implementação; e interações em fóruns de discussão on-line também são alternativas para aprender colaborativamente [6]. E o fato de as plataformas de MOOCs terem um ampliado número de recursos educacionais e tecnológicos, pode-se utilizá-las para fins de inovação no ensino, e também para tornar o conhecimento ao alcance da comunidade em geral [7].

Além de possibilitar o acesso à aprendizagem on-line por um público heterogêneo, os MOOCs voltados para o ensino de programação viabilizam o uso de técnicas de análise de aprendizagem que permitem medir o engajamento, a autorregulação da aprendizagem, examinar padrões comportamentais, descobrir quais são as habilidades de programação dos estudantes e sua influência no desempenho, bem como detectar suas dificuldades de aprendizado [8].

1.1. Fatores associados às dificuldades de aprendizado de programação

De acordo com [9], as disciplinas com ênfase em linguagens de programação integram os currículos de diversos cursos de graduação de caráter tecnológico, e por esse motivo, o aprendizado de programação ocorre desde o início do curso. Todavia, mesmo havendo rotinas contínuas de contato com algum tipo de linguagem de programação, durante o curso acadêmico há frequentes relatos referentes às dificuldades de aprendizado, que abrangem a compreensão dos conceitos lógicos, codificação e uso de uma nova linguagem [9].

Geralmente, o aprendizado de programação é iniciado com emprego de pseudolinguagem e evolui para o uso de linguagens com maiores níveis de complexidade (e.g. Pascal, C ou Java), e o fato de os estudantes terem habilidades que podem se diferenciar, para alguns programar pode consistir em uma tarefa árdua em termos cognitivos ligados à necessidade de resolver de problemas [10]. Para os estudantes que apresentam dificuldades de compreensão dos fundamentos básicos de programação o avanço no aprendizado pode ter complicações, e as barreiras para aprender programação podem estar atreladas tanto ao método de ensino empregado pelo professor, quanto ao método de estudo adotado por estudantes [11], pois os hábitos reduzidos de dedicação aos estudos, e o atraso na resolução/entrega de exercícios podem resultar em baixo rendimento e desempenho insuficiente [12].

Outro aspecto capaz de comprometer o aprendizado de programação corresponde às atitudes negativas, muitos estudantes que sinalizam dificuldades podem se sentirem incapazes de solucionar problemas e desenvolver programas/algoritmos [13]. As dificuldades mais frequentes são: aprender a sintaxe de uma linguagem de programação; entender/definir estratégias para solucionar problemas; estruturas; loops; funções; entre outros obstáculos associados à semântica e codificação, e que podem desencadear o desinteresse e, posteriormente, o abandono dos estudos de programação [14].

Além das dificuldades associadas à linguagem de programação, [15] listaram a falta de comprometimento, esforço, autoconfiança, motivação, perseverança e capacidade de abstração como elementos que também

prejudicam o desenvolvimento de competências e habilidades de programação. Em se tratando de iniciantes em programação com idades mais avançadas, outras categorias de barreiras para o aprendizado de programação podem surgir, tais como: o tempo restrito de estudo devido ao exercício profissional, limitações das habilidades de pensamento lógico, dificuldades cognitivas, medo, insegurança, frustrações em termos tecnológicos e insatisfação com os métodos pedagógicos [16].

Os materiais de ensino utilizados em aulas de programação também são frequentemente citados como elementos de pouca eficácia, uma vez que não são produzidos visando as diferentes necessidades e perfis de aprendizado dos alunos; e além dos materiais e métodos insuficientes, o fato de a disciplina de programação não estar inclusa no currículo escolar também é um fator problemático, pois atrasa o desenvolvimento do pensamento lógico, e os estudantes ingressam em cursos universitários de Computação sem ter conhecimentos prévios de programação [17].

Nesse cenário de desafios que permeiam o aprendizado de programação, novas estratégias de ensino precisam ser implementadas, incluindo mudanças pedagógicas, uso de novas ferramentas, estabelecimento de estratégias práticas para melhorar a compreensão e desempenho dos estudantes, e principalmente explorar meios alternativos de avaliação de aprendizagem que se desprendam dos métodos tradicionais [18].

1.2. Benefícios e Desafios de MOOCs com Foco em Programação

Os MOOCs enquanto ferramentas de ensino e aprendizagem criam diferentes oportunidades educacionais, e em contextos de programação, os cursos on-line viabilizam o estudo de conceitos, concretização de exercícios práticos, bem como a obtenção de familiaridade com uma linguagem [19].

Outros benefícios estão relacionados ao alcance a milhares de estudantes; possibilidades de executar tarefas em ambientes de programação on-line, evitando que estudantes efetuem download e instalação de softwares; permite compartilhar códigos, oportunizando ações colaborativas e tarefas de avaliação por pares [20].

Apesar dos diversos benefícios e contribuições dos MOOCs, há alguns desafios de aprender on-line, uma das limitações dos MOOCs está vinculada a constatação do nível reduzido de interação e comunicação entre os participantes, diferenciando-se das relações sociais e interações vivenciadas no ensino presencial, havendo pouco ou nenhum contato com o professor [20].

Outro problema refere-se ao fato de os MOOCs com ênfase em Programação obterem um elevado número de inscritos, dificultando a análise individualizada dos códigos submetidos por estudantes, por isso a alternativa é recorrer aos procedimentos automatizados de análises e classificação de códigos - que também são complexos de implementar [21]. Nesse cenário, a falta de suporte do professor e de métodos de avaliação mais avançados destacam-se dentre as principais limitações de alguns MOOCs [22].

Embora existam as limitações, há alternativas de sucesso para trabalhar programação em MOOCs, tais como a integração e o uso de pacotes e *plugins* para criar ambientes de programação na nuvem [23]; disponibilização de vídeos tutorias [24]; uso de ferramentas externas de avaliação automática de códigos [25].

2. Iniciativas para o Aprendizado de Programação em MOOCs

Na literatura científica do campo de Educação a Distância (EaD) são frequentes os relatos de experiências de desenvolvimento de MOOCs que pretendem proporcionar noções introdutórias ou avançadas em programação. No estudo de [26], por exemplo, um MOOC introdutório de programação com Python foi desenvolvido e disponibilizado na plataforma edX, o propósito foi investigar os perfis dos participantes, bem como as suas motivações para se inscreverem no curso. Conforme [26], o curso on-line foi estruturado em 6 semanas e era composto por aulas em vídeos e avaliações com aplicações de questionários, e entrega de mini projetos semanais. Os resultados divulgados pelos autores mostraram que a maioria dos inscritos apresenta entre 26 e 45 anos, 66.87% eram homens, e 74.8% eram iniciantes em Python, todavia, a maioria indicou ter conhecimentos prévios em outras linguagens de programação, e as motivações mais frequentes para participar do MOOC estavam atreladas ao interesse em Python, usar os conhecimentos na prática e obter certificado.

Em [27] foi apresentada e testada uma ferramenta implantada em MOOCs de iniciação à programação que fornece suporte na resolução de exercícios de programação. Inicialmente os autores explicam que desenvolveram um "*helpdesk*" em que instrutores e estudantes universitários ofereciam suporte aos participantes de MOOCs de programação, posteriormente, as perguntas mais frequentes recebidas pelo *helpdesk* foram utilizadas para criar categorias de problemas e soluções que auxiliassem os estudantes, gerando uma ferramenta fundamentada em árvores de decisão denominada "Murelahendaja". A Murelahendaja é utilizada para registrar problemas e sugerir dicas que apoiem as resoluções de exercícios. O sistema também disponibiliza botões para os alunos sinalizarem se a solução fornecida funcionou [27]. Os resultados da pesquisa de [27] apontaram sucesso no aumento na taxa de conclusão de dois MOOCs de iniciação a programação, todavia, os autores revelaram que trabalhar com algoritmos que fornecem respostas para as dificuldades em tarefas de programação levam a obstáculos navegacionais, pois as ramificações que compõem a árvore de decisões de soluções aumentam continuamente.

Na pesquisa de [28] é descrito o desenvolvimento de um MOOC de programação VB.NET (Linguagem de Programação Visual Basic), disponibilizado no site da "Chinese University MOOC". O curso é direcionado ao aprendizado de conceitos básicos de programação orientada a objetos. De acordo com os autores o design de aprendizagem do MOOC baseia-se em *blended learning*, combinando conteúdos curriculares, atividades e avaliações em sala de aula e on-line na plataforma de MOOCs.

O trabalho de [29] apresenta um modelo de ensino da linguagem de programação C em plataformas de MOOCs. Trata-se de um modelo misto para aplicação em MOOCs que utiliza metodologia híbrida e inclui métodos diversificados, tais como *blended learning*, *microteaching* (uso de micro aulas), sala de aula invertida, aprendizado baseado em problemas, atividades em grupo com ênfase em experiências de aprendizagem colaborativas. Os autores descrevem como conduzir cada um dos métodos, bem como os diferentes papéis que os professores, estudantes e as plataformas de MOOCs devem desempenhar.

Na investigação de [30] consta uma experiência de desenvolvimento de um MOOC introdutório de programação em MATLAB, disponível na plataforma Coursera. MATLAB se diferencia de outras linguagens de programação, pois o foco não é criar sistemas ou aplicativos, mas visa abordar conceitos numéricos, e solucionar problemas matemáticos [30]. O MOOC foi estruturado em 8 módulos semanais, e era composto por vídeos aulas explicativas apresentadas por um instrutor, e vídeos com o MATLAB em uso. As atividades avaliativas também eram semanais e havia aumento nos níveis de dificuldades das tarefas. Os autores também mencionam que as sessões de aprendizagem do MOOC eram liberadas conforme os prazos/agenda definidos. Os resultados relatados mostraram que o "MATLAB MOOC" atraiu muitos participantes (80.000) de diferentes países, e a maioria (67%) tinha curso superior. Além disso, a maioria dos estudantes que concluiu o curso indicou que seus objetivos relacionados ao MOOC foram alcançados com êxito [30].

O trabalho de [31] pauta-se na apresentação de um MOOC de introdução a programação em Java lançado na plataforma edX. O MOOC foi projetado com estimativa de duração de 5 semanas e obteve 60.000 inscritos de 190 países. Os materiais de aprendizagem integravam vídeo aulas de curta duração disponibilizadas em inglês e com opções de legendas em outros idiomas, exercícios com estabelecimento de prazos para entrega, e com correção automática, bem como atividades de revisão por pares, nesse tipo de exercício os códigos em Java são avaliados por participantes do curso. Os autores ressaltam que o MOOC tem atividades diferenciadas, sendo algumas executadas em ferramentas vinculadas a plataforma e outras executadas em ferramentas externas (Codeboard[1], Greenfoot[2], Blockly[3]). Dentre as lições aprendidas, os autores destacam que há um processo trabalhoso ao produzir e editar os materiais de aprendizagem, principalmente legendas em diversos idiomas; o custo para manter uma equipe de professores; não ter uma ferramenta própria integrada à plataforma com a finalidade de editar, compilar e executar códigos em Java; e as limitações referentes à comunicação via fórum de discussão compõem a lista de dificuldades.

Visando um público mais jovem, o MOOC concebido por [32] teve o propósito de possibilitar o aprendizado de conceitos básicos de programação de computadores para adolescentes por meio de um MOOC com ênfase na linguagem Scratch. Conforme [32], o MOOC foi produzido pela "Universidad ORT Uruguay", utilizando a plataforma "Plan CEIBAL's CREA platform". O curso on-line foi composto por vídeos, fóruns, quiz, testes avaliativos. Os projetos práticos eram enviados em fóruns para que os participantes realizassem a avaliação por pares. O MOOC obteve 1290 inscrições, constatando-se maior número de adolescentes de escolas públicas. Os resultados apresentados pelos autores indicaram que a maioria dos participantes teve uma percepção positiva sobre o MOOC, classificando-o como um curso "Muito Bom".

As iniciativas referentes ao ensino e aprendizagem de programação por meio de MOOCs descritas nessa seção revelam que os cursos on-line fornecem oportunidades de aprender diferentes tipos de linguagens de programação, havendo a possibilidade de explorar metodologias de ensino e de avaliação diferenciadas e com a contribuição de alcançar inúmeros participantes.

3. Mineração de Dados em MOOCs de Programação

Em cenários educacionais, os métodos de mineração de dados são usados para análises de diferentes tipos de dados e para fins de descobertas relacionadas ao aprendizado [33]. Em contextos de MOOCs de Programação diversos são os objetivos de uso de mineração de dados. Por exemplo, na pesquisa de [34] técnicas de mineração de dados foram utilizadas em um MOOC sobre JAVA para fins de previsão de abandono. Os dados de interação e desempenho dos estudantes foram analisados para encontrar comportamentos indicativos de evasão, por meio da construção de um modelo preditivo. Para essa finalidade três algoritmos de classificação foram testados: Regressão Logística, Random Forest, Nearest Neighbor.

Em [35], dois MOOCs da plataforma Coursera foram analisados, e um dos cursos era sobre programação para redes, o qual foi intitulado "Pattern-Oriented Software Architectures for Concurrent and Networked Software". Os autores consideraram variáveis comportamentais e temporais, que incluem acesso, visualização de materiais, submissão de tarefas, realização de atividades, tais dados foram utilizados na predição de abandono por meio de tarefas de classificação com uso de *Random forests*.

No estudo de [36] um dos objetos de investigação foi um curso on-line JavaFX, sendo que dados de cliques, tempo de aprendizagem, autoavaliações e pontuações em

exercícios foram considerados para investigar os padrões comportamentais de aprendizagem dos estudantes. Para a análise de dados, os autores utilizaram a ferramenta RapidMiner e empregaram a tarefa de clusterização, mais especificamente o algoritmo X-means, incluindo o cálculo da distância euclidiana.

A análise de [37] buscou examinar os logs de um MOOC sobre programação em C++ com a intenção de efetuar a predição de obtenção de certificado, técnicas diferenciadas de aprendizado de máquina foram utilizadas, mais especificamente *Linear Discriminant Analysis* (LDA), *Logistic Regression* (LR), *Support Vector Machine* (SVM). Esses exemplos de análises em MOOCs com temas de programação indicam as diversas alternativas do uso de mineração de dados, em que as informações coletadas de estudantes são determinantes para que avanços educacionais sejam alcançados no âmbito do ensino e aprendizado de linguagens de programação. Além disso, é possível constatar as potencialidades dos MOOCs nas tarefas de mediar o aprendizado de programação, e permitir investigações direcionadas a diferentes objetivos.

4. Ações Metodológicas de Pesquisa

Essa pesquisa consistiu em um estudo de caso de um MOOC sobre programação em Arduino disponibilizado em uma plataforma brasileira de MOOCs denominada Lumina[4]. Para a análise dos dados coletados do MOOC foi utilizado o modelo de Descoberta de Conhecimento em Base de Dados (Knowledge Discovery in Databases-KDD), descrito por [38] enquanto um processo composto por nove etapas. A primeira etapa consiste em desenvolver uma compreensão do domínio, que representa o caso/ambiente a ser analisado, e inclui organizar o cenário de aplicação e definir o objetivo do KDD. A segunda refere-se à definição do conjunto de dados, e especificação dos atributos a serem analisados. A terceira etapa compreende o pré-processamento e limpeza de dados, voltando-se para a remoção de dados que podem ser insignificantes para o contexto de análise, bem como a definição de estratégias para lidar com os dados ausentes. O quarto procedimento é voltado para a transformação dos dados, a fim de preparar e adequar os dados para mineração, esse estágio compreende a transformação de atributos para que os resultados sejam bem sucedidos. A quinta etapa diz respeito à escolha da tarefa de mineração, existem várias opções, tais como: classificação, regressão, clusterização. A sexta etapa volta-se para a definição do algoritmo mais adequado para ser utilizado na mineração, a escolha deve considerar o objetivo estabelecido. A sétima etapa visa à aplicação do algoritmo, empregando ajustes/configuração de parâmetros necessários para encontrar o melhor resultado. A oitava etapa é direcionada para a avaliação e interpretação dos resultados, de forma a averiguar se o objetivo almejado foi alcançado. A última ação corresponde à etapa nove e envolve apresentar o conhecimento descoberto e como ele será usado [38]. Na sequência são descritos os procedimentos executados considerando as nove etapas.

Etapa I- Cenário de aplicação e objetivo: A mineração de dados se efetivou a partir da análise de dados de logs de um MOOC intitulado "Introdução a Arduino, o básico para começar[5]", disponibilizado na plataforma Lúmina. No início de setembro de 2019 o MOOC registrava 2652 inscritos – período em que os logs foram extraídos. O curso é caracterizado como introdutório, de nível básico, composto por três módulos de conteúdos, e em cada módulo contém um conjunto de vídeo aulas, um fórum de discussão, e um questionário de múltipla escolha para a avaliação da aprendizagem. Há também materiais complementares, um questionário de perfil, bem como um termo com as condições necessárias para ser aprovado e receber o certificado de conclusão do curso, correspondente a 20 horas, disponibilizado gratuitamente para os estudantes que cumprirem as condições estabelecidas. No que se refere ao objetivo, teve-se enquanto propósito analisar os dados de perfil e de interação dos estudantes para descobrir seus padrões comportamentais.

Etapa II- Definição do conjunto de dados: Os dados selecionados para a análise consistiu em um arquivo de logs contendo atributos numéricos e nominais associados ao comportamento de interação de 2031 estudantes que responderam ao questionário de perfil.

Etapa III- Pré-processamento e limpeza: O conjunto original dos dados de análise foi composto por dados provenientes de duas planilhas diferentes em formato csv (comma separeted value), sendo uma planilha contendo os dados de perfil e a outra referente aos logs do curso, a qual contém o número de ações de interação dos estudantes. Na planilha de perfil os atributos irrelevantes para o contexto da análise, tais como as colunas com a identificação dos alunos (nome e e-mail), e do curso (nome e ID), nomes duplicados de estudantes que responderam duas vezes o questionário, ou com perfil incompleto foram removidos. Na planilha de logs foram aplicados filtros, e a partir do atributo número de respostas enviadas à pesquisa de perfil foram desmarcadas as opções vazio e maiores do que 1, pois já desconsiderava os estudantes não respondentes e aqueles responderam duas vezes. Posteriormente, as duas planilhas foram alinhadas e os atributos de análise foram selecionados.

Etapa IV- Transformação dos dados: Ao concluir a limpeza e organização dos dados, os atributos foram renomeados, de modo a receber abreviações. Ao final o banco de dados foi composto por 2031 instâncias e 19 atributos. No quadro 1 são descritos os atributos submetidos a análise.

Quadro 1. Atributos analisados

Nome do atributo	Descrição	Formas abreviadas
Gênero	Feminino, Masculino	F, M
Idade	Idade do participante	-
Escolaridade	Ensino médio incompleto	EMI

Nome do atributo	Descrição	Formas abreviadas
	Ensino médio completo	EMC
	Ensino superior incompleto	ESI
	Ensino superior completo	ESC
	Ensino fundamental incompleto	EFI
	Ensino fundamental completo	EFC
	Pós-graduação	PG
Tema	Descreve o sentimento do aluno em relação ao tema do curso	Gosta (G) Adora (AD) Mais ou Menos (MoM) Não gosto (N) Detesto (DT)
Total de discussões criadas	Número de discussões que o estudante criou	TDC
Total de discussões visualizadas	Número de visualizações das discussões	TDV
Total de visualizações do fórum do módulo I	Número de visualizações do primeiro fórum	TVF_1
Total de visualizações do fórum do módulo II	Número de visualizações do segundo fórum	TVF_2
Materiais Extras	Número de visualizações dos materiais complementares	ME
Total de visualizações dos questionários de avaliação de aprendizagem do módulo I	Número de visualizações do primeiro questionário	TVQ_Q1
Total de visualizações do questionário de avaliação de aprendizagem do módulo II	Número de visualizações do segundo questionário	TVQ_Q2
Total de visualizações do questionário de avaliação de aprendizagem do módulo II	Número de visualizações do terceiro questionário	TVQ_Q3
Total de visualizações dos módulos	Descreve o número de visualizações dos módulos	TVM
Total de visualizações de vídeos	Número de vídeos visualizados no curso	TVV
Total de tentativas do questionário do Modulo I	Número de respostas enviadas ao primeiro questionário de avaliação	NT_ Q1
Total de tentativas do questionário do Modulo II	Número de respostas enviadas ao segundo questionário de avaliação	NT_Q2
Total de tentativas do questionário do Modulo III	Número de respostas enviadas ao terceiro questionário de avaliação	NT_Q3
Total de tentativas	Soma o número das tentativas enviadas dos 3 questionários	TT
Total de ações no curso	Soma o número de todas as ações dentro do curso	TAC

Etapa V- Escolha da tarefa de mineração: para encontrar grupos de estudantes com padrões comportamentais semelhantes foi utilizada a tarefa de clusterização.

Etapa VI- Escolha do algoritmo: na tarefa de clusterização foi utilizado o algoritmo K-means que emprega aprendizado de máquina não supervisionado para criar clusters que agrupam dados semelhantes a partir do particionamento do conjunto de dados [39].

Etapa VII- Aplicação do algoritmo: nessa etapa os dados foram carregados na ferramenta WEKA, selecionando o algoritmo Simple K-means e o método de inicialização k-means ++, bem como testado os números de clusters que determinou o melhor resultado.

Etapas VIII e IX: as etapas oito e nove da metodologia estão respectivamente relacionadas, pois a etapa VIII intenciona a apresentação/interpretação dos padrões encontrados pelo algoritmo, enquanto a etapa IX tem o interesse em apontar qual o conhecimento descoberto/capturado nos padrões.

Sumarizando os procedimentos realizados, as etapas I a VI foram dedicadas à preparação dos dados para análise, e envolveu a escolha e apresentação do MOOC, extração dos logs, que é exportado em formato de planilha, em seguida, procedeu-se com a seleção, limpeza e transformação dos atributos. Com essas atividades completadas definiu-se a tarefa de clusterização, com o uso do algoritmo K-means. A etapa VII correspondeu à tarefa prática em que o algoritmo foi usado e gerado os padrões de forma automatizada. Restando a etapa VIII que é relativa aos resultados que mostram os clusters/padrões gerados e suas descrições, e a etapa IX que indica a descoberta feita nos dados. Essas duas últimas etapas são discutidas na Seção 5.

5. Resultados

Para a análise dos resultados, buscou-se na etapa VIII apresentar os clusters e os padrões capturados por eles, e identificar respostas aos seguintes questionamentos:

1. Qual foi o grupo predominante dentro do curso e quais os seus padrões comportamentais?
2. Qual grupo apresentou maior média de engajamento?
3. O número de tentativas de testes pode ser maior ou menor conforme a média de engajamento?
4. A média de engajamento pode ser maior ou menor conforme o sentimento relativo ao tema do curso?

Durante a configuração e testes do número de clusters (etapa VII da metodologia), identificou-se que o gênero não foi um atributo informativo. A principal justificativa para o ocorrido refere-se ao fato de que dentre o total de 2031 estudantes do banco de dados analisado, a maioria (1588) eram homens e 443 eram mulheres, e seus comportamentos não eram muito diferentes em termos de número de ações no curso. Desta maneira optou-se por excluir o atributo gênero da análise. Após excluir o atributo gênero e prosseguir com o teste de números de clusters, a primeira descoberta da análise revelou que o sentimento do aluno em relação ao tema do curso teve um valor significativo na separação dos grupos, mesmo em situações em que os estudantes tinham faixa etária e escolaridade semelhante, eram separados em grupos diferentes caso o sentimento referente ao tema do curso fosse diferente. Desse modo, a decisão do número de clusters considerou o valor WCSS (Within cluster sum of squared errors) que muda conforme o número de cluster. Após o teste com 5 clusters percebeu-se que o valor WCSS teve pouca variação, por esse motivo utilizamos o resultado com 5 clusters, disponibilizado na Figura 1.

		Cluster#				
Attribute	Full Data	0	1	2	3	4
	(2031.0)	(429.0)	(504.0)	(286.0)	(270.0)	(542.0)
Idade		20-25	26-30	15-19	31-35	20-25
Escolaridade		ESI	PG	EMC	PG	ESI
Tema		G	G	AD	G	AD
TDC		1.7626	1.7615	1.7779	1.7242	1.7638
TDV		5.7389	5.932	5.8789	6.0981	5.7754
TVF_1		3.1152	3.134	3.1449	3.2516	3.0869
TVF_2		2.116	2.1288	2.0884	2.2044	2.1435
ME		1.8032	1.687	1.9107	2.1769	1.7214
TVQ_1		3.1897	3.0315	3.7845	3.6409	3.0348
TVQ_2		3.7026	3.7118	3.8045	3.8935	3.6485
TVQ_3		4.159	4.1657	4.1521	4.1721	4.1631
TVV		5.6084	2.6448	6.6783	8.7926	4.7675
TVM		11.9464	5.4524	15.7448	20.2222	10.2306
NT_ Q1		1.749	1.7262	1.8953	1.8724	1.7007
NT_Q2		1.9697	2.006	1.993	2.1111	1.9668
NT_Q 3		2.1831	2.1918	2.1745	2.206	2.1887
TT		3.8695	3.7769	4.3093	4.5907	3.7101
TAC		15.4336	7.2282	20.6608	26.6185	13.3985

Figura 1. Número de clusters e média de ações retornadas

Conforme mostra a Figura 1 foram encontrados cinco grupos/clusters enumerados de 0 a 4, contendo as características predominantes dentro do grupo e a média de ações para cada um dos atributos. No cluster 0 foram agrupadas pessoas que indicaram gostar do tema do curso e que possuem ensino superior incompleto, nesse cluster pessoas com ensino médio incompleto também foram incluídas por terem em comum as características de gostar do tema e não terem curso superior. O cluster 1 integra em maior parte estudantes que mencionaram estar cursando ou concluiu uma pós-graduação. Esse cluster também agrupou pessoas com ensino superior completo. Desta forma, tem-se que pessoas que são formadas e indicaram gostar do tema do curso tem um perfil semelhante, formando, portanto, um único grupo. O cluster 2 agrupou estudantes de ensino médio e ensino fundamental que indicaram adorar o tema do curso. O cluster 3 tem alguns detalhes em comum com o cluster 1, referentes ao fato de a maioria das pessoas serem formadas e gostar do tema do curso. Todavia, o algoritmo agrupou separadamente em função da idade, são pessoas com idade superior a 30 anos, com curso superior e na faixa de 40 e 60 anos também integraram esse grupo. O cluster 4 também tem pontos em comum com o cluster 0, referente à idade e escolaridade, no entanto, em virtude de todos os estudantes do cluster 4 terem indicado adorar o tema do curso resultou na separação em clusters diferentes.

Para responder a pergunta 1 referente ao grupo predominante, foi considerado o maior número de pessoas com características semelhantes, o cluster 0 e 4 juntos somam o maior número de estudantes com média de idades entre 21 e 25 anos e que tem ensino superior incompleto, independente de gostar ou adorar o tema do curso. No aspecto padrões de comportamentos, pode-se perceber que os números de ações para cada um dos atributos são próximos para esses dois grupos. Por exemplo: a média de visualizações de fóruns, de criação e visualização de discussões, total de tentativas de questionários são bem próximos. Sendo estudantes que criam poucas discussões, acessa pouco os materiais extras, mas visualizam em média 5 discussões, e visualizaram pelo menos 3 vezes o fórum 1, e 2 vezes o fórum 2, e que tiveram poucas tentativas de questionários enviadas, o que sinaliza algo positivo, pois o resultado indica que esses estudantes realizaram somente uma tentativa nos questionários 1 e 2 e apenas no questionário 3 que tentaram 2 vezes.

A segunda pergunta dedicou-se a identificar qual grupo apresentou a maior média de engajamento. O engajamento foi relacionado à média do total de ações dentro do curso que está armazenado no último atributo (TAC) da Figura 1. Percebe-se que os estudantes pertencentes ao cluster 3 apresentou a maior média de engajamento/ações no curso. Desta forma, o conhecimento descoberto indica que estudantes acima de 30 anos, que possuem um curso superior e indicaram gostar do curso tiveram maior engajamento (em média 26 ações).

A terceira pergunta teve a intenção de verificar se a média de tentativas usadas em testes avaliativos pode ser maior ou menor conforme a média de engajamento. Observando os atributos referentes aos números de tentativas de cada um dos questionários (NT_Q1, NT_Q2, NT_Q3), e a média de ações no curso (TAC) da Figura 1, percebe-se que ter mais ou menos engajamento no curso não determina se o estudante vai usar uma ou mais tentativas em avaliações.

Na quarta pergunta foi investigado se a média de engajamento pode ser maior ou menor conforme o sentimento relativo ao tema do curso. Verificou-se que o sentimento em relação ao curso indicou efeito no engajamento somente para o cluster 2, pois foi o segundo grupo com maior média de ações no curso, revelando que estudantes mais jovens com menos de 20 anos e que concluíram o ensino médio e indicaram adorar o tema do curso mostraram mais ações em visualizações de vídeos e no total de ações no curso. Nos demais clusters gostar ou adorar o tema não faz tanta diferença no número de ações. É importante ressaltar que na amostra analisada contendo 2031 estudantes, somente 4 pessoas indicaram não gostar do tema do curso e três sinalizaram detestar, por ser um

valor baixo o algoritmo não definiu um cluster separado para esses participantes.

A partir desses resultados podemos, conforme previsto na etapa IX, sumarizar as principais descobertas em: (i) o gênero do estudante no contexto do curso analisado não foi um atributo relevante, o que indicou que homens e mulheres podem exibir semelhanças em seus padrões comportamentais ao participar de um MOOC de programação. (ii) o sentimento do aluno em relação ao assunto do curso teve valor significativo para o algoritmo, apontando que os estudantes podem ter o mesmo perfil em termos de escolaridade e faixa etária, mas se o seu pensamento referente ao tema do curso for diferente não é o suficiente para serem considerados semelhantes. (iii) as pessoas acima de 30 anos e formadas, e estudantes mais jovens que ainda não ingressaram no ensino superior foram mais engajadas no curso. (iv) o total de ações no curso não teve interferência no número de tentativas enviadas em testes, o que significa que estudantes com número de ações diferentes usaram em média o mesmo número de tentativas em cada um dos questionários.

Conclusões

Nessa pesquisa abordamos sobre as dificuldades de aprender programação em contextos de educação presencial e on-line. No ensino presencial destacou-se o quanto as metodologias tradicionais de ensino, bem como a falta de hábitos de estudos e prática dos estudantes são alguns dos elementos que dificultam o aprendizado de programação, requerendo buscar novos métodos e ações inovadoras que possam simplificar a aprendizagem. Nesse contexto, foram citados os MOOCs como ferramentas que podem apoiar a aprendizagem de programação, especialmente de iniciantes que buscam assuntos introdutórios. Por outro lado, também foi abordado os desafios que os MOOC possuem, referentes à implementação de tarefas práticas e de correção de exercícios. Apesar destes desafios, foram encontrados diversos estudos que relataram sucesso na proposição e nas experiências com MOOCs de programação, influenciando essa pesquisa tratar esse tema, apresentando um MOOC que versa sobre Arduino, diferenciando-se das demais pesquisas citadas em termos do conteúdo do curso, e por referir-se a um MOOC de uma plataforma brasileira.

Os resultados alcançados permitiram inferir que o curso teve uma boa audiência - considerando o número de inscritos pôde-se perceber um alto número de pessoas interessadas em aprender programação e que buscam esse conhecimento em cursos on-line. Outro detalhe refere-se à diversidade do público, havendo participantes com diferentes idades e escolaridade.

No que se refere aos resultados obtidos com emprego da mineração de dados foi possível concluir que, ao observar o número de ações em cada um dos atributos, os estudantes com diferentes perfis tiveram padrões comportamentais semelhantes, porém a diferença principal consistiu no sentimento do aluno associado a tema do curso. Outra descoberta revelou que ser mais jovem e ter ensino médio, assim como ter curso superior e ter acima de trinta anos representam os perfis com maior engajamento no curso. Essa informação foi constatada nos cluster 2 e 3, em que registra-se maior número de visualizações de vídeos, dos módulos do curso e médias mais altas no total de ações.

Diante dos resultados, podemos confirmar um aspecto em comum com o estudo de [32], relacionado ao MOOC sobre Scratch, a semelhança foi que o MOOC sobre Arduino também atraiu um público mais jovem que não tem curso superior. Por outro lado, o MOOC Scratch teve uma característica mais prática, com realização de tarefas e com supervisão docente, enquanto o MOOC Arduino, os alunos somente realizaram avaliações de múltipla de escolha a respeito do conteúdo, e sem acompanhamento de professores.

Enquanto limitação de pesquisa destaca-se não ter analisado as notas dos participantes para averiguar se existem diferenças no desempenho entre pessoas mais jovens e mais velhas, ou para identificar se um determinado perfil de estudante pode ter mais dificuldade de aprendizado de programação do que outro. O desempenho não foi considerado porque nem todos os estudantes haviam realizado as avaliações. Além disso, tarefas práticas não fizeram parte do curso. Em trabalhos futuros pretende-se conduzir uma análise complementar que considere o desempenho, e implementar tarefas práticas.

Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Notas

[1] https://codeboard.io/

[2] https://www.greenfoot.org/door

[3] https://developers.google.com/blockly/

[4] Plataforma Lumina: https://lumina.ufrgs.br/

[5] Mooc Arduino: https://lumina.ufrgs.br/course/view.php?id=30

Referências

[1] C. Xiong, X. You and P. Yu, "The Method of Improving Learning Ability of "C Language Programming Design" Course," in *Recent Developments in Data Science and Business Analytics. Proceedings of the International Conference on Data Science and Business Analytics (ICDSBA- 2017)*, M. Tavana, S. Patnaik, Eds., Springer, Cham, 2018, pp. 425-433.

[2] J. Yang, G. K. W. Wong, and C. Dawes, "An Exploratory Study on Learning Attitude in Computer Programming for the Twenty-First Century," in *New Media for Educational Change. Educational Communications and Technology Yearbook,* L. Deng, W. Ma, C. Fong, Eds., Springer, Singapore, 2018, pp. 59-70.

[3] T. Nip, E. L. Gunter, G. L. Herman, J. W. Morphew, and M. West, "Using a Computer-based Testing Facility to Improve Student Learning in a Programming Languages and Compilers Course," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18)*, 2018, pp. 568-573.

[4] T. Staubitz, H. Klement, J. Rensz, R. Teusner and C. Meinel, "Towards Practical Programming Exercises and Automated Assessment in Massive Open Online Courses," in *International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, 2015, pp. 23-30.

[5] Z. Chen, A. Nguyen, A. Schlender and J. Ngiam, "Real-time programming exercise feedback in MOOCs," in *Proceedings of the 10th International Conference on Educational Data Mining*, 2017, pp. 414-415.

[6] L. Yu, L. Zhang, X. Su and X. Liu, "Research on Case Teaching Mode of Programming Course Based on Interdisciplinarity," *Advances in Social Science, Education and Humanities Research*, vol. 286, pp. 385-388, 2018.

[7] M. S. Prince, M. Figueroa, J. A. Martínez and J. M. Izquierdo, "Curso MOOC para fomentar el desarrollo de competencias digitales en estudiantes universitarios y autodidactas," *Revista Iberoamericana de Educación en Tecnología y Tecnología en Educación*, no. 17, pp. 16-29, 2016.

[8] O. H. T. Lu, J. C. H. Huang, A. Y. Q. Huang and S. J. H. Yang, "Applying learning analytics for improving students engagement and learning outcomes in an MOOCs enabled collaborative programming course," *Interactive Learning Environments*, vol. 25, no. 2, pp. 220–234, 2017.

[9] A. Miño and J. Sales, "Programación concurrente: una aproximación a las dificultades de su aprendizaje," *Revista Iberoamericana de Educación en Tecnología y Tecnología en Educación*, no. 15, pp. 7-13, 2015.

[10] N. Arellano, J. Fernandez, M. V. Rosas and M. E. Zuñiga, "Estrategia metodológica de la enseñanza de la programación para la permanencia de los alumnos de primer año de Ingeniería Electrónica," *Revista Iberoamericana de Educación en Tecnología y Tecnología en Educación*, no.13, pp. 55-60, 2014.

[11] A. S. Hashim, R. Ahmad and M. S. S. Amar, "Difficulties in Learning Structured Programming:A Case Study in UTP,” in *7th World Engineering Education Forum (WEEF)*, 2017, pp. 210-215.

[12] S. Willman, R. Lindén, E. Kaila, T. Rajala, M.-Jussi Laakso and T. Salakoski, "On study habits on an introductory course on programming,” *Computer Science Education*, vol. 25, no. 3, pp. 276-291, 2015.

[13] N. Eltegani, "Students’ Perception about Fundamental Programming Course Teaching and Learning," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 5, no. 9, pp. 6-10, 2017.

[14] S. I. Malik, "Improvements in Introductory Programming Course: Action Research Insights and Outcomes," *Systemic Practice and Action Research*, vol. 31, no. 6, pp. 637–656, 2018.

[15] N. Dorn, D. M. Berges, D. Capovilla and P. Hubwieser, "Talking at Cross Purposes - Perceived Learning Barriers by Students and Teachers in Programming Education,” in *Workshop in Primary and Secondary Computing Education (WiPSCE)*, Potsdam, Germany, 2018.

[16] P. J. Guo, "Older Adults Learning Computer Programming: Motivations, Frustrations, and Design Opportunities," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI'17)*, Denver, Colorado, USA, 2017, pp. 7070-7083.

[17] R. A. M. S. Ahmed, S. M-Raouf Mahmood, R. M. Nabi and D. L. Hussein, "The Impact of Teaching Materials on Learning Computer Programming Languages in Kurdistan Region Universities and Institutes," *Kurdistan Journal of Applied Research (KJAR)*, vol. 3, no. 1, pp. 27-33, 2018.

[18] T. Nip, E. L. Gunter, G. L. Herman, J. W. Morphew, M. West, "Using a Computer-based Testing Facility to Improve Student Learning in a Programming Languages and Compilers Course," in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, Baltimore, Maryland, USA, 2018, pp. 568-573.

[19] N. Spyropoulou, G. Demopoulou, C. Pierrakeas, I. Koutsonikos and A. Kameas, "Developing a Computer Programming MOOC," *Procedia Computer Science*, vol. 65, pp. 182-191, 2015.

[20] J. Warren, S. Rixner, J. Greiner and S. Wong, "Facilitating Human Interaction in an Online Programming Course,” in *Proceedings of the 45th ACM technical symposium on Computer science education*, Atlanta, Georgia, USA, 2014, pp. 665-670.

[21] A. Nguyen, C. Piech, J. Huang and L. Guibas, "Codewebs: Scalable Homework Search for Massive Open Online Programming Courses," in *Proceedings of the 23rd international conference on World wide web*, Seoul, Korea, 2014, pp. 491-502.

[22] X. Su, T. Wang, J. Qiu and L. Zhao, "Motivating students with new mechanisms of online assignments and examination to meet the MOOC challenges for programming,” in *IEEE Frontiers in Education Conference (FIE)*, El Paso, Texas, 2015.

[23] A. Vihavainen, M. Luukkainen and J. Kurhila, "Multi-faceted Support for MOOC in Programming," in *Proceedings of the 13th Annual Conference on*

Information Technology Education, Calgary, Alberta, Canada, 2012, pp. 171-176.

[24] M. Grandl, M. Ebner, W. Slany, S. Janisch, "It's in your pocket: A MOOC about programming for kids and the role of OER in teaching and learning contexts," in *Open Education Global Conference*, Delft, Netherlands, 2018.

[25] A. Bey, P. Jermann and P. Dillenbourg, "A Comparison between Two Automatic Assessment Approaches for Programming: An Empirical Study on MOOCs," *Educational Technology & Society*, vol. 21, no. 2, pp. 259–272, 2018.

[26] G. Psathas, S. Demetriadi, A. Tsiara and P. Chalki, "Profiles and motivations of participants in Greek MOOC for Python programming," in *Learning with MOOCS (LWMOOCS)*, Madrid, Spain, 2018, pp. 70-73.

[27] M. Lepp, T. Palts, P. Luik, K. Papli, R. Suviste, M. Säde, K. Hollo, V. Vaherpuu and E. Tõnisson, "Troubleshooters for Tasks of Introductory Programming MOOCs," *International Review of Research in Open and Distributed Learning*, vol. 19, no. 4, pp. 56-75, 2018.

[28] X. Zhang, X. Huang, F. Wang and X. Cao, "Research on MOOC-based Blended Learning of Programming Language Course," in *International Conference on Humanities and Advanced Education Technology (ICHAET 2018)*, Guangzhou, China, 2018, pp. 586-591.

[29] Z. Zhao, "Research on Mixed Mode Teaching of C Language Programming," *Advances in Social Science, Education and Humanities Research (ASSEHR)*, vol. 230, pp. 110-114, 2018.

[30] J. M. Fitzpatrick, Á. Lédeczi, G. Narasimham, L. Lafferty, R. Labrie and P. T. Mielke, "Lessons Learned in the Design and Delivery of an Introductory Programming MOOC," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education (SIGCSE'17)*, Seattle, Washington, USA, 2017, pp. 219-224.

[31] C. Alario-Hoyos, C. D. Kloos, I.Estévez-Ayres, C. Fernández-Panadero, J. Blasco, S. Pastrana, G. Suárez-Tangil and J. Villena-Román, "Diseñando un MOOC en edX: Introducción a la Programación con Java – Parte 1," in *IV Jornadas de Innovación Educativa en Ingeniería Telemática (JIE 2015)*, España, 2015, pp. 391-398.

[32] I. F. Kereki, V. Paulós, "SM4T: Scratch MOOC for Teens: A pioneer pilot experience in Uruguay," in *IEEE Frontiers in Education Conference (FIE) Proceedings*, Spain, 2014.

[33] R. S. Baker, "Educational Data Mining: An Advance for Intelligent Systems in Education," *IEEE Intelligent Systems*, vol. 29, no. 3, pp. 78-82, 2014.

[34] Y. Chen, Q. Chen, M. Zhao, S. Boyer, K. Veeramachaneni and H. Qu, "DropoutSeer: Visualizing Learning Patterns in Massive Open Online Courses for Dropout Reasoning and Prediction," in *IEEE Conference on Visual Analytics Science and Technology (VAST)*, Maryland, USA, 2016, pp. 111-120.

[35] C. Ye, D. H. Fisher, J. S. Kinnebrew, G. Narasimham, G. Biswas, K. A. Brady and B. J. Evans, "Behavior Prediction in MOOCs using Higher Granularity Temporal Information," in *Proceedings of the Second ACM Conference on Learning @ Scale (L@S)*, Vancouver, BC, Canada, 2015, pp. 335-338.

[36] Truong-Sinh An, C. Krauss and A. Merceron, "Can Typical Behaviors Identified in MOOCs be Discovered in Other Courses?," in *Proceedings of the 10th International Conference on Educational Data Mining*, Wuhan, China, 2017, pp. 220–225.

[37] C. Zhao, J. Yang, J. Liang and C. Li, "Discover learning behavior patterns to predict certification," in *11th International Conference on Computer Science & Education (ICCSE 2016)*, Nagoya University, Japan, 2016, pp. 69-73.

[38] O. Maimon and L. Rokach, "Introduction to Knowledge Discovery and Data Mining," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon, L. Rokach, Eds., Springer, Boston, MA, 2009, pp. 1-17.

[39] R. Chang, Y. H. Hung and C. F. Lin, "Survey of learning experiences and influence of learning style preferences on user intentions regarding MOOCs," *British Journal of Educational Technology*, vol. 46, no. 3, pp. 528–541, 2015.

Información de Contacto de los Autores:

Napoliana Silva de Souza
Universidade Federal do Rio Grande do Sul
Av. Paulo Gama, 110
Porto Alegre
Brasil
souzapoliana2@gmail.com
ORDID iD: https://orcid.org/0000-0002-3201-5501

Gabriela Trindade Perry
Universidade Federal do Rio Grande do Sul
Av. Paulo Gama, 110
Porto Alegre
Brasil
gabriela.perry@ufrgs.br
ORDID iD: https://orcid.org/0000-0002-9405-4477

Napoliana Silva de Souza
Doutoranda em Informática na Educação (PPGIE/UFRGS), Mestre em Ciência da Computação, e graduada em Licenciatura Plena em Informática.

Gabriela Trindade Perry
Doutorado em Informática na Educação, Mestrado em Ergonomia e Graduação em Design. Professora adjunta da UFRGS, no curso de Design, ligado à Faculdade de Arquitetura e no PPGIE/ UFRGS.