



DIFICULDADES DE APRENDIZAGEM NO ENSINO DE PROGRAMAÇÃO: UM RELATO DE EXPERIÊNCIA UTILIZANDO MINERAÇÃO DE DADOS EDUCACIONAIS

Valter dos Santos M. Neto / Instituto Federal de Educação Ciência e Tecnologia do Maranhão /
valter.neto@ifma.edu.br

Rodrigo Miranda Feitosa / Instituto Federal de Educação Ciência e Tecnologia do Maranhão /
rodrigo.feitosa@ifma.edu.br

Carlos Costa Cardoso / Instituto Federal de Educação Ciência e Tecnologia do Maranhão /
carlos.costa@ifma.edu.br

Resumo

Aprender a programar é algo imprescindível na formação de um profissional da área de Computação, principalmente para os desenvolvedores de software. Entretanto, programar não é uma tarefa simples, pois exige dos aprendizes características e conhecimentos, em sua maioria, ausentes ou de difícil aquisição. Assim, este estudo tem como objetivo identificar as principais dificuldades dos estudantes durante o processo de aprendizagem na área de programação de computadores. A realidade identificada em sala de aula é caracterizada por alunos com sérios problemas na aprendizagem, e que por vezes, resultam em altas taxas de reprovação e evasão. Nessa perspectiva, descreve-se neste trabalho uma experiência realizada com discentes pertencentes a duas turmas do Ensino Médio, do curso de Técnico em Informática, nos turnos matutino e vespertino, durante uma disciplina introdutória de programação de computadores, em uma instituição privada de ensino, na cidade de São Luís, MA, com o uso de ferramentas e técnicas de Mineração de Dados e direcionadas pelas etapas da Taxonomia revisada de Bloom, aplicada no contexto da programação de computadores. Como instrumentos de coleta de dados foram utilizadas questões de avaliações de aprendizagem realizadas pelos discentes durante a disciplina, contextualizadas, em sua maioria, com assuntos do cotidiano do aluno, através de situações-problema, caracterizadas como situações reais ou hipotéticas. Os resultados dos experimentos demonstraram que as avaliações de aprendizagem podem gerar informações importantes a respeito dos processos de ensino e aprendizagem, especialmente quando guiada por uma taxonomia de objetivos educacionais. Foi possível identificar pontos importantes como problemas com operações matemáticas, em especial, nos assuntos iniciais abordados na disciplina; dúvidas quanto ao funcionamento e aplicações de algumas estruturas ensinadas em sala. Além disso, observou-se em quais níveis da Taxonomia os alunos apresentaram mais problemas e quais relações esses estabeleceram com os aspectos relacionados a contextualização das questões e o turno do curso dos discentes.

Palavras-chave: Dificuldades em Programação. Mineração de Dados. Taxonomia revisada de Bloom.

Abstract

Learning to program is essential in the training of a computer professional, especially for software developers. However, programming is not a simple task, as it requires learners to be mostly absent or difficult to acquire. Thus, this study aims to identify the main difficulties of students during the learning process in the area of computer programming. The reality identified in the classroom is characterized by students with serious learning disabilities, which sometimes result in high failure and dropout rates. From this perspective, we describe in this paper an experience conducted with students from two high school classes, the Computer Technician course, in the morning and afternoon shifts, during an introductory discipline of computer programming, in a private educational institution. in São Luís, MA, using Data Mining tools and techniques and guided by the stages of Bloom's revised Taxonomy, applied in the context of computer programming. As data collection instruments were used questions of learning assessments made by students during the course, mostly contextualized with the student's daily affairs, through problem situations, characterized as real or hypothetical situations. The results of the experiments demonstrated that learning assessments can generate important information about teaching and learning processes, especially when guided by a taxonomy of educational goals. It was possible to identify important points as problems with mathematical operations, especially in the initial subjects covered in the discipline; doubts about the operation and application of some structures taught

in the classroom. In addition, it was observed at which levels of taxonomy the students presented the most problems and which relationships they established with the aspects related to the contextualization of the questions and the course shift of the students.

Keywords: Programming difficulties. Data Mining. Bloom's revised taxonomy.

1. INTRODUÇÃO

Nos últimos anos, diante da difícil tarefa de criar soluções computacionais, programar e com altos índices de evasão e reprovação (WATSON; LI, 2014), os processos de ensino e aprendizagem de programação têm gerado uma preocupação crescente por parte dos pesquisadores (ZANETTI; OLIVEIRA, 2015) (SOUZA et al., 2016). A motivação está associada, sobretudo, pela importância da aquisição do conhecimento de programação na formação do profissional da área de computação.

Contudo, observa-se a necessidade de buscar mecanismos que identifiquem de forma mais precisa as dificuldades dos alunos durante o processo de aprendizagem de programação. Ou seja, é preciso utilizar métodos que apontem essas informações, que nem sempre estão explícitas.

Além disso, mostra-se relevante a busca pelo apoio de uma taxonomia para auxiliar na utilização de estratégias diferenciadas para o desenvolvimento dos alunos na aquisição de conhecimento e no processo avaliativo, e que proporcione ainda aos educadores, o planejamento estruturado de suas ações, a fim de que os discentes adquiram competências necessárias para sua formação.

Nessa perspectiva, este trabalho apresenta o uso dos conceitos e técnicas de Mineração de Dados, com a Taxonomia revisada de Bloom, em dados coletados a partir de avaliações de aprendizagem realizadas por alunos de ensino médio/técnico em uma disciplina de Programação, para extrair informações significativas sobre as dificuldades enfrentadas pelos estudantes ao longo do processo de aprendizagem.

2. TAXONOMIA REVISADA DE BLOOM

Em face às grandes transformações ocorridas no contexto educacional, Anderson e Krathwohl (2001) propõem uma revisão da Taxonomia de Bloom, com a dimensão conhecimento (conteúdo) e processos cognitivos mais claramente diferenciados. Assim, foi criado um novo modelo do Domínio Cognitivo formado por seis categorias: Lembrar, Entender, Aplicar, Analisar, Avaliar e Criar, detalhadas na Quadro 1 a seguir.

2.1. Taxonomia revisada de Bloom em Programação

Em seu trabalho, Jesus e Raabe (2009) relatam seus esforços para categorizar as questões de instrumentos de avaliação de programação, segundo a Taxonomia revisada de Bloom. O Quadro 1 abaixo apresenta uma síntese das principais interpretações de cada uma das categorias no contexto de programação segundo os autores.

Quadro 1 - Taxonomia de Bloom em programação

Categoria	Interpretação em programação
Lembrar	<i>realizar tarefas que exigem conhecimentos já adquiridos.</i> – identificar elementos específicos em um trecho de código;
Entender	<i>construção de significados a partir de mensagens instrucionais, usando para isto a interpretação, exemplificação, classificação, sumarização, inferência e explicação.</i> - explicar em palavras sobre o comportamento de um trecho de código.
Aplicar	<i>utilizar conceitos aprendidos em uma nova situação.</i> - utilizar processos e algoritmos ou padrões de projeto conhecidos pelo estudante na busca pela solução de um problema que é familiar ou nova.
Analisar	<i>dividir a informação em partes e entender a inter-relação existente entre as partes.</i> - organizar as partes componentes para alcançar um objetivo geral.
Avaliar	<i>realizar julgamentos baseados em critérios e padrões</i> - encontrar erro de lógica em um trecho de código fornecido.
Criar	<i>o ato de juntar elementos para formar um todo coerente e funcional</i> - produzir um algoritmo alternativo novo ou hipotetizar que uma nova combinação de algoritmos solucionará o problema proposto.

Fonte: Do autor, 2019.

3. MINERAÇÃO DE DADOS EDUCACIONAIS

Segundo Fayyad et al., (1996), Mineração de Dados (MD) é uma etapa no processo de descoberta do conhecimento em bases de dados que consiste na análise de dados e aplicação de algoritmos específicos para a extração padrões a partir dos dados.

Desta forma, no que diz respeito ao tratamento do conhecimento, seu papel é aplicar algoritmos sobre os dados e, usando da abstração, gerar modelos de conhecimento através da exploração dos dados. Este processo é possível através de suas tarefas, como clusterização e técnicas, como particionamento.

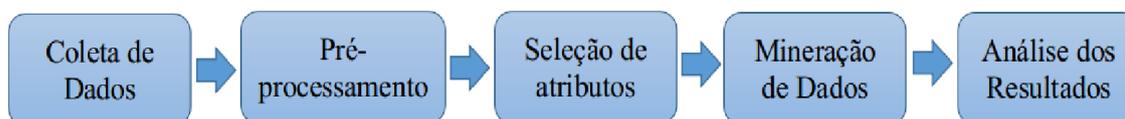
Assim, de acordo com Baker et al. (2011), a Mineração de Dados Educacionais (EDM) é uma área de pesquisa que possui como principal foco o desenvolvimento de métodos para explorar conjuntos de dados coletados em ambientes educacionais.

Para Costa et al. (2012), é uma área de pesquisa em processo de expansão, principalmente no Brasil. As informações relevantes e padrões de comportamento gerados através da mineração têm se mostrado significativos no apoio de determinadas práticas pedagógicas.

4. METODOLOGIA

Com o propósito de obter informações relevantes para compreender de forma mais precisa sobre a assimilação do conhecimento e as dificuldades de aprendizagem, aplicou-se uma metodologia baseada nos requisitos da EDM, apoiada pelo uso de uma ferramenta computacional que aplica conceitos de Mineração de Dados. A Figura 1 abaixo mostra as etapas realizadas, que serão detalhadas a seguir.

Figura 1 - Etapas da Metodologia Proposta



Fonte: Do autor, 2019.

4.1 Coleta de Dados

Para realização desta pesquisa, utilizou-se uma base de dados composta por respostas de listas de exercícios aplicadas a alunos pertencentes a duas turmas do ensino médio, do curso de Técnico em Informática, nos turnos matutino e vespertino, durante a disciplina de Lógica de Programação, em uma instituição privada de ensino.

As listas foram construídas com questões baseadas em conceitos de algoritmo e linguagem C - Introdução a Programação, Estruturas Condicionais, Estruturas de repetição, *Arrays* e Sub-rotinas (SIMÃO; REIS, 2015). Essas, foram criadas segundo os níveis da Taxonomia revisada de Bloom (Quadro 1) e seguindo dois critérios (ZANINI; RAABE, 2012): **i)** contextualizadas com assuntos do cotidiano através de situações-problema, caracterizadas como situações reais ou hipotéticas, de ordem teórica e/ou prática; **ii)** não contextualizadas.

As questões em todos os níveis, exceto no Criar, eram do tipo múltipla escolha com 4 opções. As questões nos níveis Entender, Aplicar, Analisar, Avaliar, em geral, possuíam as seguintes opções: **i)** correta; **ii)** com problemas de cunho matemático, cálculos e operadores incorretos; **iii)** com erros no uso das estruturas em estudo; e **iv)** com falhas na lógica da solução (SOUZA et al., 2016).

Nas questões classificadas como Criar, os alunos construíam programas como resolução do problema proposto utilizando as ferramentas Visualg¹, um programa que edita, interpreta e executa algoritmos com uma linguagem próxima do português estruturado como um programa normal de computador (SOUZA, 2009) e *CodeBlocks*², um ambiente integrado para desenvolvimento de software (*IDE*) para C, C++ e *Fortran* (COSTA; MATIAS, 2014).

As correções destas questões foram realizadas manualmente pelo professor da disciplina analisando os aspectos abaixo, relacionados ao objetivo do nível Criar para programação: i) **Solucionar o problema**: verifica se a solução proposta pelo aluno resolveu o problema; ii) **Código com lógica, coerente**: analisa o raciocínio lógico para a construção da solução; e iii) **Uso das estruturas**: verifica se as estruturas exigidas para a construção da solução foram utilizadas corretamente.

Assim, coletou-se manualmente os dados de 12 sessões de avaliação (104 problemas de programação), aplicadas a 43 discentes. Os dados reunidos foram tabulados em planilha eletrônica *Microsoft Office Excel*.

4.2 Pré-Processamento

Após a coleta dos dados, com o intuito de garantir a qualidade dos dados, assegurando a completude, veracidade e integridade dos dados, realizou-se a etapa de Pré-processamento. Nesta, executou-se os seguintes procedimentos (HAN; KAMBER, 2006): **i)** Preenchimento de dados de alunos que não realizaram algumas questões ou listas de exercício com resultados equivalentes aos analisados em listas anteriores ou de outros alunos com rendimentos semelhantes; **ii)** Eliminação de dados redundantes; **iii)** Remoção de dados de alunos que evadiram no início do estudo (3 alunos); **iv)** Atribuição de valores padrões para um melhor desempenho por parte do algoritmo de análise; e **v)** Transformação de valores numéricos em categóricos.

4.3 Seleção dos Atributos

Para este trabalho, adaptou-se os atributos propostos por França e Amaral (2013) e foram selecionados dois novos atributos: *TipoQuest* (ZANINI e RAABE, 2012), *Turno* (FINIMUNDI et al., 2013). A Quadro 2 a seguir apresenta esses atributos,

¹ <http://visualg3.com.br/>

² <http://www.codeblocks.org/>

com suas descrições e os domínios de valores que eles podem assumir neste trabalho.

Quadro 2 - Atributos Selecionados

Atributo	Descrição	Domínio
IdAluno	Código que identifica o aluno participante	[A1...An]
IdSessao	Código que identifica o número da sessão de avaliação	[S1...Sn]
IdQuestao	Código que identifica o número da questão da avaliação	[Q1...Qn]
NivelCognitivo	Nível cognitivo do item avaliado de acordo com a Taxonomia revisada de Bloom	LEM – Lembrar / ENT – Entender APL – Aplicar / ANA – Analisar AVA – Avaliar / CRI – Criar
RespQuestao	Opção marcada nas questões de múltipla escolha	A / B / C / D / X
ResultQuestao	Rotulo das questões de múltipla escolha	CER: Certo / ERR: Errado PAR1: Parcialmente Certo 1 PAR2: Parcialmente Certo 2 PAR3: Parcialmente Certo 3
ConceitoAval	Nível de desempenho do aprendiz numa determinada avaliação	A: $8 \geq \text{nota} \leq 10$ / B: $6 \geq \text{nota} < 8$ C: $4 \geq \text{nota} < 6$ / D: $2 \geq \text{nota} < 4$ E: $0 \geq \text{nota} < 2$
TipoQuest	Tipo da questão quanto a contextualização	CONT / NCONT
Turno	Turno do curso do aluno	MAT / VESP

Fonte: Do autor, 2019.

Por causa dos aspectos do nível Criar para programação e por ter exigido dos alunos a construção de soluções computacionais como resposta, agregou-se ao atributo *resultQuestao* os seguintes rótulos: i) **CER**: solução atendeu a todos os aspectos; ii) **PAR1**: solução atendeu de forma mais expressiva o aspecto “solucionar o problema”; iii) **PAR2**: solução atendeu de forma mais expressiva o aspecto “código com lógica”, “coerente”; iv) **PAR3**: solução atendeu de forma mais expressiva o aspecto “uso das estruturas”; v) **ERR**: solução não atendeu a nenhum dos aspectos.

O atributo *respQuestao* foi utilizado para especificar as opções disponíveis nas questões de múltipla escolha. O valor X foi atribuído as questões do nível Criar.

4.4 Mineração de Dados

Nesta etapa, foi utilizada a ferramenta *Weka*³. A escolha foi motivada pela sua interface amigável, licença livre, capacidade de otimizar o processo de extração de padrões e disponibilizar uma coleção de algoritmos para Mineração de Dados.

³ <http://www.cs.waikato.ac.nz/ml/weka>

Dentre as tarefas da Mineração de Dados, optou-se pela tarefa de Agrupamento, que possui como vantagem a possibilidade de agrupar dados similares, permitindo descrever de forma mais eficiente e eficaz as características particulares de cada grupo identificado. Isso proporciona um maior entendimento do conjunto de dados original e a descoberta de correlações relevantes entre os atributos definidos que não seriam facilmente identificadas.

Em relação aos algoritmos disponíveis na ferramenta, adotou-se o *K-means*. Este seleciona de forma aleatória k registros (centroides), de um conjunto de dados, em que cada um representa um agrupamento (*cluster*). Para cada registro restante, é calculada a similaridade entre o registro analisado e o centro de cada agrupamento. O objeto é inserido no agrupamento com a menor distância, ou seja, maior similaridade. O centro do *cluster* é recalculado a cada novo elemento inserido (COSTA et al., 2012).

A decisão foi motivada por ser um algoritmo simples e eficaz, muito conhecido na literatura e amplamente utilizado na Mineração de Dados (MELO et al., 2016).

Para obter os resultados desejados, foram construídos cinco *datasets*, referentes aos tópicos abordados durante a disciplina: Introdução a Programação (640 registros), Estruturas Condicionais (760 registros), Estruturas de Repetição (880 registros), *Arrays* (1313 registros) e Sub-rotinas (560 registros).

5. RESULTADOS

Durante os testes foram feitos agrupamentos de 2 até 6 *clusters* (grupos). Entretanto, verificou-se que o agrupamento com 5 *clusters* resultou em centroides mais consistentes para esta pesquisa. A seguir, os resultados e as análises.

Tabela 1. *Clusters dataset 1* - Introdução a programação

Atributo	Cluster#					
	Full Data (640.0)	0 (165.0)	1 (165.0)	2 (133.0)	3 (73.0)	4 (104.0)
idAluno	A1	A10	A36	A4	A23	A15
idSessao	S1	S7	S1	S7	S1	S1
idQuestao	Q1	Q1	Q5	Q2	Q5	Q7
nivelCognitivo	LEM	LEM	APL	LEM	APL	AVA
respQuestMult	D	D	A	C	A	D
resultQuestao	CER	CER	ERR	ERR	ERR	CER
concAvaliacao	C	B	C	C	D	A
turno	MAT	MAT	VESP	MAT	MAT	MAT
tipoQuest	CONT	NCONT	CONT	NCONT	CONT	CONT

Fonte: Do autor, 2019.

É possível perceber na Tabela 1 acima, que os alunos, em geral, apresentaram dificuldades neste momento inicial da disciplina, como pode ser visto pelos conceitos de avaliação C e D em 3 *clusters* (58% dos registros). Em destaque tem-se a lista de exercício S1, em que os conceitos de programação são abordados em português estruturado.

Observa-se ainda, problemas no nível Aplicar (cluster 1 e 3) da taxonomia (rótulo ERR) relacionadas a operações matemáticas. A questão em destaque (rótulos Q5 – S1) solicitava o valor final de um produto para uma dada porcentagem de desconto. A resposta em evidência (*respQuestMult A*) apresentava um erro no cálculo do desconto. Além disso, percebe-se que, mesmo a questão sendo contextualizada (rótulo CONT), alguns alunos apresentaram problemas em aplicar os conceitos abordados (rótulo ERR).

No cluster 2 destacam-se alunos, principalmente do turno matutino, com dificuldades em lembrar (rótulo LEM) o uso dos operadores relacionais, lógicos e aritméticos (rótulos S7 – Q2) para determinar o valor lógico de uma expressão. Pelo rótulo do atributo *respQuestMult (C)*, o foco está nos operadores lógicos E e OU.

Tabela 2. *Clusters dataset 2 – Estruturas Condicionais*

Atributo	Cluster#					
	Full Data (760.0)	0 (229.0)	1 (177.0)	2 (186.0)	3 (115.0)	4 (53.0)
idAluno	A1	A26	A7	A6	A31	A24
idSessao	S2	S2	S2	S8	S8	S2
idQuestao	Q1	Q4	Q9	Q2	Q1	Q9
nivelCognitivo	ENT	ENT	CRI	LEM	LEM	CRI
respQuestMult	A	C	X	C	A	X
resultQuestao	CER	CER	ERR	CER	ERR	CER
concAvaliacao	B	A	D	B	B	A
turno	MAT	VESP	MAT	MAT	VESP	VESP
tipoQuest	CONT	CONT	CONT	CONT	CONT	CONT

Fonte: Do autor, 2019.

Observa-se na Tabela 2 acima, que o desempenho dos alunos melhora neste segundo momento da disciplina (*conceitoAval A e B em 4 clusters*). Entretanto, verifica-se que alunos da manhã demonstraram dificuldades no nível Criar (rótulo CRI - *cluster 1*). Mesmo com a questão contextualizada, estes não conseguiram criar soluções que resolvessem o problema proposto (rótulo ERR).

O *cluster 3* mostra que alunos do turno vespertino tiveram problemas em lembrar (rótulo LEM) o funcionamento das estruturas condicionais *IF ELSE* e *SWITCH CASE* (rótulo Q1).

Destaca-se ainda os *clusters* 1 e 4, em que para a mesma questão (rótulos S2Q9), os resultados foram diferentes para os turnos: matutino (*respQuestMult* ERR) e vespertino (*respQuestMult* CER). Além disso, verifica-se que, em termos gerais, os alunos do turno vespertino apresentaram melhor desempenho, comparados aos do turno matutino (*clusters* 0, 3 e 4 com *concAvaliacao* entre A e B).

Tabela 3. *Clusters dataset 3 – Estruturas de Repetição*

Atributo	Cluster#					
	Full Data (880.0)	0 (129.0)	1 (191.0)	2 (288.0)	3 (188.0)	4 (84.0)
idAluno	A1	A25	A9	A1	A37	A31
idSessao	S3	S3	S3	S3	S9	S3
idQuestao	Q1	Q2	Q1	Q4	Q7	Q10
nivelCognitivo	APL	LEM	LEM	ENT	APL	AVA
respQuestMult	C	C	A	B	C	C
resultQuestao	ERR	CER	ERR	ERR	CER	CER
concAvaliacao	D	C	B	D	B	C
turno	MAT	VESP	MAT	MAT	VESP	VESP
tipoQuest	CONT	NCONT	NCONT	CONT	CONT	CONT

Fonte: Do autor, 2019.

Os dados apresentados na Tabela 3 acima, mostram que o desempenho dos alunos foi mediano (conceitos de avaliação entre B e D).

Quanto as dificuldades, destacam-se alunos do turno matutino. Apesar do conceito de avaliação B, no cluster 1 percebe-se que houve problemas no nível Lembrar (rótulo LEM), quanto ao uso e funcionamento das estruturas de repetição (rótulos S3 Q1). Além disso, tem-se no cluster 2, no nível Entender (rótulo ENT), dificuldades em compreender, mesmo em uma questão com enunciado contextualizado, um dado código em que utilizava-se a estrutura de repetição ENQUANTO..FAÇA.

Verifica-se ainda que, os alunos da manhã apresentaram desempenho pior, quando comparados aos alunos da tarde (rótulo ERR - *cluster* 1 e 2 e *conceitoAval* D *cluster* 2). Além disso, quanto ao requisito questões contextualizadas, não houve uma influência significativa neste tópico da disciplina.

Tabela 4. *Clusters dataset 4 – Arrays*

Atributo	Cluster#					
	Full Data (1313.0)	0 (308.0)	1 (277.0)	2 (305.0)	3 (309.0)	4 (114.0)
idAluno	A1	A19	A26	A20	A2	A27
idSessao	S4	S11	S4	S10	S4	S5
idQuestao	Q1	Q1	Q5	Q5	Q2	Q4

nivelCognitivo	ENT	LEM	APL	APL	ENT	APL
respQuestMult	C	A	C	D	C	B
resultQuestao	ERR	CER	ERR	ERR	CER	ERR
concAvaliacao	D	C	B	D	D	E
turno	MAT	MAT	VESP	MAT	MAT	VESP
tipoQuest	NCONT	NCONT	CONT	CONT	NCONT	CONT

Fonte: Do autor, 2019.

Analisando os conceitos das avaliações (rótulos entre C e E em sua maioria) exibidos na Tabela 4 acima, percebe-se que os alunos apresentaram um nível de dificuldade expressivo.

Em especial, destacam-se alunos do turno vespertino (clusters 1 e 4). Nestes, observa-se problemas em aplicar corretamente as estruturas estudadas. Na questão contextualizada Q5 em S4, solicitava-se o armazenamento de notas de alunos em um *array* unidimensional (vetor), mas, pelo rótulo de *respQuestMult* (C), verifica-se dúvidas na utilização correta da estrutura. Em Q4 – S5, no cluster 4, pedia-se a quantidade de um determinado elemento dentro de um *array* bidimensional (matriz). Nas soluções fornecidas como resposta, a estrutura era percorrida e utilizando o comando SE...ENTÃO, contabilizava-se o elemento pedido, caso fosse encontrado. Mas, pelo rótulo de *respQuestMult* (B), percebe-se dúvidas no funcionamento das estruturas condicionais, como foi verificado na análise da Tabela 1.

No *cluster 2*, percebe-se que alunos da manhã, também apresentam problemas em aplicar arrays nas soluções. Em Q5 – S10, os rótulos *respQuestMult* (B) e *resultQuestao* (ERR), mostram dúvidas quanto as condições presentes em IF ELSE.

Tabela 5. Clusters dataset 5 – Sub-rotinas

Atributo	Cluster#					
	Full Data (560)	0 (137.0)	1 (136.0)	2 (111.0)	3 (75.00)	4 (101.0)
idAluno	A1	A31	A5	A9	A26	A1
idSessao	S6	S6	S6	S12	S12	S12
idQuestao	Q1	Q3	Q5	Q4	Q2	Q7
nivelCognitivo	AVA	APL	AVA	ANA	ENT	CRI
respQuestMult	A	B	A	A	D	X
resultQuestao	ERR	ERR	CER	ERR	CER	ERR
concAvaliacao	D	D	D	D	D	D
turno	MAT	MAT	MAT	MAT	VESP	MAT
tipoQuest	NCONT	CONT	NCONT	NCONT	NCONT	NCONT

Fonte: Do autor, 2019.

A Tabela 5 sinaliza dificuldades nos níveis Aplicar, Analisar e Criar, principalmente em alunos do turno matutino.

No *cluster 0*, a questão contextualizada Q3 em S6, solicitava a criação de uma sub-rotina para calcular o valor final de uma compra. Analisando os rótulos *respQuestMult* (B) e *resultQuestao* (ERR), verifica-se problemas com a criação e chamada da sub-rotina.

Em *cluster 2*, os atributos *nivelCognitivo* (ANA) e *resultQuestao* (ERR) mostram dificuldades na análise de soluções para um determinado problema, que pelo rótulo de *respQuestMult* (A), são de uso incorreto das estruturas. E, no *cluster 4*, aponta para a incapacidade de alunos em criar soluções que resolvessem a questão exposta.

6. CONCLUSÃO

Com o estudo foi possível obter informações relevantes sobre as dificuldades dos estudantes de programação, como problemas com operações matemáticas, principalmente no início da disciplina; dúvidas quanto ao funcionamento das estruturas, em destaque as estruturas condicionais e de repetição e complicações no uso de *arrays*; e subrotinas. Além disso, os níveis Lembrar e Aplicar foram os mais apontados nos *clusters*.

Quanto aos fatores turno e tipo das questões, observou-se que estudantes do turno matutino apresentaram mais dificuldades em programação e que os problemas permaneceram latentes mesmo em questões contextualizadas.

A principal limitação do trabalho está na realização do processo que ocorreu, em parte, de maneira manual. Assim, como trabalho futuro pretende-se construir uma ferramenta que informe, de forma automática, os pontos críticos de cada grupo e os alunos que fazem parte destes. A proposta é que, obtendo estas informações ao final de cada etapa abordada na disciplina, seja possível construir estratégias pedagógicas para tratar as dificuldades durante o desenvolvimento da disciplina.

7. REFERÊNCIAS

ANDERSON, L. W.; KRATHWOHL, D. R. **A taxonomy for learning, teaching, and assessing: a revision of Bloom's taxonomy of educational objectives**. New York: Addison Wesley Longman. 2001.

BAKER, R.; Isotani, S.; CARVALHO, A. **Mineração de Dados Educacionais: Oportunidades para o Brasil**. RENE, v. 19, n. 2, p. 3-13. 2011.

COSTA, D. E.; MATIAS, R. **Um Estudo Experimental para Caracterização de Alocações Dinâmicas de Memória**. In: Proceedings of the IV Simpósio Brasileiro de Engenharia de Sistemas Computacionais. 2014.

COSTA, E.; BAKER, R. S.; AMORIM, L.; MAGALHÃES, J.; MARINHO, T. **Mineração de Dados Educacionais: conceitos, técnicas, ferramentas e aplicações**. Jornada de Atualização em Informática na Educação, v. 1, n. 1, p. 1-29. 2012.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. **From Data Mining to Knowledge discovery**. AI magazine, v. 17, n. 3, p. 37.1996.

FINIMUNDI, M.; RICO, E. P.; JUNQUEIRA, H.; SOUZA, D. O. **Correlação entre ritmo circadiano, turno escolar e rendimento escolar de estudantes de 11 a 17 anos de idade em escolas de ensino fundamental e médio**. REEC, v. 12, n. 2, p. 362-371. 2013.

FRANÇA, R. S.; AMARAL, H. J. C. **Aplicação de Técnicas de Mineração de Dados para o Mapeamento do Conhecimento na Aprendizagem de Programação: uma estratégia baseada na Taxonomia de Bloom**. In: XXI WEI, p. 759-768. 2013.

HAN, J.; KAMBER, M. **Data Mining: Concepts and Techniques**. Elsevier. 2006.

JESUS, E. A.; RAABE, A. L. A. **Interpretações da Taxonomia de Bloom no Contexto da Programação Introdutória**. In: SBIE, v. 1, n. 1. 2009.

MELO, Y. K.; SCHUCH, R.; FIGUEIRÓ, M. **Análise Comparativa entre Ferramentas de Mineração de Dados para Aplicação da Técnica de Clusterização**. Simpósio de Pesquisa e Desenvolvimento em Computação, v. 2, n. 1, p.1-10. 2016.

SIMÃO, D. H.; REIS, W. J. **Lógica de Programação – Conhecendo Algoritmos e Criando Programas**. Santa Cruz do Rio Pardo -SP, Viena. 2015.

SOUZA, C. M. **VisuAlg - Ferramenta de apoio ao ensino de programação**. Revista TECCEN, v. 2, n. 2, p. 1-9. 2009.

SOUZA, D. M.; BATISTA, M. H. S.; BARBOSA, E. F. **Problemas e Dificuldades no Ensino e na Aprendizagem de Programação: um mapeamento sistemático**. RENTE, v. 24, n. 1. 2016.

WATSON, C.; Li, F. W. B. **Failure rates in introductory programming revisited**. In: Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education, ACM, p. 39-44. 2014.

ZANETTI, H.; OLIVEIRA, C. **Práticas de ensino de Programação de Computadores com Robótica Pedagógica e aplicação de Pensamento Computacional**. In: Anais WCBIE, v. 4, n. 1, p. 1236-1245. 2015.

ZANINI, A. S.; RAABE, A. L. A. **Análise dos enunciados utilizados nos problemas de programação introdutória em cursos de Ciência da Computação no Brasil**. In: XX WEI. 2012.