

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
CENTRO INTERDISCIPLINAR DE NOVAS TECNOLOGIAS NA EDUCAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA NA EDUCAÇÃO

Evandro Franzen

**ESTRATÉGIA DE ENSINO E APRENDIZAGEM ATIVA APLICADA
AO APRENDIZADO DE ALGORITMOS E PROGRAMAÇÃO:
IDENTIFICAÇÃO E ANÁLISE DA MOTIVAÇÃO DOS ESTUDANTES**

Porto Alegre
2019

Evandro Franzen

**ESTRATÉGIA DE ENSINO E APRENDIZAGEM ATIVA APLICADA
AO APRENDIZADO DE ALGORITMOS E PROGRAMAÇÃO:
IDENTIFICAÇÃO E ANÁLISE DA MOTIVAÇÃO DOS ESTUDANTES**

Tese de doutorado apresentada ao Programa de Pós-Graduação em Informática na Educação do Centro Interdisciplinar de Novas Tecnologias na Educação da Universidade Federal do Rio Grande do Sul como requisito para a obtenção do título de Doutor em Informática na Educação.

Orientadora: Profa. Dra. Magda Bercht

Linha de Pesquisa: Paradigmas para a Pesquisa sobre o Ensino Científico e Tecnológico

Porto Alegre
2019

CIP – Catalogação da publicação

CIP - Catalogação na Publicação

Franzen, Evandro
ESTRATÉGIA DE ENSINO E APRENDIZAGEM ATIVA APLICADA
AO APRENDIZADO DE ALGORITMOS E PROGRAMAÇÃO:
IDENTIFICAÇÃO E ANÁLISE DA MOTIVAÇÃO DOS ESTUDANTES /
Evandro Franzen. -- 2019.
232 f.
Orientadora: Magda Bercht.

Tese (Doutorado) -- Universidade Federal do Rio
Grande do Sul, Centro de Estudos Interdisciplinares em
Novas Tecnologias na Educação, Programa de
Pós-Graduação em Informática na Educação, Porto
Alegre, BR-RS, 2019.

1. Motivação. 2. Aprendizagem Ativa. 3. Ensino e
aprendizagem de programação. 4. Problematização. I.
Bercht, Magda, orient. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da UFRGS com os
dados fornecidos pelo(a) autor(a).

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Dr. Rui Vicente Oppermann

Vice-Reitor: Profa. Dra. Jane Fraga Tutikian

Pró-Reitor de Pós-Graduação: Prof. Dr. Celso Gianetti Loureiro Chaves

Diretor do CINTED: Prof. Dr. Leandro Krug Wives

Coordenadora do PPGIE: Profa. Dra. Liane Margarida Rothenbach Tarouco

ATA DA DEFESA



SERVIÇO PÚBLICO FEDERAL
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
CENTRO INTERDISCIPLINAR DE NOVAS TECNOLOGIAS NA EDUCAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA NA EDUCAÇÃO

ATA SOBRE A DEFESA DE TESE DE DOUTORADO EVANDRO FRANZEN

As treze horas e trinta minutos do dia treze de junho de dois mil e dezenove, na sala 329 do PPGIE/CINTED, nesta Universidade, reuniu-se a Comissão de Avaliação, composta pelos Professores Doutores: Dante Augusto Couto Barone, José Palazzo Moreira de Oliveira e João Carlos Furtado para a análise da defesa de Tese de Doutorado intitulada "Estratégia de Ensino e Aprendizagem Ativa Aplicada ao Aprendizado de Algoritmos e Programação: Identificação e Análise da Motivação dos Estudantes", do doutorando do Programa de Pós-Graduação em Informática na Educação Evandro Franzen, sob a orientação da Prof.ª Dr.ª Magda Bercht.

A Banca, reunida, após a apresentação e arguição, emite o parecer abaixo assinalado.

- Considera a Tese aprovada
- sem alterações;
 - sem alterações, com voto de louvor;
 - e recomenda que sejam efetuadas as reformulações e atendidas as sugestões contidas nos pareceres individuais dos membros da Banca;

- Considera a Tese reprovada.

Considerações adicionais (a critério da Banca):

A BANCA ANALISOU QUE A CONTRIBUIÇÃO DO CANDIDATO NA ÁREA DE APRENDIZAGEM DE ALGORITMOS E PROGRAMAÇÃO É SIGNIFICATIVA PARA O NÍVEL DE DOUTORADO, SENDO O CANDIDATO RESPONDENDO ADEQUADAMENTE AS QUESTÕES FORMULADAS PELA BANCA.

SUGERE-SE QUE O CANDIDATO, NA VERGAS FOMAL, LEVE EM CONSIDERAÇÃO AS SUGESTÕES DA BANCA.

Prof.ª Dr.ª Magda Bercht
Orientadora

Prof. Dr. Dante Augusto Couto Barone
PPGIE/UFRGS

Prof. Dr. José Palazzo Moreira de Oliveira
INF/UFRGS

(vídeo conferência)

Prof. Dr. João Carlos Furtado
UNISC

AGRADECIMENTOS

Agradeço inicialmente a Deus por ter me dado tanto nesta vida, por me conceder a capacidade de aprender, permitir que eu tivesse sabedoria para tomar as decisões corretas.

Aos meus pais, Delmar e Isolde por todo o amor, carinho e por terem me mostrado a importância da honestidade e do caráter. Também agradeço pelo estímulo para aprender sempre, por me mostrarem a importância do conhecimento, mesmo que eles não tenham tido a oportunidade de dar continuidade aos seus estudos.

À minha esposa Crisna, por me acompanhar nesta e em tantas outras jornadas e pelos momentos felizes que vivemos ao longo de tantos anos. O meu desejo é que estes momentos se repitam por muitos anos ainda.

Às minhas filhas, Martina e Milena por me darem alegria, por me fazerem tão feliz. Todos os problemas ficam em segundo plano quando vejo os seus sorrisos. A família é sem dúvida a base de tudo, é por ela que vale a pena viver.

À professora Magda Bercht pela orientação, pelo apoio, conselhos e pela compreensão nos momentos de dificuldades e dúvidas.

Aos professores, membros da banca pelas observações e sugestões que contribuíram para melhorar o trabalho desenvolvido.

Aos colegas com os quais convivi durante as disciplinas cursadas, os quais não citarei os nomes para evitar esquecimentos, mas que estarão presente nas lembranças sobre estes anos tão importantes.

À Univates por ter apoiado o desenvolvimento do trabalho e por ter permitido o desenvolvimento das pesquisas nas disciplinas de programação dos cursos da instituição.

Ah, o tempo!
“Todos os dias quando acordo.
Não tenho mais o tempo que passou.
Mas tenho muito tempo.
Temos todo o tempo do mundo.
Todos os dias antes de dormir.
Lembro e esqueço como foi o dia.
Sempre em frente...não temos tempo a perder...”

Tempo Perdido. Renato Manfredini Junior (Legião Urbana)

RESUMO

Esta tese investigou estratégias para melhorar o processo de aprendizagem, que contribuam para reduzir os índices de reprovação e para desenvolver de forma mais efetiva as habilidades associadas à programação de computadores, com o apoio da percepção da dimensão afetiva do aluno, através do reconhecimento da motivação expressa por comportamentos dos alunos em interação. A justificativa para desenvolver esta investigação se origina da dificuldade enfrentada por estudantes em disciplinas de programação, que compõe os cursos de formação da área de computação, que apresentam, em sua maioria, altos índices de reprovação ou desistência. Diante das dificuldades em disciplinas introdutórias de programação, é comum perceber que os estudantes apresentam baixo esforço ou baixa confiança o que leva a uma desmotivação e conseqüentemente um desempenho abaixo do esperado. Serão apresentados os resultados da aplicação de um método baseado na problematização, e o desenvolvimento e validação de um modelo para o reconhecimento da motivação percebida do estudante. Para atingir os objetivos foi utilizada uma abordagem de pesquisa de natureza aplicada e experimental, baseada em métodos qualitativos, composta por um estudo piloto exploratório e um estudo de caso dos quais participaram 68 estudantes. A realização do estudo piloto demonstrou a viabilidade do método da problematização e a possibilidade de coletar dados para identificar a confiança e o esforço como componentes da motivação, dos estudantes. Para suportar a coleta de dados e a aplicação do método de ensino foi desenvolvido um sistema de apoio denominado PROALG (Problematização Aplicada ao ensino de Algoritmos e programação), que foi empregado para a resolução das atividades. Técnicas computacionais baseadas na mineração de dados educacionais, especialmente as Redes Bayesianas, foram utilizadas para desenvolver o modelo que permitiu determinar os níveis de confiança e esforço para cada atividade e para o estudante. Posteriormente, foi implementado um módulo que utilizou as Redes Bayesianas para determinar a probabilidade dos níveis de esforço e confiança serem categorizado como alto ou baixo. Durante o estudo piloto e o estudo de caso foram aplicados questionários para autorrelato por parte dos estudantes, cujos resultados demonstraram níveis satisfação superiores a 70%, tanto em relação ao método de ensino, quanto ao sistema de apoio. Constatou-se que os princípios adotados na classificação dos estudantes no modelo afetivo se mostraram coerentes com o autorrelato. Os resultados demonstraram também que o uso da problematização aliado ao reconhecimento a motivação é importante para o desenvolvimento de ações que visam estimular uma participação mais ativa e para o incremento da motivação dos estudantes durante a realização das tarefas. Os níveis de motivação mostrados no modelo podem complementar as observações e a percepção do professor e contribuir para identificar os estudantes que necessitam de um maior apoio e estímulo em sua aprendizagem.

Palavras-chave: Aprendizagem Ativa, problematização, ensino e aprendizagem de programação, motivação, mineração de dados educacionais.

ABSTRACT

This thesis investigated strategies to improve the learning process, which contribute to reduce failure rates and to more effectively develop the skills associated with computer programming with the support of the student's affective dimension perception, through the recognition of the motivation expressed by interacting students' behaviors. The rationale for developing this research originates from the difficulty faced by students in programming disciplines, which compose the courses in the computing area which, for the most part, present high failure or withdrawal rates. Faced with difficulties in introductory programming disciplines, it's common to see that students exhibit low effort or confidence which leads to demotivation and therefore underperformance. The results of the application of a method based on the problematization will be presented, and the development and validation of a model for the recognition of the student's perceived motivation. In order to reach the goals, an applied and experimental research approach was used, based on qualitative methods, composed of a pilot exploratory study and a case study in which 68 students participated. The pilot study demonstrated the feasibility of the problem-solving method and the possibility of collecting data to identify confidence and effort as components of student motivation. To support data collection and application of the teaching method, a support system was developed called PROALG (Problematization Applied to Algorithms and Programming), which was used to solve the activities. Computational techniques based on the mining of educational data, especially the Bayesian Networks, were used to develop the model that allowed to determine the confidence and effort levels for each activity and for the student. Subsequently, a module was implemented that used the Bayesian Networks to determine the probability of levels of effort and confidence being categorized as high or low. During the pilot study and the case study were applied to the students questionnaires for self-report, whose results showed satisfaction levels above 70%, both in relation to the teaching method and the support system. It was verified that the principles adopted in the the student's classification in the affective model were shown to be consistent with self-report. The results also showed that the use of problematization combined with the motivation recognition is important for the development of actions that aim to stimulate a more active participation and to increase students' motivation during the accomplishment of the tasks. The levels of motivation shown in the model can complement teacher observations and perceptions and contribute to identify students who need more support and encouragement in their learning.

Keywords: Active Learning, problematization, teaching and learning programming, motivation, Educational Data Mining.

LISTA DE FIGURAS

Figura 1 – Componentes da pesquisa	26
Figura 2 – Categorização dos problemas de pesquisa	38
Figura 3 – Categorização das soluções apresentadas	39
Figura 4 – Arco de Maguerez	43
Figura 5 – Quantidade de publicações sobre métodos baseados em problemas	45
Figura 6 – Roda dos Estados Afetivos	50
Figura 7 - Níveis de motivação e autodeterminação	52
Figura 8 – Etapas do processo de descoberta de conhecimento	55
Figura 9 – Exemplo de Rede Bayesiana para diagnóstico de hepatite	61
Figura 10 – Rede Bayesiana para prever desempenho de estudantes	63
Figura 11 – Métricas para avaliação de um classificador	68
Figura 12 – Exemplo de arquivo no formato arff	69
Figura 13 – Interface de pré-processamento da ferramenta Weka	70
Figura 14 – Interface do Weka para classificação usando RB	70
Figura 15 – Interface do software Netica	71
Figura 16 – Treinamento de uma rede no Netica	71
Figura 17 – Exemplo de uso da API do Netica	72
Figura 18 – Rede Bayesiana completa	73
Figura 19 – Sistema implementado para resolução das atividades	75
Figura 20 – Regras geradas	76
Figura 21 - Fluxo para o desenvolvimento das atividades	83
Figura 22 – Visão geral dos modelos para reconhecimento da motivação	90
Figura 23 – Diagrama de casos de uso para a coordenação e administração	97
Figura 24 – Diagrama de casos uso relacionado ao professor	98
Figura 25 – Diagrama de casos de uso das funcionalidades relacionadas ao aluno	99
Figura 26 – Arquitetura e tecnologias usadas no PROALG	100
Figura 27 - Modelo ER com as tabelas básicas	101
Figura 28 – Modelo ER com tabelas relacionadas aos exercícios	102
Figura 29 – Modelo ER com tabelas que armazenam dados da resolução das atividades	103
Figura 30 - Manutenção de turmas	103
Figura 31 - Manutenção de template	104
Figura 32 - Interface para definição dos pontos-chave	106
Figura 33 - Interface para definição da hipótese de solução	106
Figura 34 - Interface para elaboração do código fonte	107
Figura 35 – Listagem dos estudantes que resolveram um determinado exercício	108

Figura 36 - Tela de correção de um exercício	109
Figura 37 – Classificação da pesquisa	112
Figura 38 – Planejamento das etapas da pesquisa	113
Figura 39 – Definição inicial do problema	117
Figura 40 – Enunciado tradicional e dicas	118
Figura 41 – Processo de desenvolvimento do modelo afetivo	132
Figura 42 – Exemplo de curva normal	136
Figura 43 - Gráfico com histograma e curva para os tempos do problema da atividade 1 do estudo de caso	136
Figura 44 – Gráficos com histograma e curva para atividades com maior variabilidade nos tempos	137
Figura 45 – Conjunto de dados para determinação do esforço por atividade	142
Figura 46 – Resultados da aplicação do <i>SimpleKmeans</i> nos dados relacionados ao esforço por atividade	143
Figura 47 – Resultados da aplicação do Apriori nos dados relacionados ao esforço por atividade	144
Figura 48 – Resultados da classificação do esforço por atividade com 3 classes	146
Figura 49 – Resultados da classificação do esforço por atividade com 2 classes	146
Figura 50 – Rede Bayesiana para o reconhecimento do esforço por atividade	147
Figura 51 – Tabelas de probabilidades da rede bayesiana para o reconhecimento do esforço por atividade	148
Figura 52 – Rede Bayesiana alternativa para o reconhecimento do esforço	149
Figura 53 – Conjunto de dados para determinação da confiança por atividade	155
Figura 54 - Resultados da aplicação do SimpleKmeans nos dados relacionados à confiança por atividade	155
Figura 55 - Resultados da aplicação do Apriori nos dados relacionados à confiança por atividade	156
Figura 56 - Resultados da classificação da confiança por atividade com 2 classes	158
Figura 57 - Rede Bayesiana para o reconhecimento da confiança por atividade	159
Figura 58 – Tabelas de probabilidades da rede bayesiana para o reconhecimento da confiança por atividade	159
Figura 59 – Resultados da aplicação do método BayesNet para determinar o esforço por estudante	167
Figura 60 – Rede bayesiana para reconhecimento do esforço do estudante	168
Figura 61 – Tabela de probabilidades da rede para reconhecimento do esforço do estudante	169
Figura 62 – Resultados da aplicação do método BayesNet para determinar a confiança do estudante	172
Figura 63 – Rede bayesiana para reconhecimento da confiança do estudante	173

Figura 64 – Tabela de probabilidades da rede para reconhecimento da confiança do estudante	174
Figura 65 - Diagrama de componentes que ilustra as principais classes do módulo.	175
Figura 66 – Interface para visualização da motivação do estudante	176
Figura 67 – Exemplo de código que retorna a probabilidade da confiança por atividade	177
Figura 68 - Tabelas que armazenam os valores dos atributos usados nas redes bayesianas .	178
Figura 69 – Consulta aos dados da tabela dados_modafet_exercicio	179
Figura 70 – Resultados do módulo de acompanhamento para a atividade 5	183
Figura 71 - Configuração das redes bayesianas para confiança e esforço por atividade correspondentes ao resultado do aluno 11 na atividade 4	184
Figura 72 - Gráficos exibidos pelo módulo de acompanhamento para cada estudante	186
Figura 73 - Respostas dos estudantes classificados com alta confiança para a questão 1 (compreensão do problema)	190
Figura 74 - Respostas dos estudantes classificados com baixa confiança para a questão 1 (compreensão do problema)	191
Figura 75 - Resposta dos que tem alta confiança para a questão 2 (capacidade de resolução do problema)	191
Figura 76 -Resposta dos que tem baixa confiança para a questão 2 (capacidade de resolução do problema)	192
Figura 77 - Resposta dos alunos com alta confiança para a questão 10 (acesso as dicas)	193
Figura 78 - Resposta dos alunos com baixa confiança para a questão 10 (acesso as dicas) .	193
Figura 79 - Respostas de alunos com alta confiança para a questão 14 (satisfação)	194
Figura 80 - Respostas de alunos com baixa confiança para a questão 14 (satisfação)	195
Figura 81 - Respostas de alunos com alto esforço para a questão 5 (tempo dedicado)	195
Figura 82 - Resposta de alunos com baixo esforço para a questão 5 (tempo dedicado)	196
Figura 83 - Respostas dos que tem esforço alto para a questão 6 (tempo dedicado)	197
Figura 84 - Respostas dos que tem esforço baixo para a questão 6 (tempo dedicado)	197
Figura 85 - Respostas dos que tem esforço alto para a questão 7 (recursos)	198
Figura 86 - Respostas dos que tem esforço baixo para a questão 7 (recursos)	199
Figura 87 - Respostas dos que tem esforço alto para a questão 9 (qualidade das respostas) .	199
Figura 88 - Respostas dos que tem esforço baixo para a questão 9 (qualidade das respostas)	200
Figura 89 - Resposta dos que tem esforço alto para a questão 15 (satisfação)	200
Figura 90 - Resposta dos que tem esforço baixo para a questão 15 (satisfação)	201
Figura 91 – Respostas para a questão 3 (problematização)	201
Figura 92 - Respostas para a questão 3 (problematização)	202
Figura 93 - Respostas para a questão 13 (sistema PROALG)	203

LISTA DE TABELAS

Tabela 1 – Quantidade de artigos resultantes da pesquisa	33
Tabela 2 - Exemplo do método Naïve Bayes	64
Tabela 3 – Distribuição dos valores	65
Tabela 4 – Matriz de confusão	67
Tabela 5 – Exemplos de problemas e enunciados	86
Tabela 6 – Participantes por etapa	115
Tabela 7 – Registros de log coletados do sistema Moodle	119
Tabela 8 – Respostas com alto e baixo nível de detalhamento	121
Tabela 9 – Tempos de resolução das tarefas agrupados por aluno	122
Tabela 10 - Estatísticas sobre acessos e detalhamento por aluno	123
Tabela 11 - Tempos de resolução das tarefas agrupados por atividade	124
Tabela 12 - Estatísticas sobre acessos e detalhamento por atividade	125
Tabela 13 – Tempos de resolução das tarefas por aluno na fase 1 do estudo de caso	127
Tabela 14 – Estatísticas sobre acessos e detalhamento por aluno na fase 1 do estudo de caso	128
Tabela 15 - Tempos de resolução das tarefas por atividade na fase 1 do estudo de caso	129
Tabela 16 - Estatísticas sobre acessos e detalhamento por atividade na fase 1 do estudo de caso	130
Tabela 17 – Estatísticas sobre o nível de compreensão em cada etapa da pesquisa	131
Tabela 18 – Tempos para definição dos pontos-chave do problema nas atividades 2 e 11 do estudo de caso.	139
Tabela 19 – Pontuações para determinação da classe correspondente ao esforço por atividade	145
Tabela 20 – Tempos e determinação da classe para reconhecimento da confiança	152
Tabela 21 – Pontuações para determinação da classe correspondente a confiança por atividade	157
Tabela 22 – Síntese das variáveis para o reconhecimento do esforço por estudante	163
Tabela 23 - Pontuações para determinação da classe correspondente ao esforço por atividade	166
Tabela 24 - Pontuações para determinação da classe correspondente a confiança por atividade	171
Tabela 25 – Probabilidades de confiança e esforço por atividade e estudante	182
Tabela 26 - Probabilidades de confiança e esforço por estudante	185
Tabela 27 – Resultados da aplicação da REA	204

LISTA DE ABREVIATURAS E SIGLAS

ABP	Aprendizagem Baseada em Problema
ACM	<i>Associations for Computing Machinery</i>
ARCS	<i>Attention, Relevance, Confidence and Satisfaction</i>
API	<i>Application Programming Interface</i>
ARRF	<i>Attribute-Relation File Format</i>
ASCII	<i>American Standard Code for Information Interchange</i>
AVA	Ambiente Virtual de Aprendizagem
AVEA	Ambiente Virtual de Ensino e Aprendizagem
CA	Computação Afetiva
CCS	<i>Cascading Style Sheets</i>
EBSCO	Elton B. Stephens Co.
ECI	Esforço, Confiança e Independência
EDM	Educational Data Mining
ER	Entidade Relacionamento
DCN	Diretrizes Nacionais Curriculares
GERG	<i>Geneva Emotion Research Group</i>
GEW	<i>Geneva Emotion Wheel</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
KDD	<i>Knowledge Discovery in Database</i>
LA	<i>Learning Analytics</i>
MEC	Ministério da Educação
MP	Metodologia da Problematização
MSLQ	<i>Motivated Strategies for Learning Questionnaire</i>
PBL	<i>Problem Based Learning</i>
PHP	<i>Hypertext Preprocessor</i>
PROALG	Problematização Aplicada ao ensino de Algoritmos e Programação
REA	Roda dos Estados Afetivos
SBC	Sociedade Brasileira de Computação
SMS	<i>Short Message Service</i>
TI	Tecnologia da Informação
TIC	Tecnologias da Informação e Comunicação
SBIE	Simpósio Brasileiro de Informática na Educação
WEI	Workshop de Educação em Computação
WEKA	<i>Waikato Environment for Knowledge Analysis</i>
WIE	Workshop de Informática na Escola

SUMÁRIO

1	INTRODUÇÃO	17
1.1	O contexto da pesquisa	17
1.2	Motivação e justificativa.....	20
1.3	Problema de Pesquisa	24
1.4	Objetivo geral	25
1.5	Objetivos específicos	25
1.6	Planejamento e componentes da pesquisa	25
1.7	Organização da tese	27
2	FUNDAMENTAÇÃO TEÓRICA	29
2.1	O ensino de algoritmos e programação.....	33
2.2	Aprendizagem Ativa	40
2.3	Dimensão afetiva	46
2.3.1	Motivação.....	51
2.4	Técnicas computacionais de apoio	54
2.4.1	Descoberta de conhecimento em bases de dados	55
2.4.2	Mineração de dados educacionais	57
2.4.2.1	Aprendizado probabilístico	60
2.4.2.2	Redes Bayesianas	61
2.4.2.3	Classificador <i>Naïve Bayes</i>	64
2.4.2.4	Regras de associação.....	65
2.4.2.5	Agrupamento	66
2.4.3	Avaliação de classificadores	67
2.4.4	Ferramentas para mineração de dados.....	69
2.4.4.1	Weka.....	69
2.4.4.2	Netica	71
2.5	Trabalhos relacionados	72
2.5.1	Mapeamento de estados afetivos em um AVA	72
2.5.2	Descoberta do desânimo em AVEA.....	74
2.5.3	Identificação da frustração no ensino de algoritmos	75
2.5.4	Problematização aplicada ao ensino de algoritmos	77
2.5.5	Exploração do caminho percorrido pelo estudante na resolução de problemas de programação.....	78
2.5.6	A motivação e o ensino de programação.....	79

2.6	Alicerces para instrumentação da pesquisa	81
2.6.1	Método baseado na problematização para o ensino de programação.....	82
2.6.2	Problematização: contextualização do problema	84
2.6.3	Em busca do modelo para reconhecimento da motivação.....	87
2.6.4	Mineração de dados educacionais e sua utilização no modelo afetivo.....	92
3	PROALG: SISTEMA DE APOIO AO USO DA PROBLEMATIZAÇÃO NO ENSINO DE ALGORITMOS.....	95
3.1	Requisitos funcionais	96
3.2	Arquitetura e tecnologias	99
3.3	Projeto da base de dados	100
3.4	Interfaces do sistema.....	103
4	METODOLOGIA DA PESQUISA	111
4.1	Etapas da pesquisa	113
4.1.1	Definição do tema e proposta de estudo.....	114
4.1.2	Definição da unidade de estudo e público-alvo.....	114
4.1.3	Revisão de literatura.....	115
4.2	Estudo piloto	116
4.2.1	Análise dos dados coletados no estudo piloto	118
4.3	Estudo de caso	125
4.3.1	Análise dos dados coletados no estudo de caso	126
5	RECONHECIMENTO DA MOTIVAÇÃO: DESENVOLVIMENTO DO MODELO AFETIVO	132
5.1	Reconhecimento do esforço em cada atividade	134
5.1.1	Pré-processamento e transformação: variáveis relacionadas ao esforço	134
5.1.2	Reconhecimento do esforço por atividade: análise exploratória	142
5.1.3	Reconhecimento do esforço por atividade: construção e treinamento das redes bayesianas	144
5.2	Reconhecimento da confiança em cada atividade	149
5.2.1	Pré-processamento e transformação: variáveis relacionadas à confiança	150
5.2.2	Reconhecimento da confiança por atividade: análise exploratória	154
5.2.3	Reconhecimento da confiança por atividade: construção e treinamento das redes bayesianas	156
5.3	Reconhecimento do esforço geral do estudante.....	160
5.3.1	Reconhecimento do esforço por estudante: síntese das variáveis relacionadas às atividades.....	162
5.3.2	Reconhecimento do esforço por estudante: construção das redes bayesianas.....	165

5.4	Reconhecimento da confiança geral do estudante	169
5.4.1	Reconhecimento da confiança por estudante: síntese das variáveis relacionadas às atividades.....	170
5.4.2	Reconhecimento da confiança por estudante: construção das redes bayesianas	171
5.5	Módulo para acompanhamento da motivação do aluno	174
6	RESULTADOS E DISCUSSÃO	180
6.1	Validação do modelo para reconhecimento da motivação	180
6.2	Resultados da autoavaliação do estudante	189
6.2.1	Respostas para questões relacionadas à confiança	189
6.2.2	Respostas para questões relacionadas ao esforço.....	195
6.2.3	Questões para avaliar a satisfação quanto ao método e o sistema.....	201
6.2.4	Respostas relacionadas à Roda dos Estados Afetivos (REA)	203
7	CONCLUSÕES	206
7.1	Limitações da tese.....	210
7.2	Trabalhos futuros	211
8	REFERÊNCIAS	212
8.1	Publicações relacionadas à tese	223
APÊNDICE A – Interfaces (telas) do sistema PROALG.....		224
APÊNDICE B – Questionários aplicados no estudo de caso.....		229
APÊNDICE C – Termo de consentimento que é assinado pelo estudante.....		231
APÊNDICE D – Certificado de registro do software PROALG.....		232

1 INTRODUÇÃO

1.1 O contexto da pesquisa

Os conhecimentos sobre Algoritmos e Programação são fundamentais nos cursos da área de computação. As Diretrizes Nacionais Curriculares (DCN) e os referenciais de formação (ZORZO et al, 2018; MEC, 2016) para cursos da área de computação apresentam os cinco cursos de graduação em computação que são respectivamente: Ciência da Computação, Engenharia da Computação, Engenharia de Software, Sistemas de Informação e Licenciatura em Computação. Em comum, todos os cursos citados devem possuir em seus currículos uma base sólida de conteúdos relacionados ao raciocínio lógico e à programação de computadores.

O número de cursos na área de computação que experimentou um crescimento até 2015, permaneceu estável nos últimos dois anos. Relatório elaborado pela Sociedade Brasileira de Computação (SBC, 2017) mostra que entre 2014 e 2015 foram criados 30 novos cursos nesta área, no Brasil. O período de 2016 e 2017 apresentou uma pequena queda em alguns cursos, mas manteve um crescimento em outros. Entre os diferentes cursos, Engenharia de Software é o que apresenta a maior carga de disciplinas relacionadas à programação de computadores e, ao mesmo tempo é aquele que apresenta o maior crescimento em número de vagas oferecidas e matriculados (SBC, 2017). Algoritmos e programação se constituem nos componentes fundamentais no curso de Engenharia de Software, uma vez que o perfil profissional está diretamente associado ao desenvolvimento de sistemas de software, sendo assim, é possível inferir que os problemas relacionados ao ensino de programação afetam de forma ainda mais intensa este curso.

Entre os problemas verificados nos cursos da área de computação, no Brasil, estão as altas taxas de evasão. Estudo realizado pela Associação Brasileira de Empresas de Tecnologia da Informação e Comunicação – Brasscom (2013) indica que no período de 2010 a 2013 houve um acréscimo de 19% nos inscritos nos cursos de computação. A taxa média de evasão é aproximadamente 30% nas instituições privadas e 25%, nas públicas. Estudos recentes demonstram que o problema tem persistido nos últimos anos. O trabalho desenvolvido por Da Silva et al. (2018), demonstra que os cursos classificados na grande área de conhecimento Ciências, Matemática e Computação apresentam percentuais de evasão acima das demais áreas, com taxas de 25% em universidades pública e 33% nas instituições privadas.

As disciplinas de programação contribuem para a evasão nos cursos da área de computação. Trabalhos desenvolvidos por (HOED, 2017; PALMEIRA e SANTOS, 2014;

RODRIGUES, 2013; GIRAFFA e DA COSTA MORA, 2013) se dedicam a analisar os motivos que levam ao desligamento ou desistência e o perfil dos alunos que evadem. Altas taxas de reprovação, dificuldades de acompanhar as atividades, falta de conhecimentos prévios ou à alta carga de trabalho são indicativos comuns à maioria dos levantamentos.

Tradicionalmente a reprovação em disciplinas introdutórias de programação, tem sido significativa e constante ao longo dos últimos anos (BENNEDSEN e CASPERSEN, 2019). Watson e Li (2014) revisando o trabalho desenvolvido por Bennedsen e Caspersen (2007) demonstram que a média de reprovação é superior a 30%, ou seja, aproximadamente um terço dos alunos é reprovado. A pesquisa foi realizada em 15 países e 51 instituições diferentes e foram analisados também os fatores que influenciam estas taxas. Embora apresentem percentuais diferentes, outros estudos também confirmam que os percentuais de reprovação em programação é significativo (VIHAVAINEN, AIRAKSINEN e WATSON, 2014). O Brasil aparece como o terceiro país com maior reprovação, com percentual superior a 50% de alunos não aprovados (RAMOS et al., 2015).

Um questionamento recorrente diz respeito aos motivos que levam os alunos desistirem ou a apresentarem dificuldades nestas disciplinas. Grande parte dos alunos admite que tem dificuldade de compreender o enunciado ou de acompanhar o ritmo das aulas, mas, ao mesmo tempo reconhece que não faz perguntas ou questionamentos para resolução de dúvidas (GIRAFFA e DA COSTA MORA, 2013). Este reconhecimento por parte dos estudantes corrobora com a visão de que alunos, em muitos casos são passivos, interagem pouco e tendem a ficar ainda mais introspectivos quando não percebem uma evolução em sua aprendizagem. Estudantes que possuem mais facilidade para compreender os problemas tendem a interagir com mais frequência, inibindo a participação dos demais, que tem receio de questionar e assim demonstrar as suas dificuldades. Considerando este cenário, faz-se necessário considerar no processo de ensino e aprendizagem não apenas os aspectos racionais, mas também à dimensão afetiva do estudante (BOSCH e D'MELLO, 2017; BOSSE e GEROSA, 2017; KUNKLE e ALLEN, 2016).

Um aspecto que chama a atenção no estudo desenvolvido por Giraffa e Da Costa Mora (2013) é o reconhecimento por parte dos estudantes sobre a dificuldade que eles possuem para compreender o enunciado das questões. Na pesquisa citada, mais da metade dos estudantes indicou que têm dificuldade de compreender o problema a partir da leitura do enunciado. Sob o ponto de vista docente, esta é uma realidade conhecida, pois ao não compreender o enunciado, o estudante não consegue identificar os aspectos principais do problema e por consequência

não consegue construir uma solução eficiente (ZANINI e RAABE, 2012; QIAN e LEHMAN, 2017; VEERASAMY et al., 2019).

A identificação e especificação de problemas e à resolução destes usando algoritmos estão entre as principais habilidades que devem ser desenvolvidas nos componentes curriculares da área de programação (MEC, 2016). Entretanto, a programação de computadores é considerada como uma tarefa de difícil aprendizado, pois é um processo que visa transformar um plano mental, conceitos abstratos em uma solução que resolve um problema real (FIGUEIREDO e GARCINA-PEÑALVO, 2018). Na visão dos autores aprender a programar requer muito trabalho, dedicação e treinamento.

As dificuldades são ainda maiores para os estudantes iniciantes que precisam se adaptar às diferenças do ensino na graduação e passam a ter contato com a sintaxe, a semântica de linguagens de programação (FIGUEIREDO e GARCINA-PEÑALVO, 2018). Tal constatação justifica a adoção de métodos e recursos que permitam acompanhar cada estudante, compreendendo as suas dificuldades e a sua motivação para realização das tarefas.

A utilização de tecnologias no ensino superior é uma necessidade crescente uma vez que os novos estudantes estão cada vez mais familiarizados com estes recursos (AU-YONG-OLIVEIRA et al., 2018). Considerando a proximidade dos novos estudantes com tecnologias e as dificuldades no aprendizado de programação, constata-se que as Tecnologias da Informação e Comunicação (TIC) podem se constituir em recursos importantes para apoiar o desenvolvimento das atividades relacionadas à programação e o acompanhamento pelo docente, além de contribuir para melhorar a interação entre os estudantes e o professor.

A adoção de tecnologias não se resume ao uso de recursos computacionais, como software e hardware. Richey, Silver e Ely (2008) ressaltam que as tecnologias educacionais estão diretamente relacionadas à criação e manutenção de processos e recursos tecnológicos que contribuam para apoiar e melhorar os processos de ensino e aprendizagem. Seguindo esta visão, este trabalho se baseia na hipótese de que uma tecnologia educacional deve incluir não apenas ferramentas ou softwares, mas métodos ou o uso de tecnologias de forma a apoiar o trabalho de professores e estudantes.

Uma tecnologia educacional pode permitir, por exemplo, que o professor observe como o estudante resolve o problema, quando realiza as atividades, além de fornecer detalhes sobre a interação do aluno durante as tarefas. Tais sistemas podem contribuir também para a observação de aspectos relacionados à motivação, tais como esforço e a confiança demonstradas pelo discente. A identificação destes fatores pode apoiar os docentes durante o

desenvolvimento das atividades. A utilização de tecnologias pode permitir também um acompanhamento dos aspectos relacionados à dimensão afetiva e estimular a interação entre os estudantes (JARREL, HARLEY e LAJOIE, 2016).

Outro aspecto que chama a atenção nos ingressantes dos cursos de computação é visão, muitas vezes errônea que o aluno do ensino médio tem da área e do perfil do profissional de TI (Tecnologia da Informação). Jovens que utilizam a tecnologia em seu dia a dia e tem acesso constante aos recursos de comunicação, aplicativos para as mais variadas atividades e acreditam que possuem conhecimentos suficientes para desenvolver softwares e que em pouco tempo serão empresários de sucesso na área. Segundo Freud Oliveira (ASSESPRO, 2013) existe uma visão romantizada da área de TI, que é um setor como qualquer outro, ou seja, é necessária dedicação e esforço para ingressar e concluir o ensino superior.

Esta visão aliada à falta de conhecimentos prévios pode acarretar em um aprendizado insuficiente, na reprovação ou desistência. As dificuldades dos estudantes dos níveis médio nas disciplinas relacionadas à Matemática e Leitura são conhecidas (SASSAKI et al., 2018). Estes conhecimentos são importantes para o desenvolvimento das habilidades relacionadas a aplicação dos princípios da abstração, complexidade e raciocínio lógico, essenciais para que o aprendizado de programação se torne mais efetivo (SOUTO e TEDESCO, 2017; FIGUEIREDO e GARCINA-PEÑALVO, 2018).

1.2 Motivação e justificativa

A capacidade de compreender e elaborar algoritmos que resolvam problemas é fundamental para qualquer profissional que atuará na área da computação. As habilidades descritas nas diretrizes e referenciais dos cursos da área de computação (MEC, 2016; ZORZO et al., 2017) enfatizam a capacidade de identificar o problema, seus requisitos e suas características, visando a criação de soluções na forma de algoritmos e a consequente resolução usando recursos de programação. Desta forma, um estudante deve ser capaz não apenas de escrever e testar algoritmos e programas, mas também de compreender, analisar e propor soluções para problemas. Além disso, espera-se que o estudante seja capaz de identificar requisitos de problemas reais e propor soluções para estes, na forma de software (MEC, 2016).

Diversos autores têm buscado compreender os problemas e dificuldades associados ao ensino e aprendizagem de programação. Esforços recentes concentram-se na investigação dos conhecimentos, habilidades necessários para a programação (BOSSE e GEROSA, 2017; SOUTO e TEDESCO, 2017, FIGUEIREDO e GARCINA-PEÑALVO, 2018, VEERASAMY et al., 2019). Figueiredo e Garcina-Pañalvo (2018) afirmam que o problema é universal e atinge

a grande maioria dos estudantes que ingressam nos cursos superiores da área de computação, o que justifica o crescimento das pesquisas que visam compreender o problema e propor métodos ou recursos que melhorem este cenário.

Diante das dificuldades apresentadas pelos estudantes nas disciplinas ou cursos relacionados à programação, faz-se necessário, portanto, investigar estratégias para melhorar o processo de aprendizagem, que contribuam para reduzir os índices de reprovação e para desenvolver de forma mais efetiva as habilidades associadas à programação de computadores. Revisões sobre diferentes abordagens e o seu impacto nos resultados podem ser encontradas em (VIHAVAINEN, AIRAKSINEN e WATSON, 2014; MICAEL SOUZA et al., 2016). Técnicas baseadas em problemas, na contextualização com situações vivenciadas no mundo real, em problemas ou projetos estão entre as mais pesquisadas ou adotadas (DRINI, 2018; MARCELINO e KEMCZINSKI, 2018).

A busca por uma estratégia pedagógica que aproxime situações reais dos problemas utilizados no ensino de programação e o estímulo para que os estudantes se esforcem para compreender e refletir sobre estes problemas se constituiu no foco principal do trabalho desenvolvido nesta pesquisa. O estímulo por parte do professor para que exista uma interação entre os aprendizes e uma busca por conhecimentos, especialmente durante a análise do problema, antes da construção da solução final pode contribuir também para melhorar o processo de ensino e aprendizagem.

Observa-se que um grande número de estudantes apresenta dificuldades relacionadas tanto à lógica, quanto à compreensão das estruturas ou da sintaxe da linguagem. Segundo Bennedsen e Caspersen (2008) e Utting et. al. (2013) a abstração é um indicador de sucesso na programação de computadores e uma boa performance está diretamente ligada ao desenvolvimento desta habilidade. Outra abordagem que está relacionada à resolução de problemas e tem sido investigada em anos recentes é denominada “Pensamento computacional”. O pensamento computacional busca desenvolver habilidades relacionadas à decomposição de problemas complexos, o reconhecimento de padrões e à abstração (BRACKMANN, et al.). A aplicação de um método baseado na problematização pode contribuir para desenvolver as habilidades citadas, uma vez que o mesmo visa estimular o estudante a compreender e especificar um problema antes da construção da solução.

É possível constatar que professores e pesquisadores ainda tem dificuldade de perceber como a abstração se desenvolve e as técnicas que podem ser usadas para melhorar esta capacidade. Investigar a forma que um problema é compreendido e formalizado pode ser uma

estratégia para identificar alunos que apresentam mais dificuldades para desenvolver estas habilidades. O trabalho descrito nesta tese consistiu não apenas na aplicação um método que estimule a compreensão do problema (BERBEL e GAMBOA, 2011; BERBEL, 2012), mas também o registro das ações do estudante, visando acompanhar a trajetória durante as atividades. A análise desta trajetória contribuiu para compreender como os estudantes descrevem os problemas e como se dedicam para a resolução dos mesmos.

Ramos et. al. (2015) descrevem uma revisão dos trabalhos publicados entre 2001 e 2014, sobre o ensino de programação para iniciantes. O trabalho busca evidências sobre os aspectos que influenciam a reprovação ou a aprovação e as diversas formas de intervenção utilizadas por professores. Os índices apresentados variam de acordo com as publicações analisadas, mas em boa parte dos casos as reprovações encontram-se próximas de 40% e reduzem após a realização de intervenções. Uma das abordagens considerada mais promissora pelos autores é o uso de estratégias que motivem os estudantes ou que contribuam para a contextualização dos problemas e soluções.

Segundo Ipsen (2013), a elaboração de algoritmos pressupõe uma capacidade de resolver problemas utilizando uma sequência de passos que compõe uma solução. A verificação da aprendizagem na maioria das vezes se dá pela correção e avaliação dos exercícios por parte do professor, que estabelece conclusões sobre a evolução de cada aluno e as dificuldades apresentadas. Tentativas sucessivas de elaborar e executar uma solução, por parte do estudante podem gerar um conjunto de erros que se repetem ou se renovam, levando a um estado de frustração e por consequência a uma redução da motivação geral do aluno (IEPSEN, 2013).

Souto e Tedesco (2017) citando Silva et al. (2015) destacam a importância de desenvolver habilidades relacionadas à resolução de problemas, planejamento de ações e também habilidades emocionais ou atitudinais, tais como a confiança e autoestima. Variáveis positivas como a autoestima e a esperança são associadas à confiança por diversos autores (BAUMEISTER, 2003; QUARTO, 2016).

De acordo com Dum Dumaya, Banawan e Rodrigo (2018) a persistência do estudante é um dos aspectos fundamentais em tarefas que envolvem a resolução de problemas. A identificação do perfil dos estudantes e do esforço demonstrado na resolução das tarefas é fundamental para melhorar a aprendizagem. Os autores estabelecem uma relação direta entre persistência, esforço e motivação, apresentando estratégias que permitem medir e avaliar a persistência do estudante. Pacico e Busnello (2014) e Quarto (2016) trazem estabelecem a relação entre a autoeficácia, a persistência e o esforço, indicando que estudantes se percebem

mais eficazes, são mais persistentes e por consequência são mais capazes de mobilizar esforços diante dos desafios. Seguindo esta concepção, a persistência e o esforço estão associadas diretamente ao desempenho acadêmico.

Estudos realizados por Twenge e Campbell (2011), Bolton (2013) indicam que gerações mais novas apresentam elevados níveis de narcisismo, ansiedade e depressão, além de menor capacidade para assumir a responsabilidade pelo sucesso ou fracasso. Estas características associadas ao senso comum de que jovens dominam tecnologias digitais contribui para criar uma ideia de que estes estudantes terão facilidades nos cursos de computação e que a satisfação e o sucesso serão rápidos.

A realidade constatada por professores e coordenadores de curso é bem diferente, observa-se que muitos destes estudantes se decepcionam, o que reduz o engajamento e por consequência apresenta resultados abaixo do esperado. Ao se depararem com as primeiras dificuldades, ainda nas disciplinas básicas, especialmente em algoritmos estes estudantes tendem a se desmotivar e a considerar que o curso é inadequado ou que o tempo necessário para se tornar um profissional será grande demais e em boa parte dos casos, falta disposição e persistência para se manter na disciplina ou mesmo no curso (IEPSEN, 2013; SANTOS, 2016).

É nas disciplinas introdutórias que o problema acontece com mais frequência, em muitos casos o professor precisa atuar para manter o interesse, a motivação e para deixar claro que a carreira exige esforço e dedicação (QIAN e LEHMAN, 2017). Nestas disciplinas estratégias que contribuam para manter o foco na aprendizagem, motivar os alunos podem ser essenciais para a continuidade do estudante no curso. Em boa parte dos casos, estes estudantes demonstram uma evolução nos semestres seguintes e percebem que são capazes de superar as dificuldades encontradas nas primeiras disciplinas (KRAUSE e COATES, 2008).

Diante do exposto, surgem questionamentos como: O aluno realmente está aprendendo, está desenvolvendo o raciocínio computacional e a capacidade de abstração? As soluções não estão sendo desenvolvidas simplesmente por tentativa e erro? O que faz com que alguns alunos apresentem maior esforço ou confiança? Sistemas automatizados para a realização das atividades, que registrem dados relacionados à maneira ou estratégia adotada pelo estudante na elaboração da solução podem apoiar os docentes na busca por respostas a estes questionamentos. Considerando as questões e o cenário apresentado, é necessário investigar se estratégias que estimulem o aluno a pensar, refletir sobre o problema combinadas com mecanismos que permitam um acompanhamento da motivação podem acarretar em melhorias na aprendizagem de programação.

As metodologias ativas de ensino e aprendizagem têm sido adotadas nas diferentes áreas do conhecimento, porém a adoção destas no ensino de disciplinas introdutórias de programação é menos comum. Entre as metodologias ativas, destacam-se as que se baseiam na resolução de problemas, como a problematização (BERBEL, 2012). A utilização da problematização no ensino de programação, como estratégia para compreensão e formalização de problemas é algo raro, sendo encontradas poucas referências bibliográficas. Além disso, uma abordagem de ensino e aprendizagem ativa requer uma mudança no comportamento do estudante, um maior envolvimento, além de uma participação ativa no processo (BERBEL e GAMBOA, 2011).

A exigência de uma postura mais ativa pode causar um desconforto no estudante e requer acima de tudo um sujeito motivado e capaz de atuar de forma independente em determinados momentos e colaborativamente em outros (BERBEL, 2012). Ferramentas ou ambientes automatizados que permitam acompanhar as atividades desenvolvidas, os níveis de motivação demonstrados pelos estudantes podem se constituir em importantes recursos para apoiar o uso de metodologias ativas.

A Aprendizagem Ativa está diretamente associada ao pensamento reflexivo, a substituição da transmissão de conhecimento por atividades como síntese, avaliação e análise de problemas (BONWELL e EISON, 1991). A combinação de um método de aprendizagem ativa, com um modelo para a identificação dos fatores associados à motivação dos estudantes, aliados a um ambiente automatizado caracteriza a relevância e o ineditismo deste trabalho. O ambiente permitiu a aplicação dos princípios da problematização na resolução de exercícios de programação e a coleta de dados utilizados na elaboração do modelo criado para acompanhar o esforço e a confiança demonstradas pelos estudantes durante a realização das atividades. Os indicadores sobre a motivação tiveram como objetivo apoiar o processo de ensino e aprendizagem, permitindo que o professor identificasse estudantes que apresentaram dificuldades ou baixa motivação.

1.3 Problema de pesquisa

O problema de pesquisa configura-se da seguinte maneira:

Como utilizar a Aprendizagem Ativa para o aprendizado de algoritmos e programação, considerando a dimensão afetiva?

Este problema contém dois subproblemas:

É possível utilizar um método baseado na problematização para apoiar o ensino e aprendizagem de algoritmos e programação?

É possível apoiar o processo de ensino e aprendizagem com a identificação dos fatores associados à motivação na interação do estudante durante a problematização?

1.4 Objetivo geral

Investigar de que maneira é possível utilizar a Aprendizagem Ativa, para o aprendizado de algoritmos e programação, considerando a dimensão afetiva.

1.5 Objetivos específicos

Visando alcançar o objetivo geral, o mesmo foi desdobrado em objetivos específicos que orientam a construção do trabalho:

- investigar quais os fatores que permitem a identificação da motivação de alunos em interação de processos de aprendizagem em ambientes de ensino computacionais;
- definir os observáveis referentes aos fatores de identificação da motivação;
- investigar a possibilidade de utilizar um método baseado na problematização no aprendizado de programação;
- aplicar um método baseado na problematização para a resolução de problemas de programação;
- definir um modelo para o reconhecimento dos fatores associados à motivação;
- desenvolver um sistema que permita a resolução de problemas utilizando o método baseado na problematização e a aplicação do modelo para o reconhecimento da motivação;
- validar o sistema de apoio ao método e à aplicação do modelo desenvolvido;
- analisar os resultados da aplicação do método de ensino e do modelo para reconhecimento da motivação na aprendizagem dos estudantes.

1.6 Planejamento e componentes da pesquisa

A pesquisa descrita neste trabalho se constituiu no desenvolvimento e aplicação de uma tecnologia educacional baseada na aprendizagem ativa e no reconhecimento da motivação dos estudantes para o ensino e aprendizagem de algoritmos e programação.

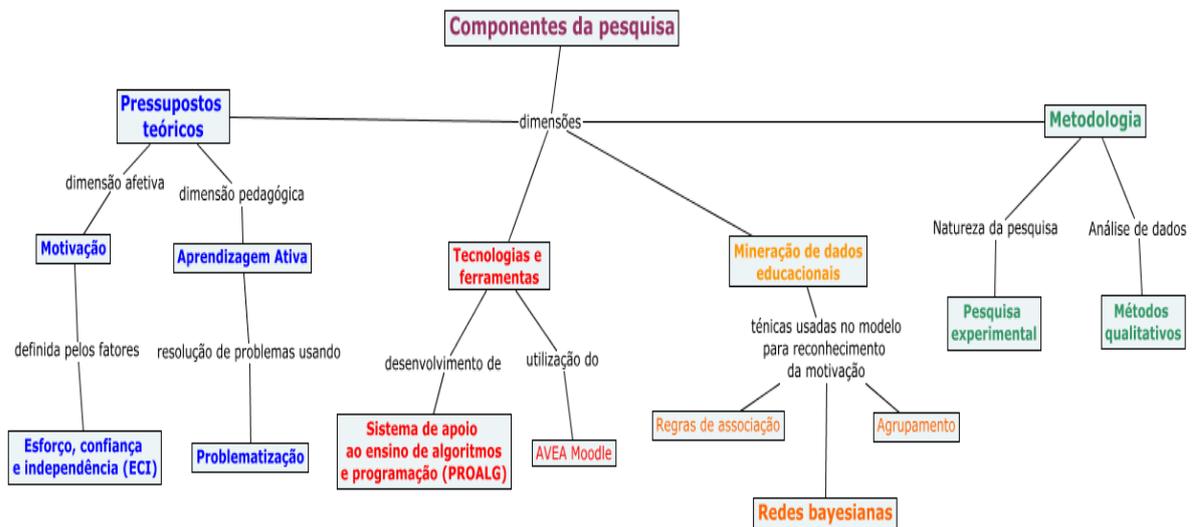
Além do método de ensino adotado, que se baseou na problematização, buscou-se coletar dados que permitiram identificar a motivação demonstrada pelos estudantes durante a interação, especialmente nas etapas que envolveram a compreensão e especificação do problema. Para apoiar a utilização do método de ensino e o reconhecimento dos fatores relacionados à motivação, foi desenvolvido um ambiente automatizado que suportou a

aplicação do método, o acompanhamento do estado afetivo do estudante, a validação da tecnologia educacional desenvolvida.

A pesquisa pode ser caracterizada como exploratória e experimental em uma primeira etapa (estudo piloto), na qual foram geradas hipóteses e um conjunto de observáveis, variáveis que foram usadas para avaliar o uso da problematização e identificar a motivação. Nas etapas seguintes foi adotada uma abordagem baseada em estudo de caso para a coleta de dados que foram utilizados no desenvolvimento do modelo para reconhecimento da motivação. Com relação às técnicas e métodos adotados, pode-se considerá-los como experimentais, alicerçados em uma abordagem predominantemente qualitativa para análise dos dados e para a elaboração e validação do modelo para reconhecimento da motivação.

A pesquisa desenvolvida pode ser dividida nas seguintes dimensões: i) os aspectos comportamentais, alicerçados na Computação Afetiva, na motivação e nos fatores de confiança e esforço, além do método utilizado no processo de ensino e aprendizagem, baseado nos conceitos da Aprendizagem Ativa; ii) as ferramentas computacionais ou tecnologias utilizadas para acompanhamento, coleta e análise dos dados e; iii) as técnicas de mineração de dados, com ênfase nas Redes Bayesianas, e; iv) os pressupostos metodológicos. O mapa mostrado na figura 1 demonstra a arquitetura da pesquisa e a relação entre os componentes.

Figura 1 – Componentes da pesquisa



Fonte: Elaborado pelo autor

A Aprendizagem Ativa se constituiu no componente pedagógico da pesquisa e dentre os diferentes métodos propostos nas metodologias ativas optou-se por efetuar uma revisão de literatura sobre as técnicas baseadas em problemas. O método adotado neste trabalho foi

inspirado nos princípios da problematização, com a adoção de parte das etapas definidas no método denominado arco de Maguerz (BORDENAVE e PEREIRA, 1982).

Para suportar o uso do método de ensino e a coleta de dados foi desenvolvido e utilizado um sistema de apoio, denominado PROALG (Problematização Aplicada ao ensino de Algoritmos e programação). Inicialmente optou-se pela utilização do AVEA (Ambiente Virtual de Ensino e Aprendizagem) Moodle para a coleta de dados no estudo piloto, período no qual o sistema citado encontrava-se em fase de desenvolvimento. A partir do estudo de caso o sistema implementado foi utilizado como ferramenta principal tanto para a coleta de dados, quanto para a elaboração e validação do modelo afetivo.

O componente motivação (Figura 1) utilizou como base o modelo ECI (Esforço, Confiança e Independência), proposto por Del Soldato e Du Boulay (1995). As atividades de definição das variáveis e elaboração do modelo para reconhecimento da motivação, foram norteadas pelos fatores citados. Técnicas de mineração de dados educacionais foram empregadas como recursos computacionais na elaboração do modelo afetivo do estudante, para o reconhecimento automático dos fatores da motivação citados.

Na elaboração do modelo para o reconhecimento da motivação foram investigadas técnicas de mineração de dados, incluindo Regras de Associação, Árvores de Decisão e Redes Bayesianas (RB). As RB foram escolhidas por apresentarem uma visão probabilística na qual o esforço e confiança são classificados como alto ou baixo, com uma certa probabilidade.

Para a análise dos dados foram adotados métodos qualitativos apoiados por ferramentas de estatística descritiva, como médias, desvios, entre outras. A pesquisa pode ser considerada de natureza qualitativa em função de não ter sido realizada uma análise estatística completa ou exaustiva. A análise qualitativa foi adotada no processo de desenvolvimento das RB, principalmente na transformação dos dados coletados. A apresentação dos resultados do autorrelato dos estudantes foi também baseada em uma análise qualitativa, com a segmentação das respostas grupos, de acordo com a classificação do esforço e da confiança.

1.7 Organização da tese

Com o objetivo de detalhar os fundamentos, a metodologia e os resultados preliminares já alcançados, este documento será estruturado da seguinte forma: o capítulo 2 apresenta os pressupostos teóricos e resultados dos levantamentos bibliográficos já realizados. Será apresentado o cenário atual relacionado ao ensino de programação, os fundamentos da aprendizagem ativa com ênfase na problematização, além das principais características, aplicações e técnicas relacionadas à mineração de dados educacionais.

Ainda no capítulo 2 serão descritos os trabalhos que serviram de inspiração para esta proposta e os alicerces para o desenvolvimento da pesquisa apresentada nesta tese. O capítulo 3 descreve o sistema PROALG e as principais funcionalidades desenvolvidas e aplicadas durante o trabalho. O capítulo seguinte apresenta detalhes sobre a metodologia utilizada, além de uma visão geral sobre as etapas da pesquisa.

Nos capítulos 5 e 6 são apresentados respectivamente o desenvolvimento do modelo para o reconhecimento da motivação, os resultados da aplicação do referido modelo e uma análise do autorrelato dos estudantes. Será apresentado também o protótipo do módulo que utiliza o modelo para reconhecer o esforço e a confiança demonstrados pelos estudantes.

2 FUNDAMENTAÇÃO TEÓRICA

Um algoritmo pode ser caracterizado com uma sequência finita de passos que serão executados para a resolução de um problema. Knuth (1998) define um conjunto de propriedades que são requisitos para a elaboração de um algoritmo:

- Finitude: o algoritmo deve terminar após um conjunto finito de passos.
- Precisão: ações devem ser definidas com precisão, não podendo haver ambiguidade na sua interpretação.
- Entradas e saídas: o algoritmo recebe um conjunto de entradas que são processadas, gerando as saídas correspondentes.

Uma linguagem de programação é formada por um conjunto de instruções, regras e uma sintaxe que deve ser respeitada durante a escrita de um programa. É possível citar como exemplo, diversas linguagens de programação utilizadas como Java, C, entre outras (FARREL, 2010). As linguagens de programação podem ser classificadas de acordo com o paradigma que indica a estrutura ou visão geral de um programa, que pode ser estruturado, orientado a objeto, funcional ou imperativo.

A programação consiste na codificação de um ou mais algoritmos em uma determinada linguagem de programação, produzindo assim o código fonte de uma aplicação ou software. O código fonte é submetido na maioria dos casos a um processo denominado compilação, que transforma o mesmo em uma representação binária, uma linguagem de máquina, que é compreendida pelos computadores.

Segundo Ascencio e Campos (2012) um programa de computador manipula dados que são armazenados em memória, denominados variáveis ou estruturas de dados. Um programador precisa identificar as estruturas adequadas para a solução do problema e as instruções que manipulam estes dados. O ensino de algoritmos e programação normalmente inicia por instruções sequenciais, de entrada, saída e manipulação de variáveis simples, passando posteriormente para o uso de instruções condicionais de repetição, além do uso de estruturas de dados mais complexas.

De acordo com Farrel (2010) as tarefas do programador vão além da escrita de instruções, sendo necessário entender o problema, especificar ou planejar a organização lógica, codificar o programa, traduzir para um linguagem de máquina e ao final executar e testar a solução. A codificação e a execução, portanto são as etapas finais do processo e são diretamente

influenciadas pela compreensão do problema e pela especificação deste. Uma interpretação incorreta do problema ou uma organização lógica mal formulada na maioria dos casos resulta em maiores dificuldades nas etapas finais.

Micael Souza, Da Silva Batista e Barbosa (2016) trazem a concepção de que programas de computador são resultados da atividade intelectual de um profissional ou estudante, desta forma o desenvolvimento da abstração, raciocínio lógico e da resolução de programas é fundamental para o ensino e aprendizado de programação.

Considerando as tarefas envolvidas na programação, é necessário analisar as habilidades a serem desenvolvidas no processo de ensino e aprendizagem. Entretanto, antes de descrever as principais habilidades relacionadas à programação e resolução de problemas, é necessário definir o que são competências, habilidades e outros conceitos importantes.

Perrenoud (1999) define competência como:

“..uma capacidade de agir eficazmente em um determinado tipo de situação, apoiada em conhecimentos, mas sem limitar-se a eles.”

Segundo o autor as ações empregadas e que demonstram as competências irão utilizar, integrar e mobilizar os conhecimentos, que podem ser declarativos, usados para representar ou descrever uma realidade ou procedimentais, que descrevem como aplicar as técnicas ou métodos (PERRENOUD, 1999).

Behar (2013) define que ser competente implica em saber julgar, avaliar e ponderar, utilizar conhecimentos para resolver com sucesso uma determinada situação. É um conjunto de atitudes, aptidões, capacidades, habilidades e conhecimentos que habilitam para atingir determinados objetivos. Segundo Behar (2013), para ser competente é preciso saber-conhecer, saber-fazer, saber-conviver e saber-ser.

A habilidade muitas vezes é confundida com competência e em muitos ambos os termos são utilizados como sinônimos, embora tenham significados diferentes. Uma habilidade está diretamente relacionada a ação, aos procedimentos e recursos que devem ser empreendidos para atingir os objetivos (PERRENOUD, 1999; BEHAR, 2013). Behar (2013) traz também a definição de um terceiro elemento, as atitudes que estão relacionadas às crenças, vontades do indivíduo, uma predisposição para enfrentar as situações e problemas.

Vários autores relacionam habilidades às competências, indicando que as habilidades devem ser desenvolvidas na busca de competências. Habilidades são desenvolvidas com o objetivo de se atingir uma competência e estão diretamente relacionadas ao saber fazer uma determinada atividade, seja física ou mental (BEHAR, 2013).

A partir destas definições é importante analisar as competências relacionadas à programação e as habilidades que devem ser desenvolvidas para formar estas competências. As diretrizes nacionais para os cursos da área de computação (MEC, 2016) estabelecem um conjunto de competências e habilidades para a formação dos profissionais nos cursos de Engenharia de Computação, Ciência da Computação, Sistemas de Informação, Engenharia de Software e Licenciatura em Computação. Um aspecto a ser ressaltado é que o texto das diretrizes não estabelece uma distinção clara entre o que são competências e habilidades.

Os referenciais de formação para os cursos de graduação em computação (ZORZO et al, 2017) é um documento que teve como referência para a sua elaboração, as diretrizes nacionais. O documento apresenta eixos de formação, cada um associado a uma competência para a formação nos diferentes cursos. Entre as competências descritas estão:

- Resolver problemas que tenham solução algorítmica.
- Desenvolver sistemas computacionais, que atendam aspectos de qualidade de processo e de produto.
- Criar, implementar e manter sistemas computacionais eficientes para diferentes problemas.
- Construir softwares considerando o projeto, as tecnologias e os ambientes de desenvolvimento disponíveis.

Desenvolver habilidades associadas à programação é algo fundamental para atingir as competências apresentadas acima. Tanto as diretrizes (MEC, 2016), quanto os referenciais (ZORZO et al, 2017) definem um conjunto de habilidades que podem ser relacionadas à programação, entre elas é possível citar:

- Reconhecer a importância do pensamento.
- Identificar os problemas que tenham solução algorítmica.
- Resolver problemas desenvolvendo algoritmos.
- Resolver problemas usando ambientes de programação.
- Identificar e analisar requisitos e especificações para problemas.
- Planejar estratégias para solucionar problemas computacionais.
- Aplicar temas e princípios, como a abstração e complexidade.

Para identificar um problema e resolver o mesmo utilizando algoritmos é necessário antes de tudo compreender os principais aspectos, possíveis entradas e resultados esperados. Após a identificação dos pontos essenciais, é possível definir uma estratégia geral, especificar as etapas ou passos que podem compor uma ou mais soluções possíveis. A elaboração do código

e os testes para determinar que este funciona corretamente constituem-se na etapa final do processo (MEC, 2016; ZORZO et al, 2017).

Além dos cursos da área de computação, as disciplinas introdutórias de programação estão presentes em vários outros cursos (STADELHOFER e GASPARINI, 2018). Gomes e Mendes (2007) em seus estudos indicam que a dificuldade de interpretar um problema é uma característica comum entre os estudantes, especialmente nas disciplinas introdutórias. Embora nas disciplinas introdutórias sejam utilizados problemas de menor complexidade, é possível constatar que o estudante deve ser capaz de realizar todas as etapas, mesmo em situações de menor porte.

A capacidade de utilizar a abstração é frequentemente citada por autores como uma habilidade importante no aprendizado de programação (HAZZAN e KRAMER, 2016). É possível definir a abstração como a habilidade de extrair características comuns a partir de exemplos e de especificar ou desenhar as características gerais de uma solução (KRAMER, 2007). Neste caso busca-se ignorar detalhes irrelevantes ou que podem até mesmo atrapalhar a resolução do problema.

Durante a compilação e execução podem ocorrer diversos erros, tanto de sintaxe, derivados da declaração e escrita incorreta de instruções e estruturas de dados, quanto de lógica, quando o programa é executado, porém não apresenta os resultados desejados (IEPSEN, 2013). Alunos que apresentam baixa capacidade de abstração para identificar e formalizar um problema tendem a buscar soluções usando métodos baseados em tentativa e erro. Antes de analisar e especificar uma possível solução, os estudantes sentem-se tentados a acessar um ambiente de programação para iniciar a escrita do código, o que leva ao tradicional processo de escrita e compilações sucessivas.

Diversas iniciativas têm sido desenvolvidas com o objetivo de desenvolver habilidades relacionadas à abstração e resolução de problemas, não apenas na graduação, mas também nos níveis de ensino fundamental e médio. Dentro da área da computação, a programação de computadores e os conteúdos relacionados as disciplinas correlatas são aqueles que geram maior preocupação entre os professores e estudantes.

A utilização de softwares como Scratch ou Alice que auxiliam na criação de jogos ou animação utilizando construções tradicionais em linguagens de programação é cada vez mais frequente (ARMONI et al., 2015; SCAICO, 2013; MARCELINO e KEMCZINSKI, 2018, PAPADAKIS et al, 2019). Estas ferramentas contribuem para o aprendizado de estruturas sequenciais, condicionais ou de repetição de forma visual e lúdica. Além do desenvolvimento

das habilidades requeridas para a área de computação, a inserção do raciocínio lógico nestes níveis de ensino permite que estudantes tenham contato maior com as atividades essenciais de um profissional de TIC.

Outras iniciativas buscam avaliar os aspectos ou problemas que podem afetar a disposição ou capacidade do estudante para resolver problemas. Experimentos demonstram que erros e resultados inesperados podem gerar diferentes sentimentos ou estados afetivos, como confusão, ansiedade ou a frustração (IEPSEN, 2013; BOSCH, 2015). Tal situação pode acarretar ainda em uma desmotivação do estudante, o que leva a um desempenho abaixo do esperado, na redução da confiança e dos esforços empreendidos pelos estudantes. Registrar as concepções do estudante sobre o entendimento e a organização lógica para enunciação do problema pode ser uma forma de conhecer as dificuldades apresentadas durante as primeiras etapas da formação profissional.

2.1 O ensino de algoritmos e programação

Esta seção apresenta um panorama geral sobre pesquisas relacionadas ao ensino de programação. O principal objetivo é demonstrar a relevância do tema e o interesse crescente de pesquisadores no assunto. Pesquisa bibliográfica realizada nas principais bases, portais ou bibliotecas digitais (Capes, ACM, IEEE Xplore e EBSCO), resultou nos dados mostrados na Tabela 1. Os dados mostrados correspondem às buscas realizadas no mês de janeiro de 2019.

Tabela 1 – Quantidade de artigos resultantes da pesquisa

Base/termos pesquisados	<i>Programming AND learning</i>		<i>Programming AND teaching</i>		Programação AND ensino		Programação AND aprendizagem	
	Todos	>=2014	Todos	>=2014	Todos	>=2014	Todos	>=2014
Capes	568297	317702	100404	25169	1030	474	666	324
ACM	71481	27797	11272	3410	43	12	11	-
IEEE Xplore	32300	13992	9630	2643	-	-	-	-
EBSCO	3366	1104	2024	657	-	-	-	-

Fonte: Elaborado pelo autor

Uma pesquisa inicial foi feita sem considerar um período específico, usando as palavras-chave encontradas com mais frequência nos artigos que descrevem trabalhos relacionados ao ensino e aprendizagem de algoritmos. A combinação de termos como programação e ensino e/ou aprendizado se mostrou a mais eficaz, a que retornou um número significativo de artigos que realmente tem como tema principal as pesquisas relacionadas ao ensino e aprendizado de programação. Posteriormente repetiu-se a pesquisa considerando somente os últimos cinco anos, ou seja, publicações com ano igual ou maior que 2014 (últimos cinco anos).

O grande número de publicações retornado demonstra que existem diversos grupos e pesquisadores interessados na pesquisa sobre o ensino e aprendizado de programação. Os números apresentados contribuem para demonstrar o interesse crescente pelo ensino e aprendizagem de programação.

Iniciativas para identificar dificuldades e problemas enfrentados pelos estudantes em disciplinas de programação são antigas. Ulloa (1980) apresenta um estudo sobre os problemas verificados no processo de aprendizagem, estabelecendo uma relação com as diferentes estruturas de linguagens e métodos utilizados. Ainda década de 80, Solloway e Ehrlich (1984) realizaram uma pesquisa que teve como objetivo identificar os conhecimentos empregados por programadores experientes no desenvolvimento de programas. Nowaczyk (1984) utilizou um conjunto de testes para avaliar a performance de programadores iniciantes e para tentar identificar as habilidades relacionadas a capacidade de resolução de problemas. Os testes mostraram que a falha na compreensão e resolução de problemas aliada às dificuldades de leitura e ao baixo conhecimento em matemática poderia indicar uma tendência de insucesso nestas disciplinas.

Muitas destas dificuldades independem da linguagem de programação, do paradigma de programação utilizado e, em muitos casos estão diretamente relacionadas às estratégias ou à capacidade de compreender um problema e planejar a elaboração de uma solução. Diversos autores têm investigado as dificuldades na aprendizagem de algoritmos e programação, especialmente aquelas apresentadas por programadores iniciantes (MAYER, 1981; JENKINS, 2002; LAHTINEN, ALA-MUTKA e JÄRVINEN, 2005; GOMES e MENDES, 2007, DA CUNHA, TONETTI e SANAVRIA, 2017; BOSSE e GEROSA, 2017, FIGUEIREDO e GARCÍA-PEÑALVO, 2018). Gomes e Mendes (2007) descrevem estes problemas sob as seguintes perspectivas: os métodos de ensino, estratégias de estudo, habilidades e atitudes demonstradas pelos alunos, natureza da programação e os aspectos psicológicos.

Os métodos de ensino são considerados muitas vezes inadequados por não serem personalizados, o que dificulta um acompanhamento e feedback e pode acarretar em dificuldades para entender como cada aluno desenvolve o raciocínio. Outra crítica diz respeito ao foco excessivo na linguagem de programação e não no desenvolvimento de habilidades para resolução de problemas computacionais. Com relação às habilidades e atitudes demonstradas pelos estudantes, destaca-se a incapacidade de compreender e resolver problemas (GOMES e MENDES, 2007). Segundo os autores, em muitos casos os discentes tentam resolver um problema, sem compreendê-lo completamente, principalmente pela dificuldade de interpretar o

mesmo. Relacionar um problema com situações anteriores, com conhecimentos já adquiridos são questões fundamentais que precisam ser consideradas antes de tentar construir um algoritmo ou programa.

A forma que são apresentadas as questões, a descrição e os enunciados influenciam diretamente na compreensão do problema, no tempo para resolução das tarefas, bem como no número de tentativas de executar a solução. Estudo conduzido por Teusner, Hille (2018), com 2400 estudantes mostrou que o nível de detalhe, a clareza na definição do problema e a disponibilização de exemplos ou dicas impactam diretamente na dificuldade demonstrada na realização das atividades.

Jenkins (2002) aponta diversas causas para o insucesso nas disciplinas de programação, entre elas a menor capacidade de abstração, a pouca aptidão para a resolução de problemas, além do uso de métodos pedagógicos pouco adequados ou que considerem os diferentes estilos de aprendizagem dos estudantes. Outro aspecto que pode afetar o aprendizado diz respeito as sintaxes e estruturas das linguagens de programação, consideradas inadequadas para aprendizes.

Disciplinas de programação em grande parte dos cursos fazem parte dos conteúdos iniciais, fundamentais e estão no início de um curso superior, sendo esse um momento em que o aluno tem dificuldades, onde tudo é novo e o próprio ingresso na universidade representa uma mudança na vida pessoal. Inserida neste contexto está a programação, que é formada por conhecimentos, conteúdos complexos, o que agrava as dificuldades em um período de transição (GOMES, HENRIQUES e MENDES, 2008). Tal cenário leva a necessidade de considerar a dimensão afetiva no aprendizado, uma vez que as dificuldades pessoais enfrentadas pelos alunos podem influenciar significativamente o desempenho.

É possível encontrar diversos trabalhos que apresentam revisões ou mapeamentos sistemáticos sobre as dificuldades, estratégias e recursos utilizados no ensino e aprendizagem de programação (PEARS, 2007; SHEARD et al., 2009; VIHAVAINEN, AIRAKSINEN e WATSON, 2014; KUNKLE e ALLEN, 2016; QIA e LEHMAN, 2017; DRINI, 2018).

Sheard et al. (2009) apresenta uma análise das publicações nas principais conferências relacionadas ao ensino de computação no período de 2005 a 2008. Foram analisadas as metodologias utilizadas para coleta e análise de dados, além dos temas principais de pesquisa. A metodologia predominante foi a quantitativa, embora a combinação com abordagens qualitativa também seja observada em boa parte das pesquisas.

O artigo mostra ainda que a grande maioria dos trabalhos concentra-se na tentativa de identificar habilidades ou competências necessárias ao aprendizado e demonstradas pelos estudantes (66%). A investigação sobre técnicas e estratégias usadas no ensino para auxiliar os estudantes foi tema de 35% das pesquisas, seguida pelo desenvolvimento e aplicação de ferramentas com 9%. Na visão dos autores (SHEARD et al, 2009) ainda são necessárias mais investigações relacionadas as teorias e modelos educacionais, que foram observados em apenas 6% dos casos.

Entre as habilidades citadas por Sheard et al. (2009) estão a compreensão de problemas, a capacidade de abstração ou criação de um modelo mental que permita especificar uma solução. Estas podem ser relacionadas ao saber conhecer e saber fazer, citadas por Behar (2013).

O trabalho desenvolvido por Pears (2007) apresenta três questões essenciais: quais linguagens de programação são usadas? Que ferramentas e ambientes suportam o aprendizado e como são aplicados? Quais estratégias pedagógicas são utilizadas e que resultados advém destas? Abordagens pedagógicas que focam no desenvolvimento da capacidade e habilidades de resolução de problemas estão entre as mais comuns de acordo com os autores, entretanto, a abordagem mais tradicional ainda concentra-se na sintaxe e na estrutura da linguagem.

Embora numerosos estudos apontem para estratégias promissoras que podem ser adotadas no ensino de programação, poucos deles se propõe a avaliar de forma quantitativa o impacto das diferentes estratégias. Estudo descrito por Vihavainen, Airaksinen e Watson (2014) analisou quantitativamente as intervenções e os princípios pedagógicos utilizados nestas. Quanto ao tipo da intervenção, as mesmas foram agrupadas e entre os aspectos mais relevantes estão:

- Colaboração: estratégias que encorajam alunos a colaborar entre si.
- Contextualização: alinhamento das atividades e conteúdos às situações reais ou contextos específicos.
- Temática de jogos: utilização de jogos ou contextos baseados em games.
- Alterações ou atualização nos materiais utilizados.
- Suporte em pares ou grupos de trabalho.
- Cursos ou treinamentos preliminares: criação de cursos ou treinamentos introdutórios, que, em sua maioria utilizam ambientes de apoio com Scratch ou Alice.

Segundo os autores, métodos baseados na contextualização das situações-problema e na colaboração estão entre os mais utilizados e demonstram bom potencial para melhorar o ensino de programação. A combinação de diferentes métodos e recursos em estratégias híbridas pode se constituir em uma alternativa promissora (VIHAVAINEN, AIRAKSINEN e WATSON, 2014).

Um comparativo entre métodos tradicionais de ensino e baseados na Aprendizagem Ativa é encontrada no estudo desenvolvido por Drini (2018). Em diferentes períodos (2013 a 2014 e 2014 a 2016) foram aplicados respectivamente métodos tradicionais de ensino e em uma segunda etapa uma aprendizagem ativa com o intuito de verificar o seu impacto na motivação dos estudantes para aprender. Os resultados mostraram que a aprendizagem ativa contribuiu para o aumento da capacidade dos estudantes de pensarem de forma lógica, além de mostrar um maior engajamento e uma participação mais ativa na resolução das tarefas.

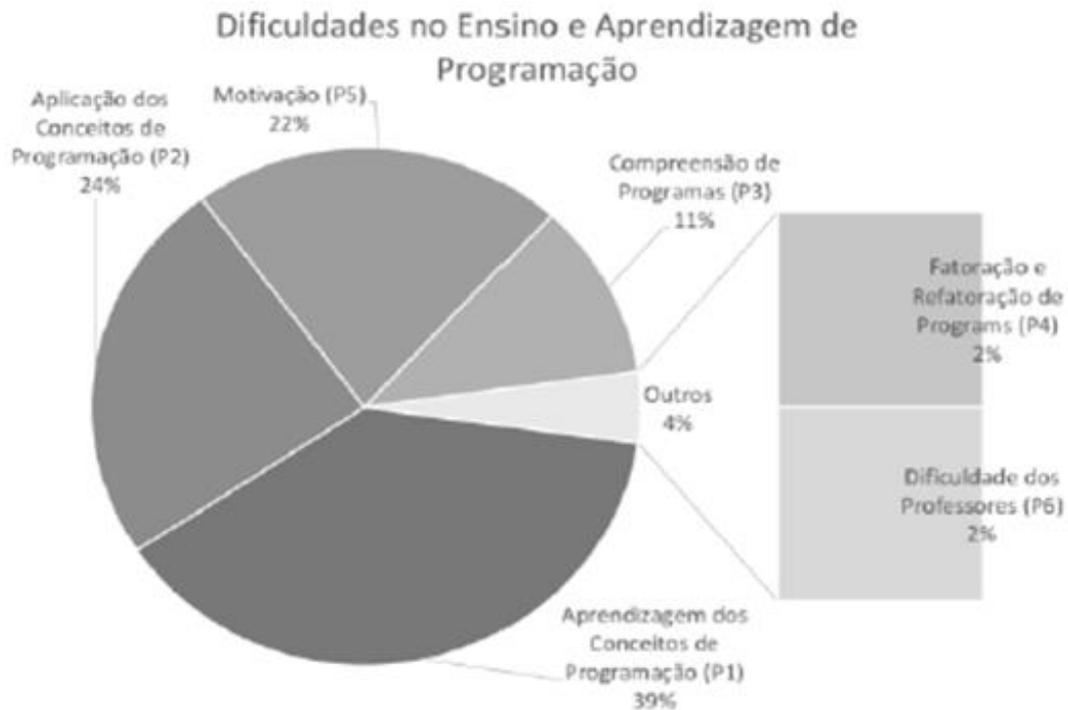
A realidade brasileira é retratada em diversas publicações (JÚNIOR e RAPKIEWICZ, 2004; DA SILVA RIBEIRO et al., 2012; AURELIANO e TEDESCO, 2012; RAMOS et al., 2015; MICAEL SOUZA et al., 2016, DA CUNHA, TONETTI e SANAVRIA, 2017; BLATT, BECKER e FERREIRA, 2017; DE BARROS COSTA e ROCHA, 2018). Uma análise dos trabalhos publicados nos anais do Simpósio Brasileiro de Informática na Educação (SBIE) e Workshop de Educação em Computação (WEI), no período de 1999 a 2003 é apresentada por Júnior e Rapkiewicz (2004). No período abordado, constatou-se que o número de trabalhos era crescente o que denotava, já naquele momento a preocupação com o ensino de programação.

Levantamentos recentes indicam que esta tendência de aumento ou na consolidação do número de publicações relacionadas à programação de computadores permanece (MARCELINO e KEMCZINSKI, 2018; STADELHOLFER e GASPARINI, 2018). Nas pesquisas, na maioria dos casos os autores utilizaram termos como *programming teaching*, *learning* e bases como IEEE Xplore, ACM e Scopus, na busca pelas publicações relacionadas.

Uma revisão do trabalho de Júnior e Rapkiewicz (2004) pode ser encontrado no mapeamento realizado por Da Silva Ribeiro et al. (2012). Foram considerados os trabalhos publicados no SBIE, WEI e WIE no período de 2004 a 2011. As iniciativas foram distribuídas em três categorias: i) ferramentas; ii) métodos de ensino e; iii) ferramentas associadas aos métodos. Na primeira categoria os autores destacavam o uso de tutores inteligentes, gerenciadores de conteúdo, além de fundamentos da computação afetiva. Dentre os métodos citados é apontado a busca por formas diferenciadas de avaliação e o uso da Taxonomia de

Bloom. A combinação de ferramentas com métodos de ensino inclui o uso de sistemas baseados em jogos, além de ambientes para o acompanhamento do aprendizdo.

Figura 2 – Categorização dos problemas de pesquisa



Fonte: Micael Souza et al. (2016)

Micael Souza et al. (2016) apresenta um levantamento sobre os problemas e dificuldades apresentadas pelos estudantes e as iniciativas ou soluções propostas, que foram categorizadas pelos autores (Figura 2). Trabalhos recentes trazem igualmente a preocupação em identificar as dificuldades e as soluções propostas (MARCELINO e KEMCZINSKI, 2018; STADELHOLFER e GASPARINI, 2018). Com relação aos problemas e dificuldades, destacam-se as seguintes categorias:

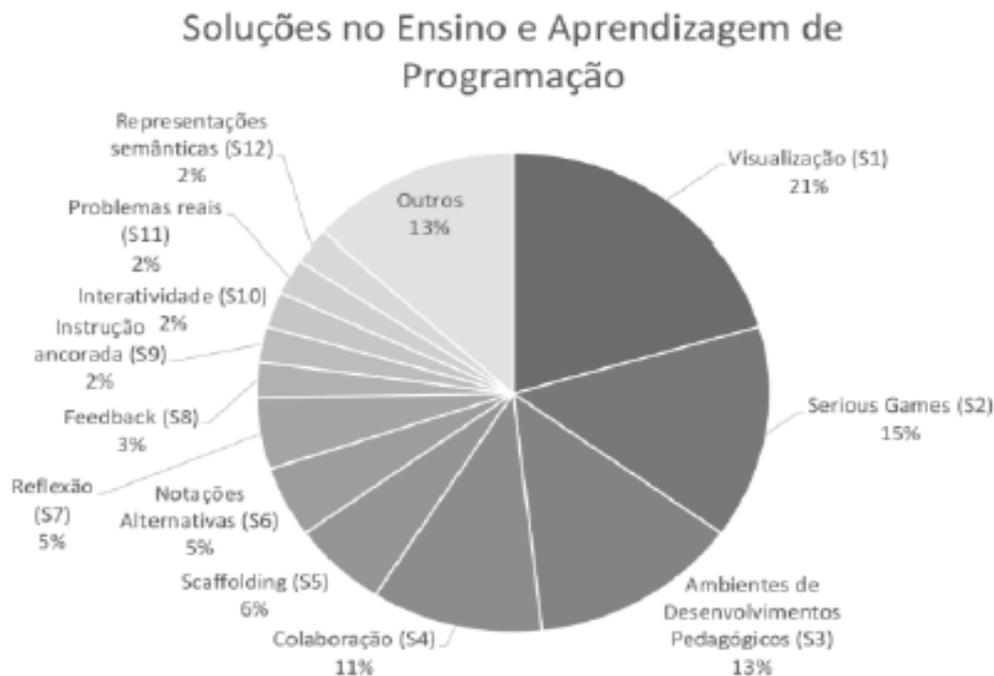
- Aprendizagem de conceitos: dificuldades relacionadas ao aprendizado de conceitos como estruturas de repetição, classes, ponteiros, entre outros.
- Aprendizagem de conceitos de programação: dificuldades para utilizar os conceitos na construção de programas.
- Motivação: falta de interesse, de motivação ou desânimo demonstrado pelo estudante na realização das tarefas.
- Compreensão de programas: dificuldade para ler e compreender as soluções.

Constatou-se uma predominância de problemas associados à dificuldade de aprender os fundamentos da programação, estruturas e sintaxe da linguagem e de aplicar estes conceitos na construção das soluções (Figura 2). Além destes a motivação também aparece como uma

dificuldade comum e em boa parte dos casos está relacionada à dificuldade de aprender e aplicar os fundamentos da programação (MICAEL SOUZA et al., 2016).

Analisando a distribuição das soluções propostas, percebe-se que categorias como a visualização de programas de forma interativa e lúdica e o uso de jogos se destacam (Figura 3). Em seguida aparecem o uso de ambientes baseados em estratégias pedagógicas e a colaboração (MICAEL SOUZA et al., 2016). A reflexão sobre experiências e conhecimentos anteriores e a utilização de problemas reais aparecem com percentuais inferiores, o que pode indicar que ainda são necessárias mais pesquisas para avaliar se estas abordagens podem contribuir para melhorar o processo de ensino e aprendizagem de programação.

Figura 3 – Categorização das soluções apresentadas



Fonte: Micael Souza et al. (2016)

Levantamento apresentado por Blatt, Becker e Ferreira (2017) mostra a evolução das pesquisas sobre os métodos e ferramentas para o ensino de programação, com destaque para o crescimento de alternativas que privilegiem abordagens interdisciplinares e a gamificação. Demonstram que ferramentas que utilizam interfaces lúdicas e visuais, como o Scratch tem sido utilizadas com frequência cada vez maior. Uma mudança citada pelos autores é o crescimento do ensino de algoritmos e programação na educação básica, o que pode contribuir com o passar do tempo para aumentar o conhecimento prévios dos estudantes na graduação.

De Barros Costa e Rocha (2018) descrevem uma mapeamento sistemático sobre iniciativas que utilizam a resolução de problemas aliada à mecânica de jogos como estratégia

para o ensino de programação. O trabalho apresenta a programação como uma tarefa que visa resolver um problema e um programa como a expressão da solução para este problema. Os resultados derivam de artigos coletados nos principais eventos (SBIE, WIE, WEI, WCBIE) e periódicos (RENOTE, RBIE) brasileiros, no período de 2008 a 2017.

Na pesquisa são apontadas diversas lacunas que segundo os autores merecem mais investigação, entre estas estão a disponibilização de recursos e ferramentas que contribuam para que os estudantes compreendam o enunciado do problema, além da necessidade de dicas e feedback durante a elaboração da solução (DE BARROS COSTA e ROCHA, 2018). As principais dificuldades percebidas no estudo, segundo os autores continuam sendo a dificuldade de compreender o problema e planejar a solução antes da elaboração do código.

A partir da contextualização da realidade do ensino de programação constata-se que a preocupação com as dificuldades apresentadas pelos estudantes é crescente, mas as iniciativas são variadas e a maioria dos problemas e estratégias carecem de investigações mais aprofundadas. É possível perceber que a combinação de métodos de ensino com ambientes de aprendizado, como os que apoiam as atividades de programação e que contribuam para um acompanhamento das atividades pode ser uma abordagem promissora. Aliar a estes a dimensão afetiva para auxiliar professores na identificação de baixos níveis de motivação pode também contribuir para o aprendizado de programação, especialmente no caso de iniciantes.

Diante deste cenário esta pesquisa teve como foco a aplicação de um método baseado na Aprendizagem Ativa, apoiado por um sistema automatizado que permite propor e resolver problemas de programação. A dimensão afetiva foi contemplada pelo desenvolvimento de um modelo para o reconhecimento do esforço e confiança demonstrados pelos estudantes na resolução das tarefas.

2.2 Aprendizagem Ativa

A busca por uma educação que envolva de forma ativa os estudantes no processo de ensino e aprendizagem não é recente. Juan Díaz Bordenave em Estratégias de Ensino-Aprendizagem (1982), discute sobre o que ele chama de educação baseada na transmissão de conhecimento, na qual observa-se um aluno passivo, memorizador e com baixa capacidade de resolver problemas. Em oposição é apresentada a ideia de educação problematizadora, na qual o aluno se torna um sujeito ativo do processo, a partir do estímulo para a reflexão sobre problemas.

Embora não exista um consenso sobre a origem da expressão Aprendizagem Ativa, diversos autores estabelecem uma relação entre esta definição e os trabalhos de John Dewey

(1959). O “pensar reflexivo” na visão de John Dewey está relacionado a um estado de dúvida, que leva ao questionamento e à investigação. Segundo o autor:

Notaremos que o pensar reflexivo, diferentemente das outras operações a que se dá o nome de pensamento abrange: (1) um estado mental de dúvida, hesitação, perplexidade, dificuldade mental, o qual origina o ato de pensar; e (2) um ato de pesquisa, procura, inquirição, para encontrar material que resolva a dúvida, assente e esclareça a perplexidade. (DEWEY, 1959, p.22)

Bonwell e Eisel (1991) apresentam um conjunto de características que geralmente estão associadas com o conceito de Aprendizagem Ativa. Menor ênfase na transmissão de informações, desenvolvimento de atividades relacionadas à síntese, avaliação e análise são exemplos de características. De acordo com o autor, estudantes devem exercitar e explorar suas próprias capacidades, atitudes e valores. Segundos os autores “Envolve estudantes que estão realizando algo e pensando sobre aquilo que estão fazendo”.

A simples introdução de atividades práticas em meio a uma estratégia de ensino tradicional pode não ser suficiente (PRINCE, 2004). É necessário propor situações que promovam o engajamento ou envolvimento do estudante, de forma efetiva, uma vez que este é um dos aspectos centrais da aprendizagem ativa. Prince (2004) apresenta diversas evidências e estudos que demonstram que uma participação ativa dos estudantes pode ser um indicativo do sucesso.

Na literatura é possível encontrar diversas classificações para as técnicas ou métodos cujas concepções associam-se à Aprendizagem Ativa. Prince e Felder (2006) descrevem um conjunto de métodos que denominam como “Métodos indutivos de ensino e aprendizagem”. Métodos instrucionais como o aprendizado baseado em problemas, em projetos, em casos, além da aprendizagem investigativa são definidos como indutivos. Outra forma de agrupar e classificar estes métodos é associando-os a definição conhecida como “Metodologias de ensino e aprendizagem ativas”.

Segundo Berbel (2012), as metodologias de ensino e aprendizagem ativas constituem-se em abordagens que estimulam o envolvimento e a participação ativa dos estudantes no processo, desenvolvendo a autonomia e a capacidade de reflexão sobre as situações vivenciadas. Estas contribuem para aumentar a curiosidade e motivação, uma vez que aproximam o estudante da realidade, fortalecendo a percepção sobre a importância da compreensão e reflexão sobre os problemas apresentados.

Em que pese a existência de diferentes formas de conceituar ou classificar estes métodos, todos os autores convergem para a mudança de uma abordagem centrada na

transmissão de conhecimento para uma estratégia baseada na participação ativa dos alunos (PRINCE e FELDER, 2006; BERBEL, 2011; MORAN, 2018).

Entre as metodologias caracterizadas como ativas, destacam-se aquelas que tem como foco principal o estudo baseado em problemas. Duas abordagens que utilizam problemas como o ponto de partida para o ensino e aprendizagem podem ser consideradas as mais citadas, a Aprendizagem Baseada em Problema (ABP) ou *Problem Based Learning* (PBL) e a Metodologia da Problematização (MP). Em ambos os casos considera-se que o conhecimento obtido pelo aluno lhe permite a resolução dos problemas propostos (GOMINHO, 2014).

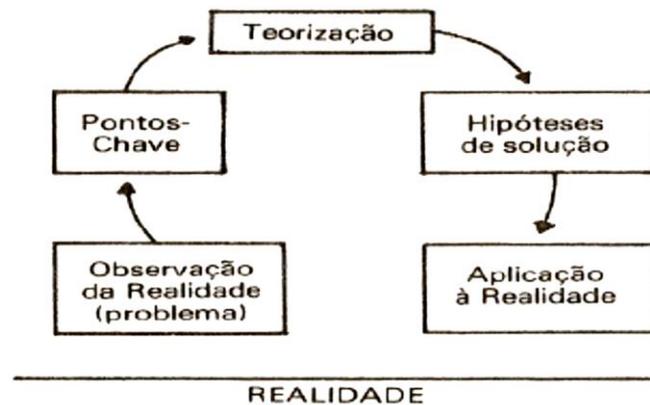
Embora os métodos citados utilizem o mesmo princípio, que é o ensino por problemas, Berbel e Gamboa (2011) identificam diferenças na aplicação destes. O ensino baseado em problemas em muitos casos se constitui em um direcionamento da organização curricular, para um curso inteiro ou para uma ou mais disciplinas. Os problemas na maioria dos casos são definidos por um ou mais professores em conjunto com os estudantes e norteiam todo o processo de ensino e aprendizagem da disciplina ou do curso.

As origens da aprendizagem baseada em problemas remetem as Universidades McMaster, no Canadá, com a implantação pioneira sendo realizada na Faculdade de Ciências Médicas, porém a referência teórica mais conhecida é a Universidade de Harvard (PEREIRA, 2007). Observa-se que a origem destas abordagens é a área médica, cuja preocupação historicamente é na responsabilidade social dos estudantes, porém, a adoção em outras áreas do conhecimento, como as ciências exatas, engenharias, computação é cada vez mais frequente (QIU e CHEN, 2010; MEIRELES e BONIFÁCIO, 2015; CACEFFO, GAMA e AZEVEDO, 2018).

A ABP apoia-se na ideia de pequenos grupos de estudantes, para os quais é colocada uma situação-problema. O tutor ou mediador é responsável por orientar os estudantes a pensar de forma lógica e racional (PEREIRA, 2007). Seguem etapas que iniciam pela definição e análise do problema, pela coleta de dados sobre o mesmo e pela sintetização destas informações. A definição dos grupos e as etapas utilizadas no processo podem variar conforme o contexto e à amplitude do problema.

A problematização teve como um dos primeiros métodos propostos o Arco de Magueres, ilustrado na figura 4. Uma das referências iniciais ao método é encontrada em Bordenave e Pereira (1982). Nela se enfatiza a ideia de que a aprendizagem deve acontecer a partir de uma visão global de um problema para uma visão analítica, que equivale à compreensão do mesmo.

Figura 4 – Arco de Maguerez



Fonte: Bordenave e Pereira (1982)

A situação-problema diz respeito à realidade ou a um recorte desta, sendo possível abordar desde problemas mais complexos aos mais simples, de acordo com a proposta do estudo. O problema é definido a partir da observação da realidade ou da contextualização desta por parte do professor em conjunto com os estudantes.

A definição dos pontos-chave do problema permite uma compreensão inicial dos principais elementos que o definem. Estabelece-se neste momento uma reflexão sobre o problema, a determinação da sua complexidade e as variáveis associadas. Busca-se identificar quais conceitos devem ser pesquisados ou revisados para a proposição de alternativas que possam solucioná-lo (BERBEL, 2012). A identificação dos pontos-chave em muitos contextos é uma atividade individual, na qual cada estudante tenta estabelecer a sua percepção sobre o problema, para, posteriormente, discutir com o restante do grupo.

A teorização é o momento no qual os alunos devem pesquisar, estudar sobre os conceitos e elementos principais do problema, buscando aumentar o conhecimento associado ao mesmo. As pesquisas podem ocorrer utilizando recursos tradicionais como livros, manuais ou ferramentas tecnológicas, como buscadores, *Ebooks*, entre outros. (BERBEL e GIANNASI, 1999). Em muitas situações, o conhecimento pode vir do material de ensino da própria disciplina, disponibilizado ou sugerido pelo professor, sob a forma de exemplos, páginas, apostilas, vídeos, entre outros.

A reflexão e estudos devem fornecer subsídios para a elaboração de possíveis soluções, ou hipóteses de solução para o problema. Quais recursos serão utilizados, quais sequências de passos são adequadas para elaborar a solução, que técnicas serão usadas para elaborar, testar e validar a solução, são exemplos de questionamentos comuns na etapa denominada “Hipóteses de solução” (BERBEL e GAMBOA, 2011). A discussão e interação entre os participantes e

professor é estimulada especialmente nesta etapa, momento no qual as diferentes hipóteses são analisadas. Na etapa final, o aluno tem a oportunidade de colocar em prática o que aprendeu durante as etapas anteriores, é o momento de aplicar à realidade, de construir a solução e analisar os efeitos ou resultados da sua aplicação.

Em relevante artigo intitulado “A metodologia da problematização com o Arco de Maguerez: Uma perspectiva teórica e epistemológica” (2012), Neusi Berbel busca responder a seguinte questão: levando em conta as características do Esquema do Arco, de Charles Maguerez, o que o fundamenta? Que teorias da educação e/ou concepções de conhecimento pode melhor explicá-lo? A autora constata que, dentre as diferentes concepções, é possível associar a problematização a:

[..]a pedagogia construtivista (Piaget) - pela preocupação em conhecer e em descobrir como ativar os esquemas de pensamento dos grupos de adultos; pela disponibilização de variadas ferramentas de raciocínio no sentido de favorecer a promoção profissional; pela preocupação em utilizar material concreto para favorecer a aprendizagem; e com outras, relacionadas com nuances da pedagogia escolanovista. (BERBEL, 2012, p. 274).

Além desta, destaca-se a relação com:

[...] a pedagogia tradicional - percebida como associada pelo planejamento, tido como estruturação prévia do programa de ensino; pela insistência na importância da sequência pedagógica simbolizada pelo arco; pela previsão de questões a apresentar obrigatoriamente aos alunos; pelas respostas que se espera obter e a explicação a que pode eventualmente o monitor se reportar; pela lista dos termos e das formas de síntese que devem ser destacadas, repetidas, escritas no quadro, lidas e copiadas; por um exame final muito completo, entre outras razões. (BERBEL, 2012, p. 274).

Uma busca, sobre publicações associadas à problematização, indica que pesquisas com o foco nas ciências exatas, como a computação, são menos comuns, mas têm crescido nos últimos anos (PRADO S. JR., FONTENELE e RODRIGUES DA SILVA, 2014; MACHADO e PINHEIRO, 2009, UNTERLEIDER, 2010; DE BARROS COSTA e ROCHA, 2018). Estudos recentes tem demonstrado que as metodologias ativas contribuem para melhorar o desempenho dos estudantes em cursos ou disciplinas nas áreas de ciências exatas (QIU, 2010; FREEMAN, 2014).

Freeman (2014) apresenta uma análise da performance de estudantes de graduação nos cursos de Matemática, Tecnologia e Engenharia comparando a aplicação de metodologias tradicionais com estratégias baseadas na Aprendizagem Ativa. A principal conclusão do estudo

é que houve um incremento considerável na performance dos estudantes e que o uso sistemático da Aprendizagem Ativa ao longo do tempo é um aspecto importante para o sucesso desta. Dentre os métodos citados pelos autores estão as técnicas baseadas na resolução de problemas e na colaboração.

Estudo realizado por Witt, Kemczinski e Dos Santos (2018) descreve a evolução das pesquisas que utilizam métodos baseados em problemas no ensino de computação. Consta-se um crescimento nos últimos anos da quantidade de publicações associadas ao tema, como pode ser observado na figura 5. As pesquisas citadas são divididas em dois grupos, as que utilizam a aprendizagem baseada em problemas e aquelas que se baseiam na metodologia da problematização.

Figura 5 – Quantidade de publicações sobre métodos baseados em problemas



Fonte: Witt, Kemczinski e Dos Santos (2018)

O estudo mostra que a maioria dos trabalhos utiliza a aprendizagem baseada em problemas e que a área de ensino que apresenta mais iniciativas é o ensino de algoritmos, programação ou a análise e desenvolvimento de sistemas (WITT, KEMCZINSKI, DOS SANTOS, 2018). A menor concentração de trabalhos que investiguem a metodologia da problematização indica que ainda é necessário investigar de forma mais aprofundada a viabilidade do uso desta na computação, e em especial no ensino de programação.

Um aspecto a ser destacado nas abordagens ativas, especialmente nos métodos baseados em problemas, é a aproximação do processo de ensino e aprendizagem com situações reais, com situações conhecidas. Ao analisar os enunciados de exercícios de programação em disciplinas introdutórias é possível constatar que predominam situações nas quais o problema é definido de maneira direta, sem uma contextualização e que os temas mais utilizados estão ligados à matemática (ZANINI e RAABE, 2012). O uso da problematização pressupõe a exposição de um problema relacionado às situações reais, que estimulem o aluno a buscar o conhecimento necessário para a sua compreensão.

A partir das considerações expostas, é possível inferir que a Aprendizagem Ativa baseada na problematização pode se constituir em um método consistente, cuja aplicação pode ser sistematizada e utilizada para estimular o raciocínio lógico e a resolução de problemas, tais como os encontrados nas disciplinas de programação.

Na definição da estratégia utilizada nesta pesquisa optou-se por um método que emprega as etapas definidas na problematização, a identificação dos pontos-chave e a formalização da hipótese de solução com o objetivo de estimular o estudante a refletir sobre o problema e elaborar uma organização lógica antes de construir a solução. As situações-problema são expostas de maneira contextualizada, evitando um direcionamento para uma determinada solução ou uma definição dos recursos que devem ser aplicados para solucioná-las.

É necessário ressaltar que o sucesso do método depende do esforço e dedicação dos estudantes. A motivação do aluno é condição essencial para desenvolver a autonomia e a capacidade de tomar decisões durante o processo de resolução de problemas (BERBEL, 2012). O uso da problematização tem como objetivo motivar os discentes, entretanto, não é possível garantir que a simples utilização do método é suficiente para atingir este objetivo. Um acompanhamento das ações dos estudantes, da forma que estes problematizam, da sua trajetória durante o processo é importante para avaliar os resultados obtidos.

Uma das abordagens que pode ser usada para observar e analisar os sujeitos durante as etapas da problematização, é a que considera a dimensão afetiva, ou, a afetividade. Considerando esta realidade, a próxima seção descreve conceitos relacionados à motivação e os fundamentos da computação afetiva, que é a área de pesquisa responsável por conferir aos sistemas computacionais recursos para reconhecer de forma automática os estados emocionais ou afetivos.

2.3 Dimensão afetiva

Esta seção descreve os fundamentos da Computação Afetiva (CA), que pode ser definida como a área da computação que agrega as pesquisas relacionadas à afetividade, os fundamentos e aplicações que visam capacitar os computadores para identificar, compreender e representar estados afetivos (PICARD, 1997).

A Computação Afetiva têm despertado interesse de pesquisadores na área educacional por permitir o desenvolvimento de ferramentas computacionais que permitam modelar e analisar os estados afetivos dos estudantes, contribuindo desta forma para a tomada de ações que visam minimizar o baixo desempenho escolar (JACQUES e NUNES, 2012). Compreender as emoções humanas e modelar as mesmas em sistemas computacionais constitui-se em um dos

objetivos da área de CA, sendo também um dos principais temas que norteou o desenvolvimento da presente pesquisa.

Para uma compreensão melhor da Computação Afetiva e os princípios adotados nesta pesquisa, serão apresentados os conceitos associados à afetividade, além das definições sobre emoção, sentimento e estados afetivos. Segundo Bercht (2001) afetividade pode ser definida como o domínio das emoções, dos sentimentos das experiências sensíveis e, principalmente da capacidade em se poder entrar em contato com sensações.

As emoções por sua vez podem ser caracterizadas como as reações ao mundo, determinadas tanto por um caminho de resposta já construído biologicamente como por mecanismos menos automáticos, mediados por significados e cognições. Uma emoção é desencadeada a partir de uma situação significativa de conteúdo cognitivo, possui grau de intensidade, ocorre em um curto espaço de tempo e influencia o comportamento e as ações de um indivíduo (LONGHI, 2011).

O sentimento é a percepção de todas as mudanças que constituem a resposta à emoção ou à diferentes ações e construções engendradas pelas emoções. O sentimento está diretamente relacionado à emoção, pois toda a emoção gera um sentimento, mas nem todo o sentimento advém de uma emoção (BERCHT, 2001). O sentimento pode caracterizar um comportamento que é muitas vezes “invisível” para outros, mas é perceptível para quem o têm (DAMÁSIO, 2004).

Como exemplo é possível ilustrar uma avaliação de uma atividade recebida por um estudante, que, se positiva desencadeia uma emoção de alegria que é de intensidade alta, podendo durar alguns minutos ou até mesmo horas (SCHERER, 2005). O desempenho positivo em uma atividade, na qual o estudante percebeu que teve sucesso na resolução pode gerar sentimentos positivos que contribuem para uma maior motivação, que se caracteriza como uma condição ou situação que pode ser mais duradoura.

Um estado afetivo pode ser definido como um conjunto de características que determinam a emoção, os sentimentos e afetividade em um indivíduo, em determinado momento. Um estado da mente e do corpo que caracteriza a emoção e os sentimentos em determinado instante (BERCHT, 2001). Estado afetivo é um termo comum associado à afetividade, que pode incluir as emoções, posturas interpessoais, além de traços de personalidade (SANTOS, 2016; SCHERER, 2005).

A preocupação com a compreensão da afetividade é antiga, desde os tempos mais remotos, filósofos como Platão e Aristóteles estabeleceram distinções claras entre a razão e a

emoção ou a dicotomia entre mente e cérebro (LONGHI, 2011). Contrapondo esta visão Damásio (2009) argumenta que a razão pode não ser tão “pura” quanto a maioria das pessoas acredita que seja e que as emoções influenciam os atos considerados racionais, de maneira positiva ou negativa.

Os estudos de Damásio dedicam-se a descrever o papel das emoções e sentimentos na formação da consciência dos seres humanos e na influência que exercem na tomada de decisões (DAMASIO, 2004, 2009). Em um ambiente educacional, por exemplo, decisões tomadas durante a resolução das tarefas ou no enfrentamento das dificuldades são influenciadas pelo estado afetivo do estudante, sendo assim, a afetividade deve ser considerada pelo docente nas ações pedagógicas.

Ao longo dos anos, a pesquisa sobre afetividade experimentou diferentes abordagens e classificações. A perspectiva cognitivista, que influencia o presente trabalho indica que as emoções estão relacionadas ao modo que os indivíduos avaliam os eventos, tanto externos, quanto internos. A afetividade, portanto, é influenciada pelo significado do evento para cada pessoa e está relacionada aos objetivos e motivações de cada um (PICARD, 1997; DAMÁSIO, 2009). A abordagem cognitivista tem influenciado ao longo dos anos os estudos sobre a afetividade em áreas como Psicologia, Educação e tem sido importante para as pesquisas em Computação Afetiva. A utilização de técnicas computacionais baseadas na Inteligência Artificial, tais como as que foram empregadas nesta pesquisa, encontra forte embasamento nas ideias da abordagem cognitivista.

Scherer (2005) argumenta que a emoção é um processo caracterizado por uma sequência de avaliações que geram mudanças em subsistemas orgânicos dinâmicos. As mudanças no estado afetivo acontecem a partir de um monitoramento contínuo dos eventos que ocorrem em ambientes físicos e sociais. Cada evento é analisado sob a perspectiva da sua relevância, expectativa e implicações no bem estar do indivíduo e após esta avaliação o mesmo é relacionado à crença na capacidade de enfrentamento ou de responder ao evento que desencadeou o processo (SCHERER, 2005). Tal abordagem está ligada ao conceito de fenômenos afetivos que segundo Scherer (2005) podem possuir diferentes origens, intensidade, duração e impacto comportamental.

Segundo Santos (2016) tanto o estado afetivo, quanto o fenômeno afetivo podem ser considerados como um conjunto de condições psicológicas e biológicas associadas à afetividade e impactam na maneira que uma pessoa encara os eventos. É possível inferir,

portanto, que dados relacionados ao comportamento e às ações dos estudantes durante a resolução dos problemas podem ser indicadores do estado afetivo em que estes se encontram.

Estado afetivo é, portanto, um termo abrangente que pode caracterizar diferentes estados ou fenômenos ligados à afetividade, tais como estados de ânimo, personalidade, frustração e até mesmo a motivação (IEPSEN, 2013; BERCHT, 2001). O presente trabalho considerou a motivação como um estado afetivo e investigou um conjunto de técnicas e recursos que permitiram modelar e reconhecer este estado afetivo em um contexto educacional.

A importância da motivação e de outros estados afetivos na aprendizagem têm sido enfatizada por diversos pesquisadores e pedagogos (MORAIS et al, 2017). A motivação é considerada um fator fundamental para a aprendizagem, alunos motivados buscam encontrar respostas para os seus problemas, além de tentar satisfazer suas necessidades pessoais (JAQUES e NUNES, 2012). Conferir a um sistema recursos para identificar que um estudante se encontra desmotivado, por exemplo, pode permitir a indicação de ações ou tarefas alternativas, durante a resolução de um problema.

Picard et al. (2004) destaca que o uso da Computação Afetiva na educação visa corrigir um desequilíbrio decorrente do uso de recursos computacionais como ferramenta de apoio ao ensino, que privilegia o cognitivo sobre o afetivo. O afeto está diretamente ligado à cognição e ambos são importantes para orientar o comportamento racional, que inclui a memória, a criatividade e a capacidade de resolver problemas (PICARD et al., 2004).

Conforme Rosalind Picard (1997) um sistema computacional pode expressar ou possuir emoções, além de reconhecer estados afetivos. As diferentes linhas de pesquisa e modelos focam em uma determinada funcionalidade (reconhecer, expressar) ou na combinação destas. Expressar consiste em instruções para demonstrar emoções a outros agentes, que podem ser através de mudanças sonoras, imagens, atitudes, entre outros. Um sistema que simula possuir emoções ou estar em um determinado estado afetivo, deve ser capaz de simular um comportamento emocional mais amplo, com diferentes expressões afetivas.

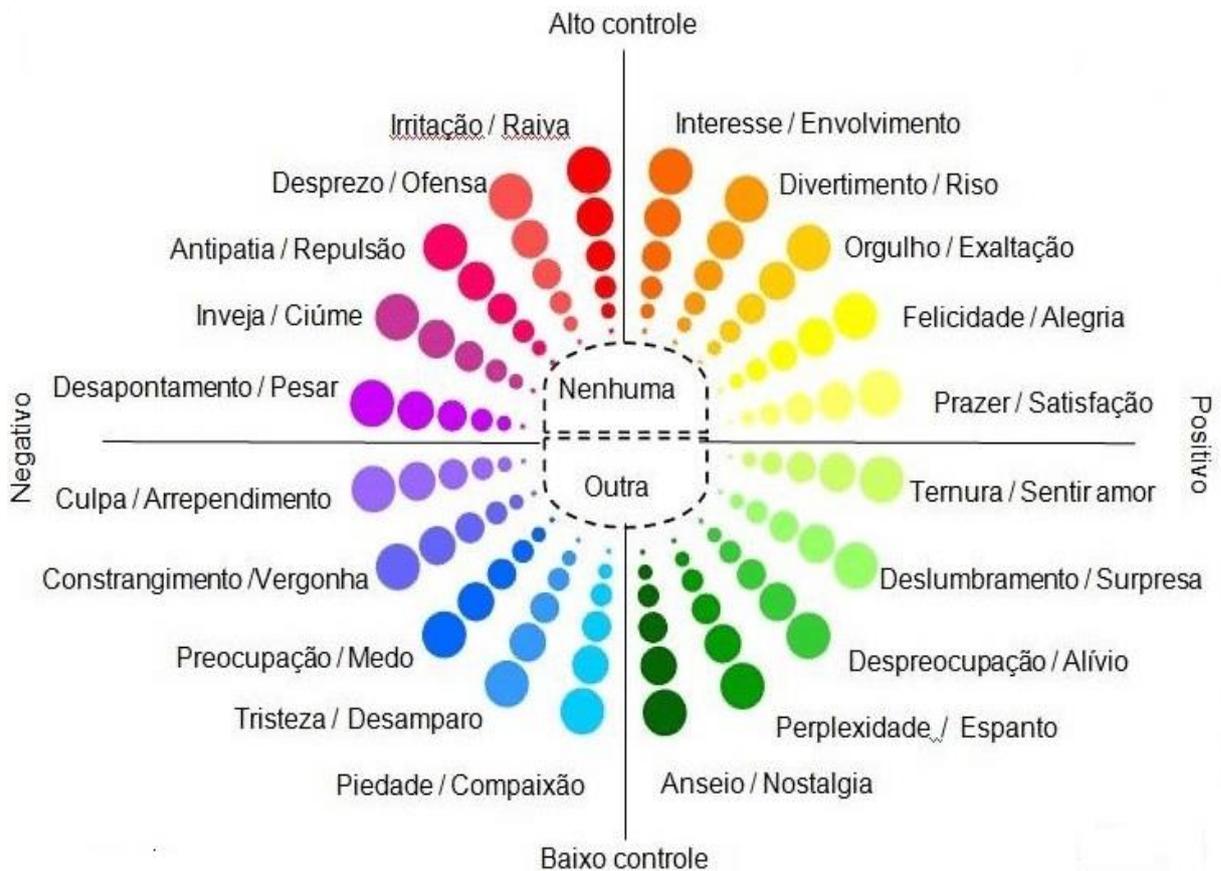
O reconhecimento de estados afetivos é caracterizado como a tarefa de monitorar e sensoriar sinais humanos, tais como, movimentos, ações, expressões faciais, sinais fisiológicos (pressão, pulso), além de dados relacionados ao comportamento de uma ou mais pessoas. Segundo Bercht (2001) o processo pode ocorrer de baixo para cima (*top down*) a partir da captura de sinais de baixo nível e conversão destes em símbolos ou padrões.

Estes sinais ou dados capturados podem ser denominados observáveis e podem se originar de diferentes fontes e estratégias de coleta. Como exemplo é possível citar dados como

os tempos para resolução da atividade, o detalhamento das soluções, a busca por recursos que auxiliem a elaboração da solução para o problema, que servem de base para os atributos para determinar a motivação nesta pesquisa.

Além da coleta de dados em sistemas computacionais, outra técnica para obter informações sobre estados e fenômenos afetivos é o uso de instrumentos para autorrelato dos participantes. Os trabalhos de Scherer e do GERG (*Geneva Emotion Research Group*) da Universidade de Genebra, deram origem a um instrumento para autorrelato, denominado *Geneva Emotion Wheel* (GEW), ou Roda dos Estados Afetivos (REA). A GEW tem como propósito medir reações emocionais relacionadas a eventos, objetos ou determinadas situações (TRAN, 2004; SACHARIN, SCHLEGEL, SCHERER, 2012).

Figura 6 – Roda dos Estados Afetivos



Fonte: Adaptado de Sacharin, Schlegel e Scherer et al. (2012)

Em sua versão gráfica, mostrada na figura 6, os termos relacionados a diferentes famílias de emoções são alinhados sistematicamente em um círculo, que apresenta na parte no eixo o potencial de controle ou de superação (alto ou baixo) e no eixo horizontal a valência (positiva/negativa ou prazerosa/desagradável). Os tamanhos dos círculos e a sua proximidade com as extremidades indica maior ou menor intensidade, quanto maior o círculo e mais próximo

da extremidade, maior a intensidade da emoção e maior a valência ou o controle. Busca-se capturar as emoções e a intensidades destas, durante a participação de um indivíduo em uma atividade ou enquanto este vivencia certas experiências, a partir de um autorrelato feito pela própria pessoa (SACHARIN, SCHLEGEL, SCHERER, 2012).

O questionário baseado na versão 2 da REA foi adaptado e aplicado nos trabalhos de Longhi (2011) e posteriormente em Santos (2016), este último com o objetivo de identificar estados de ânimo dos participantes da pesquisa. O questionário aplicado por Santos (2016) foi adaptado na presente tese com o objetivo de comparar o autorrelato dos estudantes com a classificação adotada no modelo computacional desenvolvido. O principal objetivo foi verificar se estudantes que apresentaram níveis mais baixos de esforço e confiança apresentavam maiores valores associados a aspectos afetivos negativos e menores valores associados aos positivos.

Uma vez que a motivação foi o estado afetivo que serviu de base para este trabalho a seção 2.3.1 dedica-se a definir como a motivação é compreendida no contexto deste trabalho. São exploradas também os aspectos relacionados à motivação intrínseca e extrínseca e como estas influenciam os agentes envolvidos em um contexto educacional.

2.3.1 Motivação

Esta seção descreve os fundamentos e modelos relacionados ao estado afetivo da motivação, em uma abordagem associada à resolução de problemas. Na literatura é possível encontrar diferentes conceituações para a motivação. De acordo com Parellada e Rufini (2013, p. 744):

A motivação humana é um constructo interno, complexo e multideterminado. Diversas teorias elaboradas nos últimos 40 anos colocaram-se diante do desafio de compreendê-la, explicá-la e propor estratégias para intervenção, visando seu incremento.

Keller (1987) estabelece uma reflexão sobre a motivação em um contexto educacional, que, segundo o autor na maioria dos casos é associada com o controle e as regras definidas na sala de aula, o reforço na aprendizagem e nos aspectos afetivos decorrentes do processo instrucional. Na visão do autor, a motivação para compreensão e resolução de problemas deve focar em quatro elementos fundamentais: **atenção**, que é pré-requisito para o aprendizado, a **relevância**, aspecto essencial para manter o foco e atenção do aluno no problema, a **confiança**, que consiste na capacidade do estudante em acreditar que é capaz e a **satisfação**.

Outra perspectiva é apresentada no trabalho de Canto (2015), no qual são descritos os princípios que associam a motivação com a autodeterminação e com níveis de intensidade, que

permitem obter indicativos crescentes de motivação derivados de um controle externo ou interno (Figura 7). A teoria foi proposta por Ryan e Deci (2000) e avalia a motivação sob o ponto de vista da regulação, origem e relevância do processo regulatório.

Figura 7 - Níveis de motivação e autodeterminação

Ambiente Autodeterminação		Motivação extrínseca				Motivação intrínseca
Motivação	Sem motivação					
Estilo regulatório	Não regulado	Externo	Introjetado	Identificado	Integrado	Regulação intrínseca
Percepção da origem	Impessoal	Externa	Relativamente externa	Relativamente interna	Interna	Interna
Relevância do processo regulatório	Não intencional Sem valor Incompetência Sem controle	Submissão, recompensas e punições externas	Controle próprio, ego. Recompensas e punições internas.	Importância pessoal, valorização consciente.	Congruência, consciência, identificação consigo.	Interesse, apreciação, satisfação inerente.

Fonte: Adaptado de (CANTO, 2015; RYAN, DECI, 2000)

Na ótica dos autores, a motivação intrínseca é uma situação na qual o interesse é interno e a regulação é dada pela própria pessoa, sem uma necessidade maior de interferência externa. A motivação extrínseca por outro lado pode ser incremental, iniciando por um estilo regulatório externo, composto por recompensas, punições, passando por estágios em que cresce a capacidade do indivíduo de se autorregular, de perceber a importância e relevância dos eventos, objetos ou atividades que estão sendo realizadas (CANTO, 2015; RYAN, DECI, 2000).

Em um contexto educacional, estudantes que se apresentam sem motivação não percebem qualquer importância ou não exibem interesse no processo de ensino. A motivação intrínseca ocorre quando o objeto de estudo é percebido como algo que lhe dá satisfação pessoal, neste caso não há uma necessidade de regulação externa. Os demais níveis ocorrem em situações em que há uma influência externa, seja pelo objeto de estudo, pelo professor ou qualquer outro fator que impacte de forma positiva ou negativa. Segundo os autores, punições, recompensas, desafios ou a valorização podem influenciar na motivação extrínseca (RYAN e DECI, 2000).

É possível estabelecer uma relação entre a motivação e a complexidade das tarefas que serão desenvolvidas pelos estudantes. Em uma abordagem definida como superficial, estudantes podem visualizar a tarefa simplesmente como um obstáculo que deve ser vencido e,

desta forma utilizam estratégias básicas como a simples memorização. De maneira oposta, ao aumentar a complexidade ou os requisitos de uma atividade é utilizada uma abordagem profunda de aprendizagem o que pode levar ao aumento ou redução da motivação, de acordo com o perfil do estudante e da percepção deste em relação à atividade (CANTO, 2015).

Segundo De Vicente e Pain (1998), a detecção do estado motivacional do aluno é importante, por exemplo para a implementação de ambientes de apoio ao ensino, de Sistemas Tutores Inteligentes (STI). Alguns exemplos de modelos propostos para identificar a motivação são o ARCS (Attention, Relevance, Confidence and Satisfaction) proposto por Keller (1987) e o MSLQ (Motivated Strategies for Learning Questionnaire) definido por Pintrich et al., (1991). Além desses, tem-se o modelo Esforço (E), Independência (I) e Confiança (C) (ECI), o qual foi escolhido como base para este trabalho (BERCHT, 2001; DEL SOLDATO e DU BOULAY, 1995, 2016).

No modelo ECI, a motivação é definida de forma pragmática, em uma combinação do esforço empreendido para o aprendizado, da confiança que o aluno demonstra em sua habilidade de aprender ou resolver um problema em um contexto específico com um mínimo de auxílio o que caracteriza um comportamento mais independente (DEL SOLDATO e DU BOULAY, 1995). Na visão dos autores, o esforço pode ser medido a partir da dedicação e perseverança do estudante, e de uma análise de como ele enfrenta dificuldades e obstáculos. A confiança e a independência podem ser obtidas pela observação da quantidade de auxílio demandada durante a atividade. Alunos desmotivados na maioria das situações não se esforçam ou não acreditam que podem ter sucesso na resolução das tarefas (DU BOULAY e DEL SOLDATO, 2016).

Para Serres (1993) apud Bercht (2001), a motivação está relacionada ao esforço, à independência e confiança demonstradas pelo aluno. O esforço é um indicativo de motivação intrínseca e a sua utilização na maioria dos casos é mais indicada do que outros aspectos, como o desempenho, por exemplo (DEL SOLDATO e DU BOULAY, 1995). Uma revisão do modelo foi descrita pelos autores em (DU BOULAY e DEL SOLDATO, 2016).

Considerando uma perspectiva pedagógica, é possível utilizar sistemas computacionais que capturem informações sobre o comportamento do aluno, tais como o número de erros cometidos em uma atividade, o tempo de dedicado à tarefa, ou ainda o número de tentativas para resolver o problema (LONGHI, BEHAR e BERCHT, 2010; Iepsen, 2013). As estratégias de coleta de dados podem combinar indicadores oriundos de coletas automáticas com informações observadas e registradas manualmente por um professor ou pesquisador.

Entre as pesquisas que visam identificar os aspectos essenciais para identificar a motivação na resolução de problemas, é possível citar o trabalho desenvolvido por Dundunmaya, Banawan e Rodrigo (2018). Entre as variáveis usadas no estudo para medir a persistência, destacam-se as relacionadas aos tempos associados a tutoria, explicações sobre o problema, leitura de dicas ou ainda quantidade de dicas requisitadas. O trabalho citado será descrito com detalhes na seção 2.5, pois a partir dele foi possível estabelecer uma relação entre os tempos e quantidades de acessos à persistência e por consequência ao esforço do estudante.

No contexto da investigação descrita nesta tese, a motivação foi definida como uma combinação dos fatores de esforço, confiança e independência, seguindo a definição de Del soldato e Du boulay (1995, 2016). São adotados ainda os princípios da motivação extrínseca e da autodeterminação, que é importante para que os estudantes sejam capazes de resolver os problemas propostos.(RYAN e DECI, 2000; CANTO, 2015).

A utilização do modelo ECI neste trabalho se deveu a simplicidade do mesmo, a possibilidade de trabalhar com diferentes fatores, como esforço e confiança e principalmente pela ideia de associar a motivação a um conjunto de dados coletados durante a realização das atividades. Outro aspecto importante é que os indicadores descritos por Del Soldato e Du Boulay (1995) são claros e não dependem de uma autoavaliação do estudante, pois podem ser coletados durante a interação com um sistema computacional de apoio ao ensino.

O ECI (Esforço, Confiança e Independência) foi escolhido também por permitir uma análise da motivação com base em três fatores que podem ser considerados essenciais no aprendizado de programação. O esforço para compreender um problema e elaborar uma solução influi diretamente no desempenho do estudante. Além disso problemas relacionados a confiança e independência do aluno podem fazer com que este não se sinta seguro para realizar uma tarefa e desta forma impactar na motivação como um todo. Cabe ressaltar que dentre os fatores que compõe o modelo, apenas o esforço e a confiança foram representados no modelo, uma vez que as variáveis que determinam a confiança podem indicar indiretamente também a independência.

2.4 Técnicas computacionais de apoio

Nesta seção serão apresentadas as técnicas computacionais e algoritmos que servirão de apoio na elaboração do modelo computacional para o reconhecimento da motivação dos estudantes, durante a resolução das atividades. As referidas técnicas estão relacionadas a uma área que tem despertado crescente interesse em pesquisas na área da educação, à mineração de

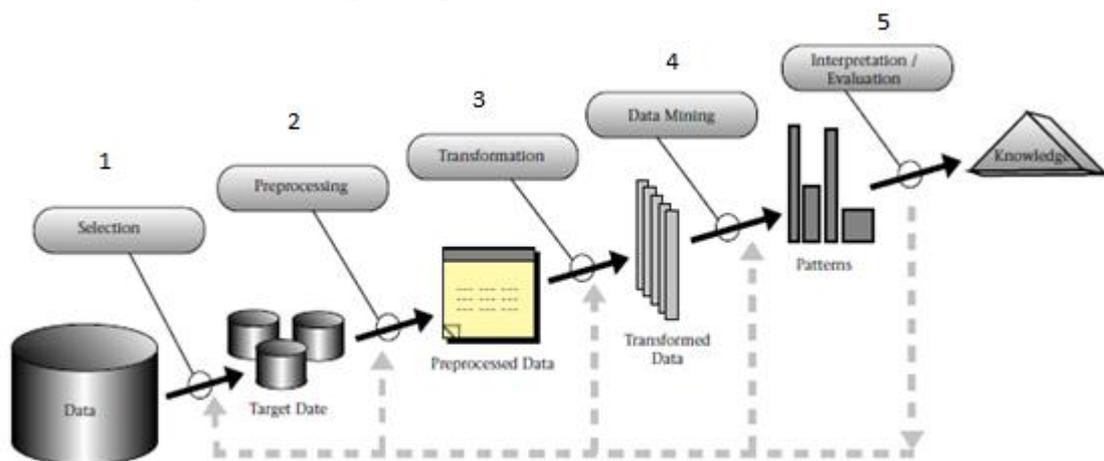
dados educacionais. Inicialmente serão apresentados os fundamentos da descoberta de conhecimento e a seguir serão detalhados os conceitos e técnicas utilizadas.

2.4.1 Descoberta de conhecimento em bases de dados

Com o aumento no volume de dados armazenados e as possibilidades cada vez maiores para a coleta destes dados, tecnologias que tem como objetivo extrair conhecimento e padrões a partir destes dados tem gerado um interesse cada vez maior. O processo denominado KDD (*Knowledge Discovery in Database*) congrega uma série de passos e etapas que visam obter conhecimento analisando um conjunto de dados. A Mineração de Dados ou *Data Mining* pode ser considerada parte deste processo. Esta seção apresenta os conceitos básicos sobre a mineração de dados e os princípios fundamentais da sua aplicação na área educacional.

Descoberta de conhecimento em bases de dados é o processo de identificar padrões que sejam válidos, novos e, potencialmente úteis e compreensíveis, com o objetivo de compreender um problema ou contribuir com a tomada de decisão em diferentes contextos (FAYADD, PIATETSKY-SHAPIRO e SMYTH, 1996). O processo de KDD é interativo e iterativo, envolve diversas etapas e decisões feitas pelo usuário. Para se ter uma visão do processo, destaca-se as etapas na figura 8.

Figura 8 – Etapas do processo de descoberta de conhecimento



Fonte: (FAYADD, PIATETSKY-SHAPIRO e SMYTH, 1996)

Existem diversas atividades que ocorrem antes da aplicação da mineração dados, propriamente dita. Um processo cuidadoso de coleta de dados ou à seleção dos registros (1) adequados a partir dos dados disponíveis é a etapa inicial do processo. Torna-se importante selecionar um conjunto de dados, pertencentes a um domínio, para que, a partir de um critério definido pelo especialista do domínio, possa ser analisado.

O pré-processamento (2) na maioria dos casos é composto de atividades que tem como objetivo preparar os dados para a etapa de mineração. A limpeza dos dados tem o intuito de melhorar a completude, acuracidade e integridade dos dados coletados. Na transformação (3), os dados podem se tornar numéricos ou categóricos, dependendo do algoritmo que será utilizado. Codificar os dados de forma numérica ou categórica é, muitas vezes essencial para que os algoritmos da etapa seguinte possam ser aplicados (DA SILVA, PERES e BOSCARIOLI, 2017). Um exemplo de categorização nesta pesquisa se deu com o tempo de duração de uma tarefa executada por um estudante, que pode foi categorizado como alto, médio ou baixo.

A mineração de dados (4) é a etapa na qual ocorre a descoberta de novas relações, com a utilização de métodos de Inteligência Artificial, através de uma análise sistemática e exaustiva sobre os registros. Segundo Fayyad, Piatetsky-Shapiro e Smyth (1996) as ações consistem na aplicação de algoritmos específicos, que extraem padrões ou relacionamentos entre variáveis ou atributos. As tendências e relacionamentos identificados, em sua maioria encontram-se ocultos nos registros e a sua detecção por seres humanos é uma tarefa complexa. Vários autores (HAN e KAMBER, 2011; RUSSEL e NORVIG, 2013) propõe uma classificação para os métodos e tarefas típicas executadas nesta fase:

- Classificação: busca-se encontrar um modelo ou função que associe conjuntos de registros a rótulos categóricos pré-definidos, ou classes. Após o processo de aprendizado, o modelo pode ser aplicado a novos registros de forma a prever a classe em que tais registros se enquadram.
- Predição: modelos usados para prever o valor de um atributo, que foi perdido ou é desconhecido, a partir dos dados dos demais atributos.
- Clusterização ou análise de agrupamento: também chamada de classificação não supervisionada. É considerada uma abordagem descritiva, enquanto a classificação supervisionada é definida como preditiva (HAN e KAMBER, 2011). Os agrupamentos na maioria dos casos são baseados na análise da similaridade ou em modelos probabilísticos. A análise de *cluster* tem como objetivo verificar a existência de grupos em um conjunto de dados, e em caso de sua existência, determinar quais as suas características (FUHR, 2015).
- Análise de afinidades ou associação: possibilita a identificação de características comuns entre atributos. Permite reconhecer padrões da ocorrência de determinados eventos nos dados em análise, visando determinar possíveis fatos que ocorrem

simultaneamente com uma grande probabilidade (co-ocorrência) ou quais itens de uma massa de dados estão presentes juntos com uma certa chance (correlação).

Sob a ótica do processo de aprendizado, observa-se duas abordagens distintas, a supervisionada e não supervisionada (HAN e KAMBER, 2011). Métodos supervisionados requerem a definição prévia de classes, um atributo que defina a qual classe pertence cada entrada. Neste caso, o algoritmo aprende a classificar de acordo com a classe indicada e os novos registros são classificados conforme o aprendizado. Técnicas não supervisionadas não possuem uma classe pré-determinada e a definição de relações ou associações entre as entradas é feita a partir dos dados, valores dos atributos submetidos ao processo de mineração.

Em muitos contextos, são utilizados métodos não supervisionados com o intuito de explorar um conjunto de dados, analisando possíveis relações entre as entradas, para, posteriormente obter e validar um modelo para classificação supervisionada. Algoritmos baseados em agrupamento ou regras de associação podem ser usados na fase exploratória que busca possíveis relações entre os atributos. Posteriormente as técnicas supervisionadas são aplicadas para comprovar a viabilidade de reconhecer de forma automática a classe ou as relações entre os atributos.

Considerando que este trabalho propõe a utilização de mineração de dados em um contexto educacional, a seguir serão descritos os fundamentos e conceitos relacionados à área de pesquisa denominada Mineração de Dados Educacionais. As técnicas de mineração de dados serão utilizadas na elaboração do modelo afetivo do estudante, com o objetivo de estabelecer relações entre as variáveis que indicam o comportamento do estudante. Técnicas baseadas em classificação permitirão a definição de um modelo para reconhecimento automático dos níveis de esforço, confiança e independência. No capítulo 5 serão detalhadas as técnicas aplicadas nesta pesquisa e a importância de cada uma delas para o reconhecimento da motivação.

2.4.2 Mineração de dados educacionais

A mineração de dados educacionais ou *Educational Data Mining* (EDM) é definida por Romero e Ventura (2010) como um campo de pesquisa emergente e multidisciplinar que se utiliza de técnicas de mineração de dados para explorar dados que se originam de um contexto educacional. O objetivo principal é analisar estes dados para apoiar educadores no processo de ensino e aprendizagem, além de tentar resolver problemas tradicionais na área da educação.

Compreender os estudantes, a forma como aprendem, os fatores que influenciam a aprendizagem, os impactos de uma determinada abordagem instrucional, estão entre as aplicações ou contextos em que a EDM é utilizada. Além destes, observa-se que em outras

situações pode-se utilizar as técnicas para inferir se o aluno está desmotivado, confuso, o que permite adequar o ambiente ou os métodos empregados (BAKER, ISOTANI e CARVALHO, 2011).

A mineração de dados pode ser utilizada nos seguintes contextos educacionais (ROMERO e VENTURA, 2010):

- Educação presencial: onde ocorre a transmissão de conteúdo ou desenvolvimento de habilidades, de forma presencial, face a face. Técnicas podem ser aplicadas para analisar aspectos comportamentais ou a performance dos estudantes, entre outros fatores.
- Educação a distância: neste modelo pode-se utilizar dados de logs e bases de dados coletados durante as ações dos estudantes no ambiente. Análise da colaboração, da comunicação pode contribuir para melhorar o processo.
- Sistemas de tutores inteligentes ou sistemas de ensino adaptativos: técnicas podem ser utilizadas para adaptar o sistema, para moldar o comportamento do tutor, de acordo com o contexto do estudante. A partir dos dados da interação, é possível, por exemplo, sugerir outros exercícios, gerar alertas, entre outras ações.

A utilização de abordagens híbridas, que combinam ensino presencial, com recursos e ferramentas online, além de sistemas de apoio que reconheçam o contexto do estudante é uma justificativa para o uso da mineração de dados no contexto educacional. Educadores podem se utilizar das suas observações e percepções, apoiadas em ferramentas de coleta e análise de dados para rever as estratégias de ensino adotadas, bem como, reconhecer estudantes que apresentam dificuldades de aprendizado.

Outra área de pesquisa considerada emergente é a denominada *Learning Analytics* (LA) ou Analíticas de Aprendizagem, em português. Assim como na mineração de dados educacionais, na LA as atividades estão associadas a coleta e análise de dados relacionados ao contexto dos estudantes, tendo como objetivo principal melhorar o processo de ensino e aprendizagem (MOISSA, GASPARINI e KEMCZINKSKI, 2015).

Analisando as duas áreas percebe-se que em ambas o processo é composto por uma série de etapas, que iniciam pela coleta, seguida do pré-processamento dos dados, passando pela análise com o apoio de algoritmos tradicionais de mineração de dados, culminando com um pós-processamento. Segundo Chatti (2012), LA consiste em um processo cíclico, no qual as análises são seguidas de intervenções com a participação ativa de pessoas, especialmente

professores e alunos, enquanto EDM apresenta um processo sequencial, que se encerra em determinado momento.

Levantamento realizado por Moissa, Gasparini e Kemczinski (2015) discute e corrobora com a afirmação de Baker e Inventado (2014), segundo a qual a mineração de relações e baseada em modelos é mais comum em EDM, enquanto, LA apresenta um número maior de trabalhos associados à análise de textos e ao processamento de linguagem natural. O trabalho descrito nesta pesquisa se insere no campo de pesquisa da EDM, uma vez que a mineração de dados é usada em determinado período de tempo para desenvolvimento de um modelo computacional. Além disso as técnicas empregadas neste trabalho são mais comuns em EDM.

Uma categorização das técnicas mais comuns em EDM é apresentada em (BAKER, ISOTANI e CARVALHO, 2011). Dentre as categorias destacam-se a predição, agrupamento e a mineração de relações. Na predição, os métodos mais comuns estão relacionados à tarefa de classificação supervisionada, enquanto na mineração de relações, pode-se citar a mineração de regras de associação.

Na mineração de regras de associação, a meta é encontrar regras do tipo Se-Então que indiquem casos nos quais um determinado conjunto de valores de atributos está associado a ocorrência de um ou mais valores de outro atributo (BAKER e INVENTADO, 2014). Seria possível, por exemplo, obter padrões de comportamento associados à uma melhor performance em uma disciplina, o que contribuiria para assessorar alunos com dificuldades de aprendizado. Outra possibilidade de aplicação é associar dados coletados ao estado afetivo do estudante, como observa-se no trabalho de Iepsen (2013), onde o engajamento foi relacionado ao número de compilações e erros durante a elaboração de programas.

No presente trabalho técnicas para mineração de regras foram usadas com o objetivo de encontrar relações entre os atributos utilizados no reconhecimento da motivação. Estas relações foram consideradas na construção das redes, especialmente na arquitetura das RB criadas. O presente trabalho segue ainda a abordagem apresentada em Silva et al. (2016), que aplicou mineração de dados em dados de atividades, contrapondo o que os autores definem como visão holística, a qual permite apenas análise em nível de disciplina ou de curso.

No trabalho os autores (SILVA et al, 2016) demonstram que a orientação por atividade contribui para acompanhar a interação constante dos estudantes e para monitorar desenvolver ações pontuais durante o desenvolvimento das tarefas. Entre os objetivos do estudo estão a

definição de modelos para acompanhar e prever o desempenho dos estudantes nas diferentes atividades em um curso de pós-graduação.

2.4.2.1 Aprendizado probabilístico

O estudo e cálculo de probabilidades teve origem em tempos remotos e foi tema de diversos trabalhos desenvolvidos por pensadores como Blaise Pascal (1623-1662), especialmente em problemas relacionados à apostas e jogos de azar. A definição, por exemplo, da probabilidade de uma bola branca ou preta ser sorteada a partir do número total de bolas brancas ou pretas era algo relativamente simples e de fácil determinação.

Ao longo dos anos o raciocínio inverso passou a ser estudado, como por exemplo, o que é possível inferir sobre as bolas na urna quando se conhece as que foram sorteadas? Neste caso busca-se inferir qual a probabilidade de um determinado evento ou sintoma, dado que se conhece o resultado ou a consequência. Um dos teoremas de probabilidade mais importantes foi formulado por Thomas Bayes (1702-1761). O matemático propôs uma regra para calcular a probabilidade de determinados eventos ocorrerem dada a relação com outros eventos ou condições. A equação geral que representa o teorema de Bayes é (RUSSEL, NORVIG, 2013):

$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

Neste caso, $P(A)$ e $P(B)$ são as probabilidades a priori de B e A ocorrerem, $P(A|B)$ é a probabilidade de A, condicionada a B, ou seja, a chance de A ocorrer, dado que B ocorre. Analogamente, $P(B|A)$ é a probabilidade de B ocorrer, considerando que A ocorreu. O principal objetivo do teorema proposto é possibilitar a determinação das probabilidades em situações que se conhece um dos eventos ou condições e a probabilidade destes ocorrerem (RUSSEL, NORVIG, 2013).

Considerando um conjunto de variáveis ou eventos, é possível calcular a distribuição conjunta ou total usando o produto das probabilidades condicionais. A equação para determinar esta distribuição é:

$$P(B) = \sum_{i=1}^n P(B / A_i) P(A_i)$$

A equação permite determinar a probabilidade de uma variável (B) a partir da multiplicação das probabilidades condicionais associadas às demais variáveis ($A_1, A_2..A_n$). Para isso é necessário determinar a relação entre as variáveis, ou seja, identificar quais os atributos que influenciam a probabilidade B.

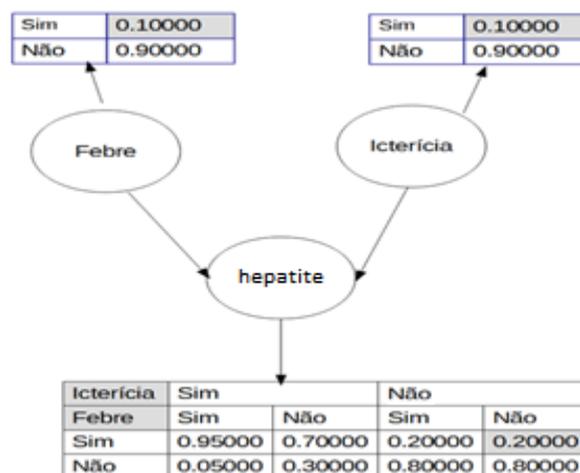
2.4.2.2 Redes Bayesianas

As Redes Bayesianas (RB) são uma das estratégias mais usadas para representação do raciocínio probabilístico, para a representação de conhecimento e informações probabilísticas permeadas pela incerteza (FLORES e BARONE, 2003). Representada por uma rede semântica, com um grafo orientado acíclico (DAG), no qual os nós representam probabilidades associadas a variáveis do problema, estas podendo ser discretas ou contínuas.

Sob a ótica matemática, uma RB é uma representação da conjunção das probabilidades relacionadas a um determinado problema. A partir das fórmulas mostradas anteriormente e do desenho da rede, é possível calcular a probabilidade de um evento ou atributo a partir do conhecimento de probabilidades a priori e outras informações adicionais, como as probabilidades condicionais (LONGHI, 2011).

Computacionalmente pode ser considerada uma estrutura de dados que representa dependências entre variáveis, além de uma distribuição de probabilidades associadas a elas. O grafo possui diversos nós, que representam as variáveis do problema e armazenam informações sobre as probabilidades (LONGHI, 2011). As arestas que ligam os nós, representam uma precedência, ou influência causal entre os nodos (variáveis do problema). As precedências dos nodos pais em relação aos filhos determina uma distribuição de uma probabilidade (P), gerando uma quantificação da influência dos pais em cada um dos filhos (LONGHI, 2011). Quando um nodo não possui nenhuma aresta chegando nele é considerado condicionalmente independente, ou seja, não existem influências de outros atributos sobre ele.

Figura 9 – Exemplo de Rede Bayesiana para diagnóstico de hepatite



Fonte: Adaptado de Flores e Barone (2003)

A figura 9 mostra um exemplo de RB com três nodos (Febre, Icterícia e Hepatite). O nodo da hepatite é condicionalmente dependente de febre e icterícia, enquanto estas são

independentes. A tabela de probabilidades condicionais (TPC ou CPT) é um componente fundamental em uma RB, pois indica as probabilidades associadas a todas as combinações dos valores dos pais, com os valores possíveis para a variável filha.

No exemplo, as variáveis febre e icterícia estão associadas às tabelas de estados que indicam as probabilidades destas a priori, enquanto a variável hepatite está associada à uma tabela (CPT) que mostra a distribuição das probabilidades de acordo com os pais (febre e icterícia). É possível observar que a chance de uma pessoa ter hepatite nos casos em que ela tem febre (=Sim) e icterícia (=Sim) é de 95% e apenas 5% de não ter a doença. Já no caso de ter febre e não icterícia a chance cai para 20%. A rede contribui para a percepção de que o atributo icterícia é o que mais impacta na probabilidade de hepatite. (FLORES e BARONE, 2003).

Outro aspecto importante é a densidade da rede, que é determinada pelo número de nodos que pode influenciar cada nodo. O tamanho da tabela de probabilidades está relacionado ao número de valores dos atributos e o número de nós que impactam em um nó específico. No exemplo da figura 9, o cálculo seria $2*2*2=8$, considerando que todos os atributos possuem 2 valores possíveis. A construção de redes mais complexas com um grande número de valores ou dependências pode dificultar o inviabilizar o aprendizado ou a especificação direta das probabilidades.

Redes bayesianas podem ser elaboradas de duas maneiras distintas. A primeira é a construção e a determinação das probabilidades, de acordo com o conhecimento de um especialista. Esta estratégia é denominada de modelo clássico que envolve um especialista humano. O processo de aquisição de conhecimento, é a etapa que consiste em obter do especialista o conhecimento necessário para representar os componentes qualitativo e quantitativo da rede. A obtenção das probabilidades permite expressar o componente quantitativo, enquanto a definição das relações entre as variáveis se constitui na representação qualitativa da rede (JÚNIOR, WILGES e NASSAR, 2014).

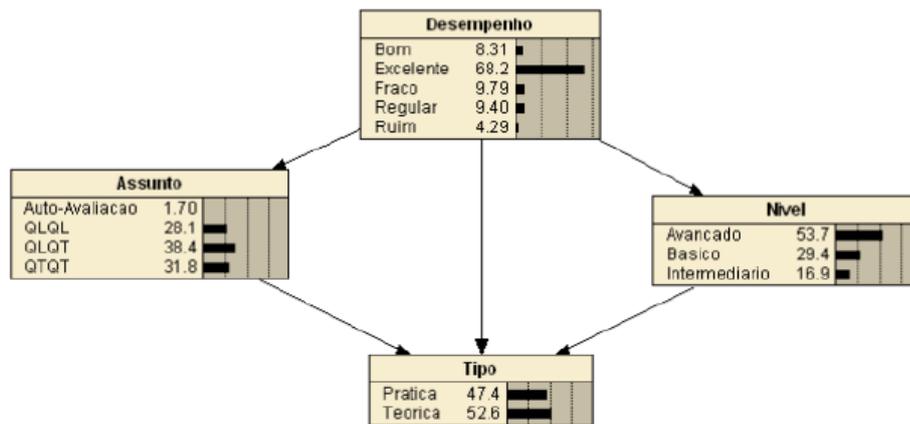
Em muitas situações este processo manual pode ser demorado, custoso e complexo. Podem existir dados incompletos ou desconhecidos, o que pode dificultar muito a construção da rede. Em outras situações o especialista acredita que exista uma alta ou baixa probabilidade associada à determinadas condições, mas não consegue quantificar as mesmas (JÚNIOR, WILGES e NASSAR, 2014). Os aspectos qualitativos, na maioria das situações são mais conhecidos ou facilmente expressados, pois a relação de influência entre determinados atributos é perceptível, embora seja difícil de quantificar.

Outro modelo para elaboração de uma RB é a aprendizagem usando um conjunto de dados sobre os quais são aplicados algoritmos de *Machine Learning*. No processo pode ocorrer tanto o aprendizado da estrutura da rede, quanto dos parâmetros ou probabilidades associadas às entradas. A aprendizagem da estrutura da rede identifica as dependências entre os atributos e a direção da causalidade. Se a estrutura da rede for pré-definida, a principal atividade no treinamento é obter a tabela de probabilidades, usadas na classificação dos casos de entrada (Vier, Glutz e Jacques, 2015).

As etapas gerais para o desenvolvimento de uma RB são:

- Selecionar o conjunto de atributos ou variáveis que sejam suficientes para descrever o problema.
- Transformar cada variável em um nodo, adicionando este na rede.
- Definir os valores possíveis para cada nodo (variável)
- Determinar (ou aprender) as associações entre os nodos, os pais de cada atributo dependente.
- Determinar (ou aprender) as tabelas de probabilidades associadas a cada nodo dependente.

Figura 10 – Rede Bayesiana para prever desempenho de estudantes



Fonte: (JÚNIOR, WILGES e NASSAR, 2014)

Pesquisas relacionando Redes Bayesianas e o reconhecimento de emoções ou estados afetivos na área da educação podem ser encontrados, sendo um dos mais relevantes, o trabalho desenvolvido por Magali Longhi (LONGHI, 2011; LONGHI, BEHAR e BERCHT, 2010). Os principais objetivos foram a identificação dos estados de ânimo dos estudantes, dotando um AVEA de recursos para identificação dos mesmos. Fatores de comportamento e personalidade foram utilizados pela rede para mapear o estado afetivo exibido pelo estudante. A aplicação de

Redes Bayesianas no aprendizado de programação pode ser observada no trabalho desenvolvido por Vier, Glutz e Jaques (2015).

A figura 10 mostra uma rede que visa prever o desempenho de estudantes (JÚNIOR, WILGES e NASSAR, 2014). É possível visualizar uma variável que é a classe (desempenho) e os demais atributos, dependentes, que indicam nível, assunto e tipo de atividade. Segundo os autores, a representação permitiu identificar falhas na elaboração das questões e nas avaliações propostas.

2.4.2.3 Classificador *Naïve Bayes*

O *Naïve Bayes* pode ser considerado como um classificador bayesiano simplificado que utiliza também os fundamentos bayesianos, mas que, diferentemente das RB desconsidera as relações entre os atributos. O termo *naïve* (ingênuo) deriva do fato de que o método acredita que todos os atributos são condicionalmente independentes, o que em muitos contextos não ocorre (CHAKRABARTY, 2002).

Embora utilize uma premissa simplista e ingênua, é um dos classificadores mais utilizados, tendo obtido um excelente desempenho em diferentes tarefas de classificação. Além disso é computacionalmente barato por não exigir um grande poder de processamento, uma vez que mantém somente informações numéricas e contagens de frequências. O processo de treinamento consiste basicamente em determinar as probabilidades de cada valor em relação à uma determinada classe e no cálculo de qual a combinação ou distribuição de valores que possui o maior valor.

Tabela 2 - Exemplo do método *Naïve Bayes*

Faltas	Atividades não entregues	Desistente
Muitas	Muitas	S
Muitas	Poucas	S
Poucas	Muitas	S
Muitas	Muitas	S
Muitas	Poucas	N
Muitas	Muitas	N
Poucas	Poucas	N
Poucas	Poucas	N
Poucas	Muitas	N
Muitas	Poucas	N
Poucas	Poucas	N

Fonte: Elaborado pelo autor

A tabela 2, apresenta um conjunto de dados relacionados a um exemplo hipotético, que permite inferir se aluno irá desistir em uma disciplina, enquanto na tabela 3 é possível visualizar a distribuição dos valores de cada atributo em relação aos valores (S/N) do atributo

que indica a desistência. Tomando por base os casos em que desistente é igual a N, a variável que representa as faltas, apresenta 2 ocorrências para muitas, 5 para com o valor que indica poucas faltas. Já nos casos em que ocorreu desistência, há uma predominância da categoria que indica muitas faltas (3/4).

Tabela 3 – Distribuição dos valores

Faltas			Atividades não entregues			Desistente	
Desistente	S	N	Desistente	S	N	S	N
Muitas	3	3	Muitas	3	2	4	7
Poucas	1	4	Poucas	1	5		

Fonte: Elaborado pelo autor

As probabilidades são usadas para determinar a classe de novas entradas, como no caso de um registro de um estudante que está com muitas faltas e muitas atividades não entregues. A probabilidade de cada uma das classes para a desistência seria dada pelas fórmulas:

- Sim = $3/4 \times 3/4 = 0,562$.
- Não = $2/7 \times 3/7 = 0,122$.

A normalização dos valores obtidos pode ser obtida pela divisão do valor obtido na ocorrência pela soma das ocorrências, usando as seguintes fórmulas:

- Sim = $0,562/(0,562+0,122) = 82,12\%$.
- Não = $0,122/(0,562+0,122) = 17,88\%$

Conclui-se neste caso que existe maior probabilidade (82,12%) de um estudante desistir, portanto, a entrada é classificada na classe sim, ou seja, o classificador indica que neste caso o aluno irá desistir. A saída do *Naive Bayes* é simplesmente a classe, não existindo uma definição sobre a probabilidade maior ou menor de ocorrer uma determinada classe.

2.4.2.4 Regras de associação

Em muitos contextos, não é possível efetuar uma classificação supervisionada pela falta de atributos de classe ou pelo fato de que o objetivo é descobrir relações entre atributos. Neste caso é necessário empregar métodos exploratórios, usados para descrever os dados, para encontrar informações úteis que, devido ao grande volume dos dados passam despercebidas na análise humana.

Os métodos baseados na análise de associação se constituem em ferramentas importantes e úteis para descobrir relacionamentos em grandes conjuntos de dados (TAN, STEINBACH e KUMAR, 2009). Os relacionamentos são representados na forma de regras de associação, que representam itens do tipo $A \rightarrow B$ (se A então B) que aparecem com frequência

nos dados de entrada. Neste exemplo, A é definido como o antecedente, ou condição e B é chamado de consequente da regra.

Um dos exemplos mais citados na literatura é conhecido como “transações de cestas de compras”, na qual são registrados os produtos que cada cliente levou em uma compra (TAN, STEINBACH e KUMAR, 2009). A descoberta de que clientes que compram fraldas também compram cerveja ilustra um conhecimento oculto e inesperado que foi percebido somente com a aplicação de métodos para extração de regras.

Um dos algoritmos mais utilizados para extração de regras é o Apriori. O método descrito por Tan, Steinbach e Kumar (2009), utiliza a seguinte premissa: se um conjunto de itens é frequente, então todos os seus subconjuntos também deverão ser frequentes. Por exemplo, se $\{a, b, c\}$ é um conjunto de itens que aparecem com frequência nos dados, transações que contém $\{a, b, c\}$ devem conter seus subconjuntos $\{a, b\}$, $\{a, c\}$, $\{b, c\}$. O algoritmo inicia buscando os itens que aparecem com frequência em determinada fração das transações e após gerar um conjunto de itens candidatos, percorre as transações para mediar a frequência de cada um.

A qualidade ou peso de uma regra pode ser medida ou avaliada usando duas métricas principais: o suporte e a confiança. O suporte indica a frequência na qual uma regra é aplicada em um determinado conjunto de dados, ou seja, o percentual de registros que possuem os itens A e B combinados. A confiança permite inferir a frequência na qual os itens B aparecem nas transações que contém A, em outras palavras nas transações que possuem A, o percentual que também possui B (SILVA, CRUZ e SILVA, 2014).

2.4.2.5 Agrupamento

Assim como ocorre na análise de associação, os métodos de agrupamento são considerados exploratórios e descritivos, ou seja, são utilizados para descrever um conjunto de dados, encontrar relações entre os mesmos. Tan, Steinbach, Kumar (2009) definem a análise de grupos como uma tarefa que divide o conjunto de entradas em grupos, capturando a estrutura ou organização dos dados.

Na maioria dos casos o agrupamento permite compreender os dados, obter classes ou grupos que possuem características comuns. Em muitos casos o agrupamento é usado para obter valores de atributos que caracterizem classes, neste caso, os valores representam uma ou mais entradas que representam conjuntos de registros (TAN, STEINBACH e KUMAR, 2009).

Um dos algoritmos para agrupamento mais conhecido é o *K-means*, que é baseado no conceito de centroide (TAN, STEINBACH e KUMAR, 2009). O centroide pode ser definido

como a representação de um ponto no espaço que caracteriza um grupo de dados. O método inicia pela definição de um conjunto (K) de centroides iniciais, sendo K o número de grupos escolhido pelo usuário. Cada entrada é associada a um centroide mais próximo, a partir de um cálculo de proximidade, normalmente usando uma métrica como a distância Euclidiana (L_2). Ao final do processo espera-se que os centroides representem os diferentes grupos e a similaridade existente entre os dados de entrada.

2.4.3 Avaliação de classificadores

Ao aplicar um método de classificação supervisionada é necessário avaliar o modelo obtido para verificar se apresenta um desempenho aceitável e se o mesmo pode ser utilizado para classificar novas entradas, não utilizadas no treinamento. Os resultados do processo de treinamento na maioria dos casos incluem uma matriz de confusão, semelhante à mostrada na tabela 4.

Tabela 4 – Matriz de confusão

Classes	Classe verdadeira P	Classe verdadeira N
Classe prevista P	TP	FP
Classe prevista N	FN	TN

Fonte: Adaptado de Witten e Frank (2005)

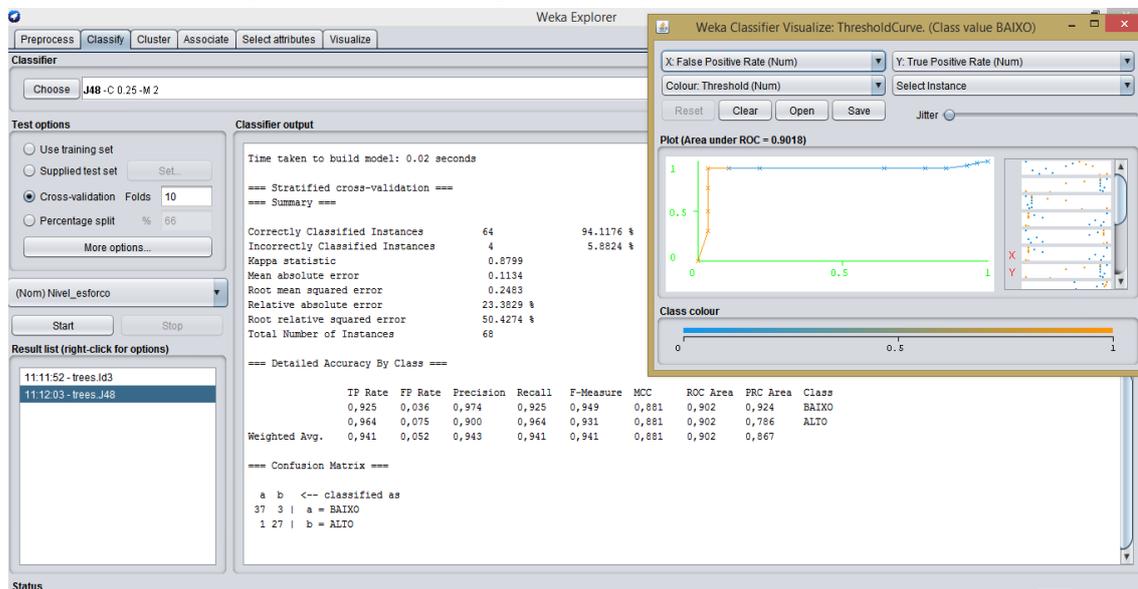
A classe prevista é o rótulo atribuído à cada entrada nos dados de treinamento, por exemplo, se uma entrada foi classificada com o valor positivo, espera-se que o algoritmo após o treinamento seja capaz de classificar a mesma desta forma. Os verdadeiros positivos (TP) são valores que foram definidos como positivos nas entradas e classificados desta forma pelo algoritmo. Os falsos negativos (FN) são os positivos classificados pelo algoritmo como negativos. Falsos positivos (FP) são valores negativos, que o método classificou como positivos e, por fim, os verdadeiros negativos (TN) são os negativos classificados corretamente como negativos. (WITTEN e FRANK, 2005).

A partir da matriz de confusão é possível definir diversas métricas para análise dos resultados e do desempenho do classificador.

- Revocação (*Recall*) ou TPR (*True Positive Rate*): descreve a porção que foi classificada corretamente como exemplos positivos, $TP/(TP+FN)$. É definida por alguns autores pelo termo “sensibilidade”. É o caso de prever como positivos, os que realmente são positivos.

- *Specificity (SPEC)* ou TNR (*True Negative Rate*): porção que foi classificada corretamente como exemplo negativo. Capacidade do sistema de identificar claramente os que são negativos (N), por exemplo.
- *Precisão (Precision)*: proporção de exemplos positivos que foram classificados corretamente: $TP/(TP+FP)$, considerando o total que foi classificado pelo método como positivo.
- *Medida-F (F-Measure)*: as medidas precisão e revocação podem ser enganosas quando examinadas separadamente. Uma precisão elevada geralmente significa sacrificar a revocação e vice-versa. A medida F indica um possível ponto de equilíbrio, $(2 * (precisão * revocação)) / (precisão + revocação)$.
- *Acurácia (Accuracy)*: porcentagem de amostras positivas e negativas classificadas corretamente sobre a soma de amostras positivas e negativas, ou seja, trata-se da proporção de classificações corretas: $(TP+TN)/(TP+TN+FP+FN)$.
- *ROC (Receiver Operating Characteristic Curve)*: medida e gráfico que demonstra a relação entre TPR e TNR. Usa um espaço contínuo e uma curva. Quanto mais próximo da margem superior esquerda é o ponto de corte, melhor.

Figura 11 – Métricas para avaliação de um classificador



Fonte: Elaborado pelo autor

Após aplicar o método é necessário avaliar as métricas em conjunto para verificar se o modelo é robusto e consistente e se o mesmo pode ser utilizado para classificar novos dados, casos reais. Embora seja uma das medidas mais importantes, a acurácia sozinha não é suficiente. Em alguns casos o método apresenta boa acuracidade, mas apresenta uma curva ROC ou uma

precisão ruim para um dos valores de classe, por exemplo. A figura 11 mostra os resultados das métricas após a aplicação de um classificador.

2.4.4 Ferramentas para mineração de dados

A investigação e o desenvolvimento de ferramentas para aplicação de mineração de dados em diferentes problemas não é uma preocupação recente. É possível encontrar várias suítes, compostas por diferentes algoritmos, suportando os mais variados métodos. Entre estas ferramentas estão os softwares Weka e Netica, que serão descritos a seguir.

2.4.4.1 Weka

O Weka (*Waikato Environment for Knowledge Analysis*) é uma ferramenta desenvolvida na Universidade de Waikato na Nova Zelândia, tendo sido implementada na linguagem Java, sendo um produto gratuito, de código aberto e possuindo diversos algoritmos para a execução de suas tarefas. A mesma possui uma interface gráfica, denominada *Explorer*, que engloba as etapas de pré-processamentos, processamento e pós-processamento. Estão disponíveis diversos métodos de associação, classificação e clusterização. A inclusão ou remoção de novos métodos pode ser realizada de forma simples e rápida, o que torna a ferramenta customizável e expansível. (RUSSEL e MARKOV, 2017).

Figura 12 – Exemplo de arquivo no formato arff

```
@relation dados_esforco_atividade

@attribute Classe_tp_prob { BAIXO,MEDIO,ALTO }
@attribute Classe_tp_hip { BAIXO,MEDIO,ALTO }
@attribute Classe_prop_ph { PROPORCIONAL,DESPROPORCIONAL }
@attribute Classe_cod { BAIXO,ALTO,MEDIO }
@attribute Visualizou_pseudo { S,N }
@attribute Classe_detalhe_prob { MEDIO,ALTO,BAIXO }
@attribute Classe_detalhe_hip { BAIXO,ALTO,MEDIO }
@attribute Nivel_compreensao { MEDIO,ALTO,BAIXO }
@attribute Executou { N,S }
@attribute Nivel_esforco { BAIXO,ALTO }

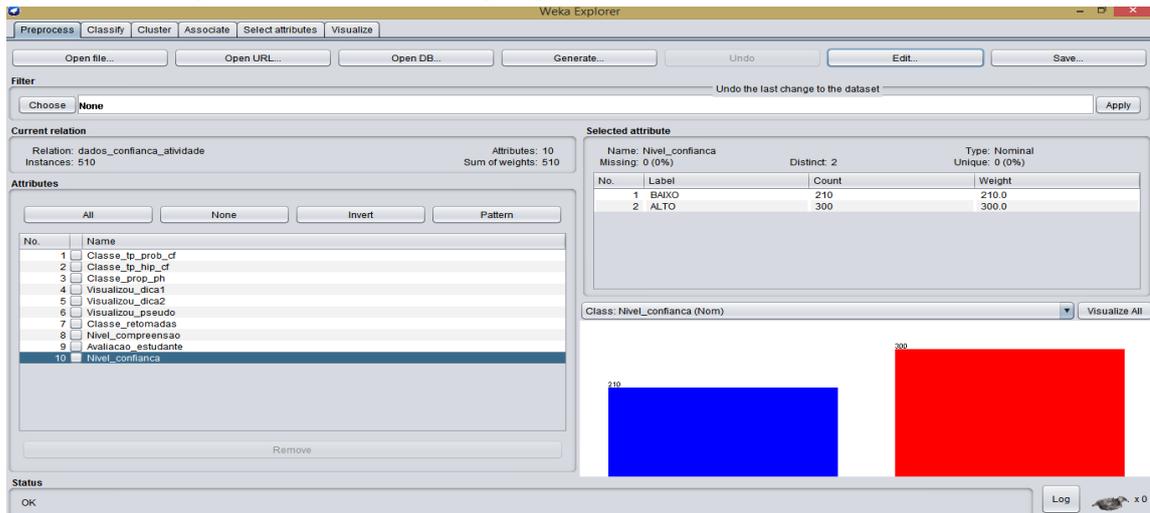
@data
BAIXO,BAIXO,PROPORCIONAL,BAIXO,S,MEDIO,BAIXO,MEDIO,N,BAIXO
MEDIO,MEDIO,PROPORCIONAL,ALTO,N,ALTO,ALTO,ALTO,S,ALTO
```

Fonte: Elaborado pelo autor

O Weka suporta a manipulação de arquivos do tipo *Attribute-Relation File Format* (ARFF), baseados no formato ASCII (*American Standard Code for Information Interchange*), no qual são definidos os atributos e seus valores. Um exemplo arquivo no formato arff é mostrado na figura 12. Inicialmente é definido o nome da relação e na sequência os atributos, com os possíveis valores. Os dados que serão usados no treinamento são relacionados após a tag *@data*.

Os resultados podem ser exibidos sob a forma de histogramas, modelos gráficos de redes neurais, além da estrutura hierárquica de árvores de decisão (RUSSEL e MARKOV, 2017). Versões mais recentes permitem a conexão direta a bancos de dados, ou arquivos em formato XLS, entre outros formatos. Um dos pontos negativos é a limitação do volume de dados que pode ser manipulado, que é limitado à capacidade de memória principal.

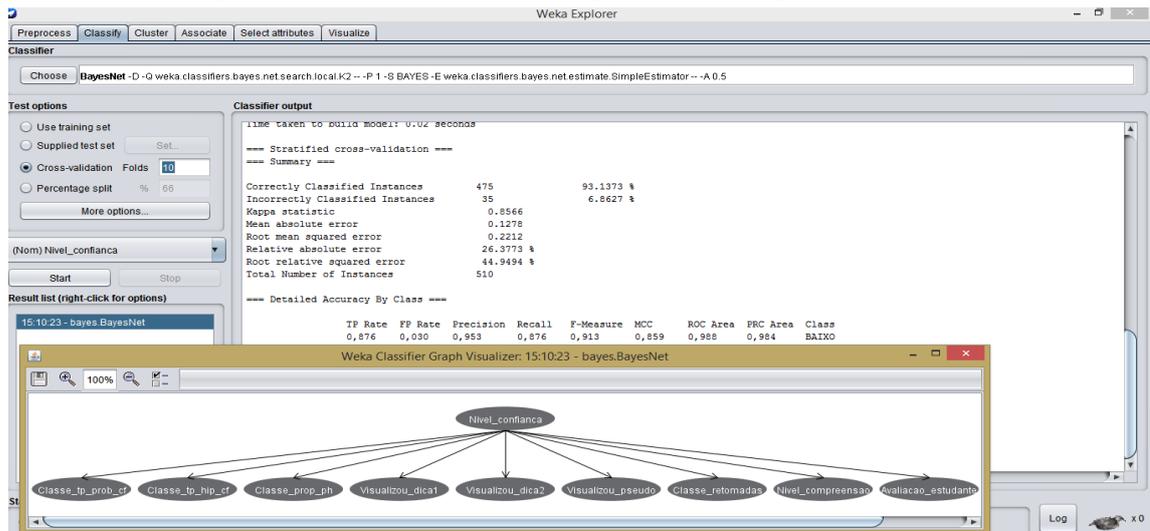
Figura 13 – Interface de pré-processamento da ferramenta Weka



Fonte: Elaborado pelo autor

A tela para o pré-processamento, mostrada na figura 13 permite selecionar um arquivo e analisar a distribuição dos dados de diferentes atributos. Estão disponíveis também diversos métodos para executar um pré-processamento ou uma transformação dos dados.

Figura 14 – Interface do Weka para classificação usando RB



Fonte: Elaborado pelo autor

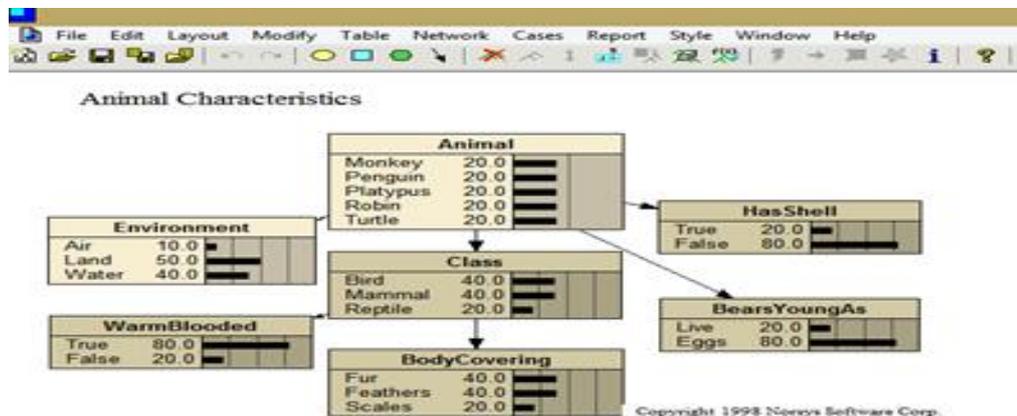
A figura 14 ilustra a tela para aplicação de um algoritmo de classificação baseado em Rede Bayesiana, no qual é possível observar a representação gráfica da rede gerada no processo de mineração. Além do método exibido, o software disponibiliza uma variedade de algoritmos

que podem ser aplicados, não apenas para classificação supervisionada, mas também para agrupamento, extração de regras e para seleção de atributos.

2.4.4.2 Netica

O Netica é um software desenvolvido pela empresa Norsys (<https://www.norsys.com/netica.html>) para a construção de Redes Bayesianas que apresenta uma interface gráfica, de simples utilização. Além dos recursos para desenhar uma rede, a ferramenta permite ainda a edição direta das probabilidades e valores de cada atributo. A figura 15 mostra a interface da ferramenta e um exemplo de rede disponível na biblioteca de exemplos disponibilizada pelo aplicativo.

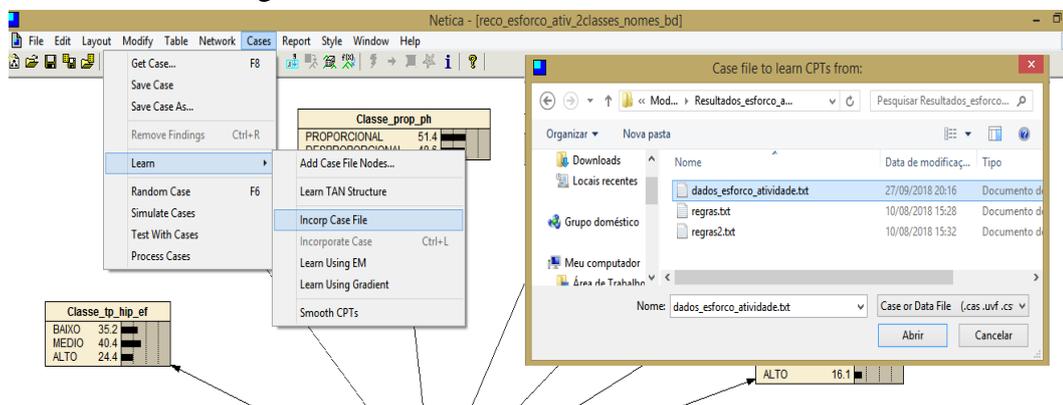
Figura 15 – Interface do software Netica



Fonte: Norsys: Capturado de <https://www.norsys.com/netica.html>

Após construir a rede, é possível executar um processo de treinamento com o objetivo de determinar a tabela de probabilidades associadas às variáveis. Para tanto deve ser utilizado um arquivo de texto com as respectivas entradas. A figura 16 exhibe a alternativa que permite abrir um arquivo e a partir deste realizar o treinamento da rede.

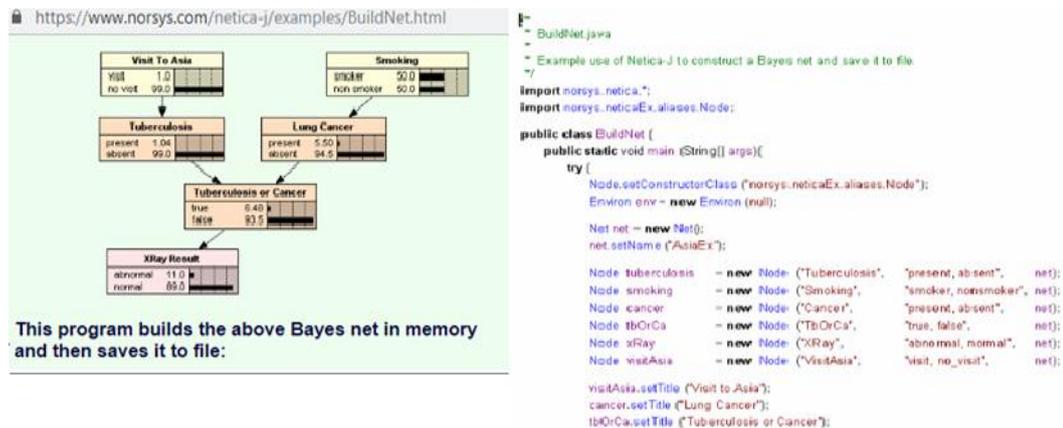
Figura 16 – Treinamento de uma rede no Netica



Fonte: Elaborado pelo autor

Uma das facilidades oferecidas pela ferramenta é a possibilidade de simular uma entrada definindo os valores e percentuais de determinadas variáveis para obter as probabilidades das demais. Além disso, estão disponíveis diversas APIs (*Application Programming Interface*) em diferentes linguagens de programação, o que contribui para facilitar a integração ou inserção da rede em aplicações reais, tanto em plataformas *desktop*, quanto *web*. A figura 17 exibe um exemplo de API para a linguagem Java.

Figura 17 – Exemplo de uso da API do Netica



Fonte: Norsys: Capturado de <https://www.norsys.com/netica.html>

Na pesquisa descrita nesta tese as duas ferramentas foram utilizadas, o Weka em uma fase inicial, para aplicação de métodos de agrupamento e regras de associação e o Netica para a construção e treinamento das Redes Bayesianas. O software Weka contribuiu também para avaliar o método bayesiano como um classificador, uma vez que a ferramenta apresenta um resumo das métricas para um classificador, quando este é aplicado.

2.5 Trabalhos relacionados

As pesquisas e levantamentos bibliográficos realizados até o presente momento permitiram reunir e descrever um conjunto de trabalhos e pesquisas que serviram de base para a elaboração desta proposta. Serão descritas iniciativas relacionadas à problematização, ao reconhecimento de estados afetivos, além da aplicação de técnicas baseadas na mineração de dados educacionais.

2.5.1 Mapeamento de estados afetivos em um AVA

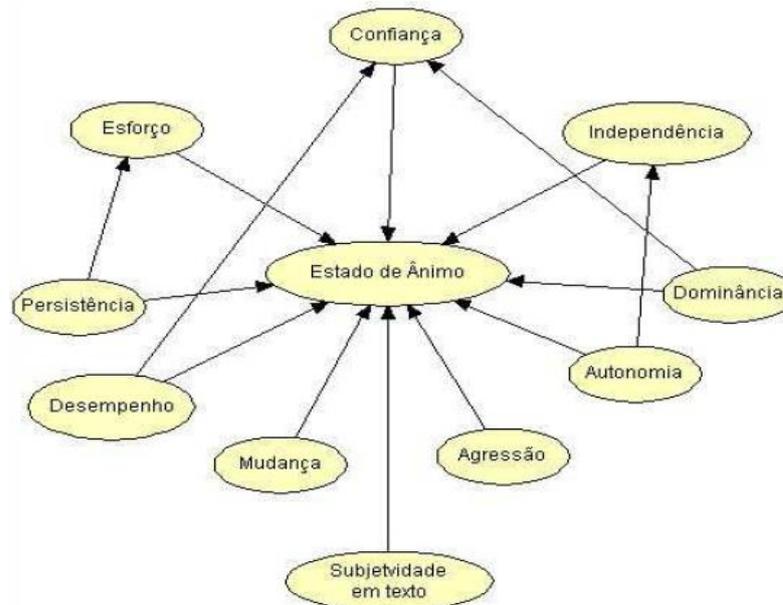
Reconhecer o estado de ânimo dos estudantes em Ambientes Virtuais de Aprendizagem (AVA), é o tema principal da tese de doutorado desenvolvida por Longhi (2011). Uma das questões essenciais que motivaram o trabalho foi a possibilidade de acompanhar e monitorar o estado de ânimo em disciplinas ministradas na modalidade EAD. Além de investigar os fatores que poderiam indicar o estado de ânimo dos estudantes, outro objetivo importante foi a

aplicação de técnicas computacionais que permitiram reconhecer o estado de ânimo do aluno, além de incorporar estes mecanismos como um recurso pedagógico em um AVA.

Com relação aos modelos afetivos, a principal referência adotada foi o modelo de Scherer (2005), combinado o modelo ECI (Esforço, Confiança e Independência) para a motivação. Um questionário baseado na Roda dos Estados Afetivos (REA) foi utilizado para obter informações sobre o estado afetivo do aluno, no início, durante e no final das atividades realizadas. O modelo ECI (DE BOULAY, 1995) foi aplicado para tratar padrões comportamentais, inferindo o grau de motivação do estudante. Foram definidas variáveis relacionadas ao comportamento observável, o número de testes, de reinícios, pedidos de ajuda, tempo de permanência em cada sessão e o modo de finalização são exemplos de variáveis.

O raciocínio probabilístico, baseado em Redes Bayesianas foi utilizado como estratégia computacional para a identificação dos estados de ânimo (Figura 18). Em uma segunda etapa foi desenvolvido um recurso chamado ROODAAfeto que teve como objetivo, mapear os estados afetivos dos estudantes no AVA ROODA. Para a implementação do mesmo foram propostos modelos de estados de ânimo, animado e desanimado. Os resultados apresentados confirmaram a hipótese de que é possível reconhecer os estados de ânimo dos alunos a partir de fatores relacionados à personalidade e comportamento (LONGHI, 2011).

Figura 18 – Rede Bayesiana completa



Fonte: (LONGHI, 2011)

Esta tese baseia-se em diversos aspectos do trabalho desenvolvido por Longhi (2011). O conjunto de observáveis que compõe o modelo para reconhecimento da motivação inclui variáveis relacionadas aos tempos, pedidos de ajuda, reinícios, aspectos que também serão

utilizados no presente trabalho. O modelo para reconhecimento da motivação desenvolvido e apresentado nesta tese se baseia em Redes Bayesianas de forma análoga ao trabalho de Longhi (2011).

É possível constatar entretanto, que no presente trabalho foram implementadas várias redes cada uma associada ao um fator (confiança e esforço) e contexto (atividade e estudante). A possibilidade de reconhecer a motivação em cada atividade, para cada estudantes representa uma inovação em relação ao trabalho descrito por Longhi (2011).

2.5.2 Descoberta do desânimo em AVEA

A pesquisa desenvolvida por Santos (2016) consistiu na identificação do desânimo manifestado por estudantes em um Ambiente Virtual de Ensino e Aprendizagem (AVEA), em disciplinas ministradas na modalidade EAD. Segundo a autora, reconhecer estados afetivos negativos é essencial para evitar um desempenho abaixo do esperado e também que o aluno desista do curso. O acompanhamento dos fatores associados ao comportamento do estudante em um AVEA permite que o professor execute ações que podem estimular ou fazer com que o estudante retome as suas atividades, participando de forma ativa do processo de ensino e aprendizagem.

Foram propostas e utilizadas variáveis relacionadas ao comportamento exibido pelo aluno na interação em um AVEA. A partir dos indicadores coletados foi proposto um modelo para identificação do desânimo, e este posteriormente foi incorporado ao ambiente de apoio. O questionário baseado na REA foi adaptado de Sachrin, Schlegel e Scherer (2012) e empregado como ferramenta para obter um autorrelato do estudante. Análises estatísticas indicaram que os sentimentos mais comuns eram constrangimento, vergonha e culpa, especialmente nas situações em que não cumpriam os prazos ou apresentavam dificuldades para realizar determinada atividade.

Dados contidos nos logs do AVEA Moodle foram utilizados na definição de variáveis que representassem o comportamento observável do estudante. Foram propostos quatro componentes para agrupamento das variáveis, tarefas, arquivos e fóruns. Técnicas computacionais baseadas na mineração de dados educacionais foram usadas para extrair regras que pudessem indicar desânimo a partir dos dados coletados nos arquivos de log.

Foram constatadas relações entre o fato da tarefa ter sido entregue com atraso, a frequência dos atrasos, o conceito do estudante e o desânimo indicado no autorrelato. Além da extração de regras, o algoritmo J48 foi aplicado também para a classificação dos dados e para a geração de árvores de decisão. Entre os resultados destaca-se a incorporação do modelo à

plataforma Moodle, em um *dashboard*, permitindo a identificação dos alunos que apresentam comportamento mais propenso ao desânimo (Santos, 2016).

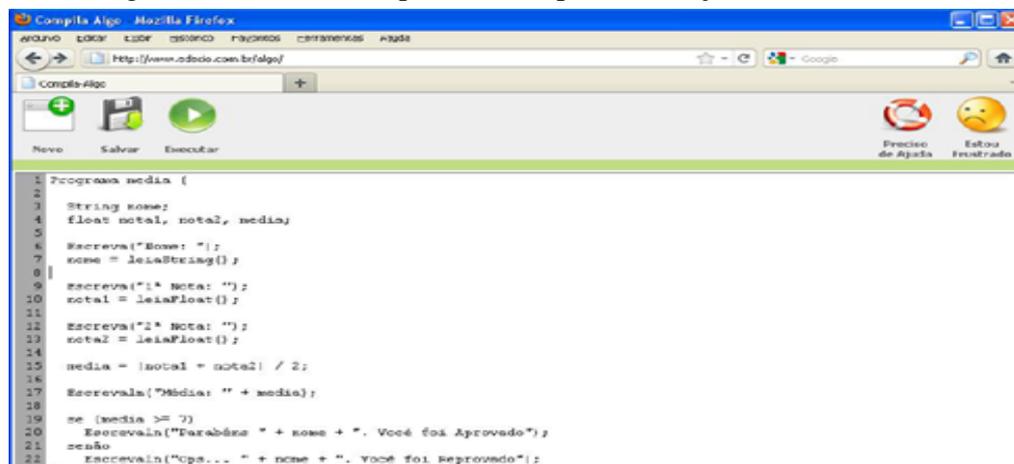
O trabalho de Santos (2016) também possui diversos aspectos que serviram de base para a elaboração desta proposta, desde a proposição de um modelo baseado em um conjunto de variáveis associadas ao comportamento do estudante, quanto o uso de técnicas computacionais baseadas na mineração de dados educacionais. Regras de associação também foram utilizadas na presente pesquisa, embora o seu uso tenha se restringido à uma fase exploratória, na construção do modelo afetivo.

Constata-se que as principais diferenças do trabalho descrito e da presente proposta residem também na investigação da motivação associada à resolução de problemas de programação e na aplicação de um método baseado na problematização, o que representa um ineditismo em relação aos trabalhos de Santos (2016) e Longhi (2011). Outra questão a ser ressaltada é que nos dois trabalhos apresentados foram utilizados ambientes de ensino já existentes e foram criadas funcionalidades complementares nestes, enquanto na pesquisa aqui descrita foi implementada uma ferramenta completa que permitiu a aplicação do método de ensino, além da validação do modelo para reconhecimento da motivação.

2.5.3 Identificação da frustração no ensino de algoritmos

Um dos trabalhos que serve também como referência para esta pesquisa foi desenvolvido por Iepsen (2013). Nele são utilizadas técnicas de mineração de dados baseadas em regras para encontrar padrões de comportamento associados à frustração no aprendizado de algoritmos. O principal objetivo de Iepsen (2013) foi identificar sinais de frustração demonstrados pelo aluno durante a resolução de exercícios de programação, além de propor e executar ações para minimizar a frustração e apoiar o aprendizado dos estudantes.

Figura 19 – Sistema implementado para resolução das atividades



```

1 Programa media {
2
3 String nome;
4 float nota1, nota2, media;
5
6 Escreva("Nome: ");
7 nome = leiaString();
8
9 Escreva("1ª Nota: ");
10 nota1 = leiaFloat();
11
12 Escreva("2ª Nota: ");
13 nota2 = leiaFloat();
14
15 media = (nota1 + nota2) / 2;
16
17 Escreva("Média: " + media);
18
19 se (media >= 7)
20     Escreva("Parabéns " + nome + ". Você foi Aprovado");
21 senão
22     Escreva("Ops... " + nome + ". Você foi Reprovado");

```

Fonte: (IEPSEN, 2013)

No trabalho citado foi desenvolvido ainda um sistema utilizado pelos alunos para escrever e compilar o algoritmo e enquanto estes resolvem a atividade, têm a possibilidade de indicar se está frustrado clicando em um botão disponível no software (Figura 19).

Figura 20 – Regras geradas

```

IF-THEN RULES:
7)  duração is Alta
    and número_compilações_com_erros is Alta
    if and only if
    frustrado is Sim
    The rule exists in 4 records.
    Significance Level: Error probability < 0,001

10)  If número_compilações_com_erros is Alta
    Then
    frustrado is Sim
    Rule's probability: 0,800
    The rule exists in 4 records.
    Significance Level: Error probability < 0,01
    Deviations (records' serial numbers):
    11

12)  If número_compilações_sem_erros is Alta
    Then
    frustrado is Sim
    Rule's probability: 1,000
    The rule exists in 3 records.
    Significance Level: Error probability < 0,1

17)  If duração is Alta
    and programas_seguidos_com_erros is Alta
    Then
    frustrado is Sim
    Rule's probability: 1,000
    The rule exists in 3 records.
    Significance Level: Error probability < 0,1

18)  If programas_seguidos_com_erros is Alta
    and número_compilações_com_erros is Alta
    Then
    frustrado is Sim
    Rule's probability: 1,000
    The rule exists in 3 records.
    Significance Level: Error probability < 0,1

28)  If programas_seguidos_com_erros is Alta
    Then
    frustrado is Sim
    Rule's probability: 0,750
    The rule exists in 3 records.
    Significance Level: Error probability < 0,1
    Deviations (records' serial numbers):
    9
  
```

Fonte: (IEPSEN, 2013)

Durante a utilização do software, são coletados dados relativos à número de compilações com erros, além dos tempos para a elaboração do código fonte. A informação que indica se o aluno está frustrado também é gravada caso o aluno opte por clicar no botão “Estou frustrado”. O processo de mineração dados utiliza técnicas para extração de regras, sendo utilizado para tanto o software WizRule. Observa-se que as regras mais significativas relacionam alto número de compilações, a incidência recorrente de erros em programas ou uma maior duração da

atividade com o estado de frustração indicado pelo usuário. Algumas destas regras são mostradas na figura 20.

Tanto o trabalho desenvolvido por Iepsen (2013), quanto o desenvolvido e descrito nesta tese têm como principal preocupação o ensino e aprendizagem de programação. As técnicas de mineração de dados utilizadas diferem, uma vez que o primeiro utilizou métodos para extração de regras, enquanto na presente pesquisa foram adotadas RB. Além disso foi possível complementar o trabalho de Iepsen (2013), com a coleta e análise de dados nas etapas anteriores à escrita do código fonte. A análise do comportamento do estudante durante as etapas de problematização, não apenas na escrita do código contribuiu para uma percepção mais completa sobre o estado de afetivo do estudante.

2.5.4 Problematização aplicada ao ensino de algoritmos

A problematização foi aplicada no ensino de programação no trabalho desenvolvido por De Oliveira Balduino e Ferreira (2015). Foi aplicada a problematização com base no arco de Magueres combinada com tecnologias como SMS e modalidades de ensino semipresencial. A metodologia proposta foi composta de quatro etapas:

- Ponto inicial: no qual o estudante analisa individualmente o problema e identifica os pontos-chave do mesmo.
- Junção em grupos: os problemas são analisados em grupos e os alunos que enviarem primeiro a sua análise, são considerados líderes do grupo.
- Apresentação da solução final: os líderes de todos os grupos se reúnem para escolher e apresentar a solução ótima escolhida a partir das diversas soluções debatidas.

No trabalho, as TIC são usadas como ferramentas que permitem a colaboração e a interação entre os participantes. As principais tecnologias usadas no trabalho foram o Google Docs e recursos móveis baseados em SMS. Como forma de justificar a aplicação da metodologia e o uso das tecnologias citadas, os autores apresentaram levantamentos relativos ao número de estudantes que possuem celular e utilizam com frequência SMS e a internet a partir destes dispositivos.

O referido artigo foi citado como um trabalho relacionado por tratar-se de uma das raras aplicações da problematização ao ensino de programação. Entretanto, a investigação realizada por De Oliveira Balduino e Ferreira (2015) não apresenta uma coleta sistemática de dados e não tem como foco identificar a motivação apresentada pelos estudantes. Também não foi desenvolvido um sistema de apoio para a aplicação da metodologia nas atividades

desenvolvidas, a opção recaiu sobre tecnologias disponíveis como recursos de compartilhamento de documentos e envio de mensagens. O sistema PROALG implementado neste trabalho e o modelo para reconhecimento da motivação associada ao método da problematização se constituem em uma evolução e ampliação do uso da problematização no ensino de programação.

2.5.5 Exploração do caminho percorrido pelo estudante na resolução de problemas de programação

Estudo desenvolvido por Hosseini, Vihavainen e Brusilovsky (2014) apresentou como tema principal a análise do caminho e dos passos do estudante durante a elaboração de soluções para problemas de programação, utilizando a linguagem Java. Segundo os autores, quando o programa é submetido, entregue é possível avaliar a solução final, mas pouco se sabe sobre a forma e estratégia usada pelo estudante durante a construção.

Ao longo dos anos os sistemas de apoio para realização de exercícios de programação e avaliação destes pelos professores tem sido aperfeiçoados constantemente, permitindo mudanças no ensino de programação. Entretanto, um importante impacto destes sistemas ainda é pouco debatido, que é o incremento da capacidade de compreender como os estudantes resolvem os problemas. Na maioria dos casos os sistemas permitem acesso somente à solução final, sem considerar as informações sobre o processo de elaboração da solução. Na visão dos autores, os passos intermediários devem ser avaliados e podem ser essenciais para entender as dificuldades enfrentadas pelos estudantes.

Foi proposta uma forma de capturar *snapshots*, dados que mostram cada passo usado pelo estudante enquanto este escreve o código fonte. Foram feitas comparações sobre o estado do código a cada passo, alterações feitas pelo estudante, observando principalmente a estrutura e complexidade do código. Uma das principais conclusões foi que os estudantes implementam soluções de forma incremental, adicionando novas funcionalidades a cada passo. Outro aspecto a ser salientado é que as alterações entre um *snapshot* e outro foram maiores para estudantes iniciantes, ou seja, estes tendem a mudar mais a solução a cada passo, tanto na remoção quanto na adição das funcionalidades.

Os participantes foram classificados de acordo com o padrão de atividade detectado ao longo de um conjunto de exercícios. Os “construtores”, que se constituem no grupo predominante, resolvem problemas preferencialmente pela adição de novas funcionalidades, enquanto “redutores” iniciam por um grande conjunto de funcionalidades e vão reduzindo aos poucos para chegar a uma solução final.

Assim como no trabalho desenvolvido por Hosseini, Vihavainen e Brusilovsky (2014), a pesquisa aqui apresentada também tem como preocupação entender como os alunos resolvem problemas de programação. Uma semelhança entre as pesquisas é a gravação das ações dos estudantes durante a escrita do código, o que neste trabalho foi automatizada na ferramenta PROALG. Entretanto, o foco principal desta proposta é nas etapas anteriores a escrita do código e na identificação dos fatores associados à motivação dos estudantes com o objetivo de compreender como estes compreendem e especificam o problema antes de elaborar o código e a influência deste processo na codificação.

2.5.6 A motivação e o ensino de programação

A preocupação com a motivação dos estudantes e com o impacto desta no aprendizado de programação têm sido tema de várias pesquisas, ao longo dos anos. Pesquisadores tem tentado compreender quais fatores estão relacionados a motivação e como estes afetam o aprendizado, tanto negativa quanto positivamente. A seguir serão descritos trabalhos que foram considerados de relevância e influenciaram na elaboração desta proposta.

Bergin e Reilly (2005b) apresentam um estudo dos fatores que influenciam a performance dos estudantes em módulos de programação introdutórios. No artigo são apresentados resultados relacionados a dois fatores: motivação e nível de conforto. A motivação é definida pelos autores como uma combinação da vontade, da necessidade, desejo e a disposição do aluno para participar do processo de aprendizagem. Considera-se dois tipos de motivação, a intrínseca, na qual o aluno realiza a tarefa por iniciativa própria, pela sua própria realização e a extrínseca que depende de premiações, punições ou outros tipos de estímulo. A definição de nível de conforto incorpora a facilidade de um aluno para resolver as questões de programação, sua autopercepção e sua autoestima.

A investigação foi realizada com estudantes de um curso que utiliza PBL como metodologia de ensino e aprendizagem, na universidade de Sussex, no Reino Unido. As principais conclusões apontam para o fato de que estudantes que apresentam maiores graus de motivação intrínseca tem performance superior aos que exibem altos graus de motivação extrínseca, que apresentou menor impacto na performance dos programadores. Além disso o grau de importância e utilidade que o estudante atribui à tarefa, as crenças e esforços, quando positivos também incrementam a performance destes (BERGIN, REILLY, 2005). Com relação ao nível de conforto, não foi percebido um impacto significativo na performance de estudantes que apresentam menores ou maiores níveis de auto estima e autoconfiança.

O trabalho desenvolvido por Nigel Bosch e Sidney D’Mello (2017) identificou um conjunto de emoções e estados afetivos demonstrados por estudantes em um experimento de programação. Um conjunto de 99 estudantes participou de atividades que consistiam na resolução de problemas de programação, em intervalos que intercalavam atividades intensivas, durante 25 minutos, com paradas previstas de 10 minutos. Utilizando como base o registro de vídeo feito dos estudantes, enquanto estes realizavam as atividades, buscou-se reconhecer a ocorrência das emoções e estados afetivos e a sequência em que estes ocorreram.

Os estados afetivos que ocorreram com mais frequência foram engajamento, confusão, frustração, tédio e curiosidade. Emoções como ansiedade, felicidade, surpresa, raiva, desgosto ou medo também foram identificados, porém com menor incidência. Transições recíprocas entre engajamento e confusão e entre confusão e frustração foram percebidas durante o experimento. Uma das conclusões é que a escrita do código foi o elemento central, catalisador das transições entre os diferentes estados e emoções.

Estados afetivos de frustração ou confusão foram detectados, geralmente após a ocorrência de erros ou dificuldades para compilar ou executar um programa. A seguir normalmente o estudante buscava auxílio ou acessava um recurso de ajuda. A curiosidade ou o engajamento foram verificados diversas vezes após o início da codificação da solução, entretanto, a confusão foi um estado demonstrado várias vezes, após a leitura e tentativa de interpretação do problema.

A relação entre a persistência e a motivação é investigada na pesquisa desenvolvida por Dundunmaya, Banwan e Rodrigo (2018). Os autores argumentam que estudantes que persistem nas tarefas, mesmo diante das dificuldades encontradas demonstram maior esforço e são mais habilitados para lidar com a frustração. O foco principal dos autores é a análise da persistência demonstrada pelos estudantes na resolução de problemas, mesmo em situações em que os alunos falham nas primeiras tentativas.

Para mediar a persistência são utilizados dados de logs das tarefas realizadas pelos estudantes em um sistema de tutoria inteligente. Foram considerados dados como o tempo gasto na resolução ou tentativa de resolver a atividade, o que segundo os autores é um indicativo de esforço e por consequência, de motivação, além de registros de abandono, retomadas e tentativas (DUNDUNMAYA, BANAWAN E RODRIGO, 2018). Técnicas baseadas em agrupamento foram aplicadas e resultaram em dois grupos distintos de estudantes, os com maior e menor persistência. A conclusão final dos autores é que o grupo de estudantes que apresentou

maior persistência foi mais efetivo na resolução dos problemas, em consequência de uma maior motivação e engajamento nas tarefas.

Os três trabalhos relacionados à motivação, descritos nesta seção, contribuíram para o desenvolvimento desta pesquisa, demonstrando que a motivação ou outros estados afetivos podem afetar significativamente os resultados obtidos nas disciplinas de programação. Uma abordagem que foque na compreensão do problema pode contribuir para que o estudante só inicie a codificação no momento em que já tiver uma visão mais clara de como irá elaborar a solução o que pode reduzir as dificuldades técnicas e aumentar a motivação intrínseca dos estudantes, e de acordo com Bergin (2005) pode melhorar o aprendizado.

A coleta de dados relacionados ao comportamento do estudante durante a aplicação das etapas de problematização, além do questionário para o autorrelato pode permitir a identificação das emoções exibidas pelos estudantes, de forma análoga ao trabalho de Bosch e D’Mello (2017). A observação das transições entre estados afetivos contribuirá também para a elaboração do modelo para reconhecimento da motivação que será proposto e aplicado.

A relação estabelecida entre persistência e motivação e a utilização de dados relacionados aos tempos ou retomadas de tarefas descrita no trabalho de Dundunmaya, Banawan e Rodrigo (2018) foi importante para a definição dos observáveis deste trabalho. Percebe-se no trabalho a relação entre os tempos das atividades e os observáveis definidos por Du Boulay e Del Soldato (2016), como, por exemplo, o número de tentativas de resolver um problema. A presente tese utilizou estes princípios como base para definição das variáveis das RB, especialmente no caso dos tempos para definição dos pontos-chave e hipótese que serão detalhados posteriormente.

2.6 Alicerces para instrumentação da pesquisa

Ao longo deste capítulo foram apresentados os fundamentos e pressupostos teóricos que embasaram o desenvolvimento deste trabalho. O panorama da pesquisa no ensino de programação demonstrou que o número de trabalhos na área tem crescido nos anos recentes. Percebe-se também que boa parte das pesquisas é focada no desenvolvimento de ferramentas ou ambientes de apoio, como juízes online, jogos, entre outros. É possível encontrar diversos trabalhos que focam nas estratégias pedagógicas e também na investigação da motivação dos estudantes. Observou-se que há espaço para iniciativas que combinem metodologias, análise de estados afetivos e um ambiente que suporte o aprendizado.

Foram descritos ainda os conceitos e passos utilizados na metodologia da problematização com o Arco de Magueres. Salienta-se que a problematização foi a base para a

definição da estratégia pedagógica aplicada nesta pesquisa. A definição dos conceitos sobre a motivação foi fundamental também para definir as características do modelo para reconhecimento da motivação, além dos observáveis que embasaram a coleta de dados. Os pressupostos para a motivação, definidos por Del Soldato e Du Boulay (1995) foram utilizados como base para elaboração do modelo afetivo apresentado no capítulo 5.

Os métodos e algoritmos para mineração de dados educacionais se constituem nas técnicas computacionais que foram utilizadas para elaboração do modelo para reconhecimento automático da motivação. As Redes Bayesianas foram descritas com mais detalhes por terem sido as técnicas mais importantes no contexto do trabalho, pois foram utilizadas diretamente na elaboração do modelo. Métodos de agrupamento e regras de associação foram empregados como técnicas exploratórias no início do processo de construção do modelo afetivo.

Nesta seção serão apresentados os alicerces utilizados na construção da pesquisa, inicialmente as seções 2.6.1 e 2.6.2 descrevem o método baseado na problematização que foi fundamental na realização das atividades, na especificação e desenvolvimento do sistema PROALG e na construção do modelo para reconhecimento da motivação. Nas seções seguintes são apresentadas a visão geral e as bases para o modelo afetivo e os princípios para utilização das técnicas de mineração de dados para a construção das Redes Bayesianas.

2.6.1 Método baseado na problematização para o ensino de programação

Um dos princípios fundamentais que norteou a definição do método e o uso da problematização foi a necessidade de estimular o desenvolvimento das habilidades relacionadas à resolução de problemas, que são essenciais para o aprendizado de programação (SOUTO e TEDESCO, 2017; ZORZO et al, 2017; MEC, 2016). A abstração é outra habilidade importante que foi considerada no método aplicado (UTTING et al., 2013; BENNEDSEN e CASPERSEN, 2008).

A problematização foi aplicada através de um conjunto de passos, que incluíram uma definição da situação-problema por parte do professor e uma análise da mesma por parte do estudante. O conjunto de passos baseia-se nos conceitos do Arco de Magueréz apresentados por Bordenave e Pereira (1982), Berbel e Gamboa (2011).

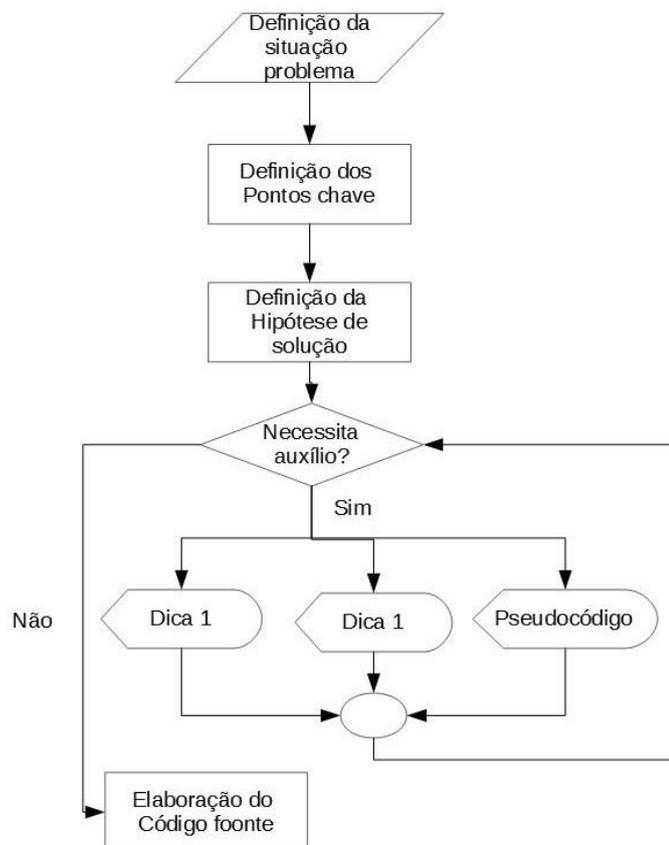
A definição dos pontos-chave consiste em apontar os elementos que serão utilizados na resolução (Figura 21). O foco principal é compreender um problema e identificar os elementos (entradas, saídas, estruturas, construtos, etc) e o conhecimento necessário para elaborar a solução deste. A reflexão sobre o problema e a necessidade de identificar os pontos-chave

contempla as definições da abstração que é considerada uma habilidade de extrair as características essenciais de um problema ou contexto (KRAMER, 2007).

O passo seguinte é a elaboração de uma hipótese para a solução do problema, uma sequência de passos, etapas, ações ou ainda uma especificação em alto nível da solução (algoritmo). A hipótese permite perceber o raciocínio ou a organização mental formulada pelo estudante na compreensão do problema ou a capacidade de elaborar um modelo mental da solução antes de iniciar a escrita do código.

Por fim, é elaborado o código fonte que constitui-se na solução final para o problema. Na definição da hipótese e na construção do código, o estudante pode recorrer a um conjunto de dicas ou mesmo ao pseudocódigo da solução, como apoio para esclarecer dúvidas. O estudante passa a ter acesso as dicas apenas depois de definir os pontos-chave, o que estimula a compreensão e reflexão sobre o problema e seus principais elementos antes de que sejam dados mais detalhes sobre a resolução do mesmo.

Figura 21 - Fluxo para o desenvolvimento das atividades



Fonte: Elaborado pelo autor

A incorporação de recursos de ajuda, dicas, ocorreu para atender a um dos aspectos principais do modelo ECI proposto por Del Soldato e Du Boulay (1995, 2016) e para permitir

a coleta de dados relacionados ao acesso dos estudantes a estes recursos. É possível citar também outras iniciativas que propõe a apresentação de exemplos ao estudantes como forma de auxílio, como o trabalho desenvolvido por Nagano (2016).

O método apresentado nesta seção foi um dos pilares da pesquisa desenvolvida, tendo sido essencial tanto na especificação dos requisitos do sistema PROALG, que será detalhado no capítulo 3, quanto na definição dos atributos usados na elaboração do modelo para o reconhecimento da motivação.

Entretanto, para viabilizar a aplicação do método da problematização uma das principais questões foi a descrição ou contextualização do problema, ou seja, a elaboração de um texto que instigasse o estudante a refletir sobre o problema, além de aproximar o mesmo de uma situação real. Uma descrição que desafie o estudante que faça com que ele tenha que buscar conhecimentos é importante e pode influenciar na motivação extrínseca e autodeterminação do aluno (RYAN e DECI, 2000; CANTO, 2015).

2.6.2 Problematização: contextualização do problema

Nesta seção serão apresentados os princípios que nortearam a contextualização dos problemas, além de exemplos utilizados ao longo da pesquisa. Um dos principais desafios para a aplicação da problematização no ensino de programação diz respeito a observação ou contextualização do problema, algo que é ainda mais complexo para disciplinas introdutórias. Nestes componentes curriculares não é possível utilizar problemas de alta complexidade visto que o principal objetivo em muitas atividades é o ensino dos conceitos básicos, das instruções fundamentais de programação.

Além disso, alunos em início de curso possui experiência limitada e muitas dificuldades para analisar situações reais, o que dificulta a elaboração do problema por parte dos estudantes. Constata-se portanto, que o professor deve ser capaz de trazer o problema para em cada atividade, elaborando descrições ou exemplos que contribuam para o que o estudante perceba em quais situações reais é possível aplicar os conhecimentos de programação (ZANINI, RAABE, 2012).

Diversos estudos na área de programação buscam compreender como as descrições dos problemas e enunciados influenciam na compreensão do estudante e na elaboração da solução. Teusner e Hille (2018) demonstram que a clareza, o nível de detalhamento da situação-problema influenciam no tempo, no número de compilações e nos erros encontrados nas soluções elaboradas.

Segundo Witt, Kemczinski e Dos Santos (2018) a maioria das iniciativas que utiliza a problematização no ensino de computação baseia-se em problemas trazidos pelo professor e em problemas ligados à realidade. Observa-se também que sistemas que permitem a manutenção de atividades como o URI Online (BEZ, TONIN, RODEGHERI, 2014) possuem recursos para uma descrição do problema e diversos exemplos que visam aproximar a atividade do mundo real.

Com o objetivo de estimular a reflexão do estudante, já no estudo piloto foi decidido que seriam expostas duas descrições para o estudante, o problema e um enunciado para elaboração do código. A descrição do problema foi apresentada no início de cada atividade, sendo utilizada para definição dos pontos-chave e a hipótese do problema, enquanto o enunciado passou a ser mostrado apenas no momento em que o estudante deveria elaborar o código fonte.

Para a elaboração da descrição do problema foram estabelecidas algumas premissas básicas. Inicialmente deveria ser criada uma introdução sobre a situação real, sobre o contexto no qual está inserido o problema, incluindo uma definição geral sobre os resultados esperados. A descrição deve sempre que possível instigar a busca por algum conceito sobre o caso apresentado, uma complementação do conhecimento, embora não deva requerer conhecimentos muito complexos em uma determinada área. Foi estabelecido ainda que recursos de programação, como variáveis, matrizes ou mesmo instruções utilizadas não são mencionados na definição do problema, cabendo ao estudante buscar materiais didáticos, leituras ou exemplos que irão subsidiar a construção da solução.

O enunciado deve focar nos elementos utilizados no desenvolvimento da solução, na elaboração do código fonte. Neste caso é importante indicar o que é esperado, entradas, saídas e um maior detalhamento das fórmulas ou mesmo da lógica usada na implementação. A apresentação de descrições diferentes para problema e enunciado teve como objetivo também incrementar a percepção do estudante sobre a passagem de uma especificação mais abstrata para uma visão mais próxima da implementação. A tabela 5 apresenta exemplos de descrições de problemas e enunciados.

Tabela 5 – Exemplos de problemas e enunciados

Descrição do problema	Enunciado para elaboração do código
<p>A final do campeonato regional vai ocorrer no próximo final de semana. Espera-se um grande público. O estádio no qual vai acontecer a final possui dois tipos de acomodações, as arquibancadas e as cadeiras numeradas. As pessoas que compram cadeiras pagam um valor de 80 reais por cada ingresso e quem fica em arquibancada paga 40 reais. O estádio possui um total de 2000 lugares, considerando tanto arquibancada quanto cadeiras. Para facilitar a apuração da renda e da ocupação, crie uma solução para calcular a renda total, a partir do número de pessoas que comprou cadeiras e arquibancadas. Além disso, indique o percentual de ocupação total do estádio.</p>	<p>Faça um programa que receba a quantidade de público presente nas cadeiras e nas arquibancadas de um estádio de futebol, em uma final de campeonato. Calcule e mostre o percentual de ocupação e a renda total do jogo, considerando que o público nas cadeiras paga 80 e nas arquibancadas 40 reais. Neste caso deve-se multiplicar o público presente em cada local pelo valor pago para obter a renda total. No estádio cabem 2000 pessoas, sendo assim, é necessário dividir o público total pela capacidade para saber a ocupação.</p>
<p>Uma empresa possui um conjunto de salas para a realização de treinamentos. Todas as salas têm o mesmo tamanho, mas o número de cadeiras colocadas em cada uma delas é diferente. Um dos objetivos é que exista um padrão, ou seja, que todas as salas têm um número semelhante de cadeiras. O responsável pela administração das salas ficou incumbido de fazer um levantamento sobre o número de cadeiras em cada sala e após calcular o número médio do conjunto de salas. A seguir ele deverá apresentar um relatório indicando quantas salas possuem menos cadeiras que a média e quantas possuem mais cadeiras que a média geral. É importante salientar que existem 30 salas nesta empresa.</p>	<p>Ler e armazenar em um vetor, a quantidade de cadeiras existentes em 30 salas de uma empresa. Totalizar as cadeiras existentes em todas as salas e ao final, calcular a média.</p> <p>Após o cálculo da média, percorrer o vetor comparando a quantidade de cada sala com a média, para contar quantas possuem menos e mais cadeiras. Mostrar ao final as contagens.</p>
<p>A cada ano é necessário elaborar a declaração do imposto de renda. Você sabe como calcular o imposto de renda que é descontado do seu salário? Para calcular o desconto do imposto de renda é necessário conhecer o salário bruto e sobre este aplicar o cálculo do INSS, que pode ser de 8% (até 1317), 9% (entre 1317 e 2905) ou 11% (valores maiores, conforme o valor do salário bruto).</p> <p>Após descontar o valor do INSS, deve-se calcular o valor a ser deduzido de acordo com o número de dependentes. Cada dependente permite um desconto de R\$ 189,00. O percentual do imposto e a parcela de dedução fixa também mudam de acordo com o valor do salário após descontar o INSS e o valor relativo aos dependentes. Para</p>	<p>Elabore um programa que receba o salário bruto e o número de dependentes para um conjunto de pessoas e para cada uma calcule os descontos de INSS e IR, além do salário líquido a receber. É necessário diminuir o salário bruto do inss e do número de dependentes multiplicado por 189 para obter o valor usado para enquadrar na tabela de IR. Com o enquadramento na tabela é possível saber o valor fixo a deduzir e o percentual do IR.</p> <p>Mostrar os dados calculados.</p>

<p>exemplificar, vamos observar a tabela abaixo, na qual estão mostrados os percentuais e valores a deduzir no ano de 2015. Tabela para o cálculo do imposto:</p> <ul style="list-style-type: none"> - De 1.787,78 até 2.679,29 7,5% R\$ 134,08 - De 2.679,30 até 3.572,43 15% R\$ 335,03 - De 3.572,44 até 4.463,81 22,5% R\$ 602,96 - Acima de 4.463 27,5% R\$ 826,15 <p>Para auxiliar as pessoas a calcular o desconto de IR no salário vamos elaborar um programa que receba o salário bruto e o número de dependentes de um conjunto de pessoas. Para cada uma delas, calcular e mostrar o desconto do INSS, do IR e o salário líquido. Se você tiver dúvidas sobre o cálculo faça uma pesquisa para compreender melhor a fórmula usada para calcular este desconto.</p>	<p>As fórmulas de cálculo são:</p> <p>INSS = Valor bruto * percentual desconto</p> <p>IR = ((Valor sem INSS - (dependentes*189)) - Parcela fixa a deduzir)* percentual IR/100</p> <p>Liquido = Valor bruto - INSS - IR</p>
---	---

Fonte: Elaborado pelo autor

Os exemplos mostrados na tabela 5 demonstram as diferenças entre problema e enunciado, sendo o primeiro uma definição mais ampla, em muitos casos, mais extensa, com foco no caso real, nas necessidades de possíveis usuários. O enunciado apresenta informações muitas vezes complementares, sendo menor nos exemplos exibidos.

Tomando por base o terceiro caso, que apresenta um problema de cálculo do imposto de renda, as fórmulas e os dados que devem ser lidos no algoritmo são mostrados apenas no enunciado. O problema apresenta uma contextualização sobre o que caracteriza o imposto de renda e uma tabela geral para o cálculo deste, o que pode demandar uma pesquisa por parte do estudante para compreender melhor como é feito na prática o cálculo. O objetivo não é que o estudante conheça profundamente os detalhes de uma declaração, mas que busque conhecimentos básicos que permitam a resolução do problema.

No segundo exemplo o problema não apresenta nenhum detalhe sobre armazenamento, mas no enunciado fica clara a necessidade de usar um vetor, uma vez que são diversas salas com diferentes capacidades. Ao longo das atividades, os estudantes deveriam ser capazes de perceber os tipos de variáveis ou estruturas a serem usadas e descrever as mesmas nos pontos-chave ou na hipótese de solução.

2.6.3 Em busca do modelo para reconhecimento da motivação

O modelo afetivo para o reconhecimento da motivação é sustentado por três pressupostos: i) o modelo ECI (Esforço, Confiança e Independência) apresentado por Du Boulay e Del Soldato (1995, 2016); ii) os dados relacionados às ações do estudante durante a

realização das atividades, usando o método da problematização e; ii) técnicas para mineração de dados educacionais, com ênfase nas Redes Bayesianas (RB).

Serviram de inspiração também os resultados dos trabalhos desenvolvidos por Longhi (2011), Santos (2016) e DundunMaya, Banawan e Rodrigo (2018). O primeiro apresenta o uso de redes bayesianas, demonstrando como o raciocínio probabilístico pode ser usado para a classificação de estados de ânimo. De forma análoga a presente tese mostra como estas redes se adaptam bem a identificação dos fatores de esforço e confiança. Já o trabalho de Santos (2016) empregou técnicas de classificação supervisionada e diversas variáveis associadas ao comportamento do estudante, tal como na pesquisa descrita nesta tese.

Cabe ressaltar ainda que os atributos relacionados aos tempos para conclusão das tarefas diferem do número de tentativas, que foi usado como indicativo de esforço em Del Soldato e Du Boulay (1995, 2016) e em Bercht (2001). Entretanto estão alinhados com outras iniciativas, como a persistência apresentada por Dundunmaya, Banawan e Rodrigo (2018), que estabelecem uma relação entre o esforço e os tempos das tarefas.

Outra observação relevante diz respeito aos fatores da motivação. O modelo apresentado neste capítulo é baseado nos princípios de Del Soldato e Du Boulay (1995, 2016), mas foram implementadas redes para reconhecimento apenas do esforço e da confiança, sem um reconhecimento explícito da independência.

Durante a seleção dos dados e o pré-processamento foi possível perceber que a maioria dos atributos permitia reconhecer a confiança e indiretamente também a independência, não sendo possível distinguir claramente os dois fatores. Du Boulay e Del Soltado (2016) destacam que tanto a confiança, quanto a independência podem ser indicadas pela ajuda requisitada ou acessada pelo estudante ao longo da tarefa, o que também justifica o uso apenas da confiança como indicativo indireto da independência nesta pesquisa.

Considerando o conjunto de variáveis disponíveis, optou-se, portanto, em não discriminar separadamente a confiança e a independência. Ao analisar os dados, foi possível perceber que o conjunto de dados que poderia ser usado como indicador de confiança, também seria usado para indicar a independência, desta forma, seria mais conveniente indicar somente a confiança, que neste caso também pode ser considerada uma representação da independência demonstrada.

Outro aspecto importante do modelo afetivo é a possibilidade de determinar a motivação por atividade, ao longo de um componente curricular, o que permite avaliar os efeitos das atividades, dos desafios e problemas no esforço e na confiança demonstrados. Este

acompanhamento segue as ideias da autodeterminação e motivação extrínseca apresentada por Ryan, Deci (2000) e Canto (2015), pois contribui para verificar quais estudantes estão intrinsecamente motivados e quais necessitam de ações para incrementar o esforço e a confiança.

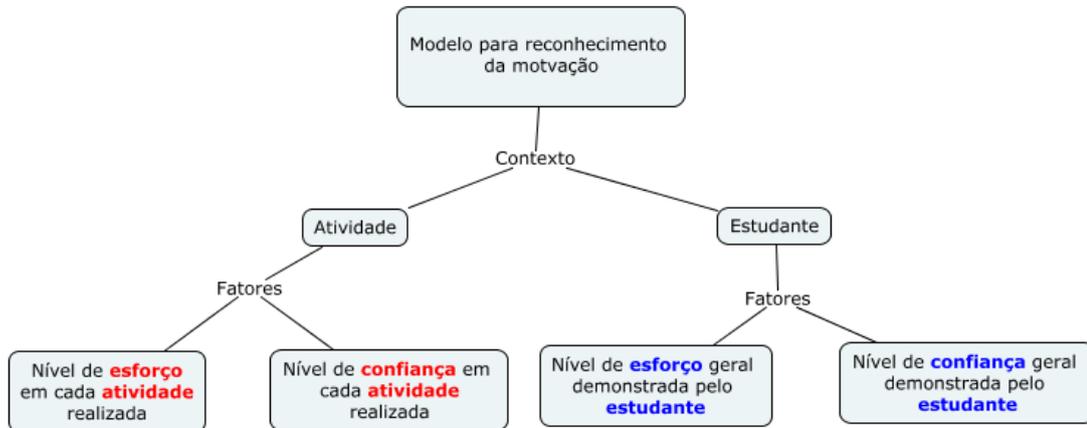
Antes da apresentação do modelo e das suas características, se faz necessária uma discussão sobre o uso do termo modelo e a abrangência do mesmo. Em síntese o modelo referido até o momento, no texto, é composto por diversos modelos menores ou específicos, cada um sendo responsável por reconhecer um dos fatores, no contexto de uma atividade ou do estudante.

De agora em diante o conceito de modelo será apresentado de forma específica, modelos probabilísticos para reconhecimento da confiança ou esforço em uma atividade ou para cada aluno. Conforme citado anteriormente, as Redes Bayesianas se constituíram no método adotado para a construção dos modelos específicos. Considera-se portanto que cada rede bayesiana construída personifica o modelo probabilístico para classificação do nível associado a cada um dos fatores (esforço ou confiança).

Embora o uso da mesma palavra “modelo” nos diferentes sentidos possa causar alguma confusão, não foi encontrada uma definição mais adequada para conceituar tanto o modelo completo quanto os específicos e por este motivo optou-se pelo uso do mesmo termo. Em alguns momentos as expressões “modelo geral”, “modelo específico” serão adotadas para conceituar o nível macro e os componentes específicos.

Para atender o objetivo de identificar os dois fatores, foram propostos quatro modelos específicos com objetivos distintos, sendo dois para classificar os fatores da motivação associados a cada atividade realizada, e outros dois para classificação do estudante, considerando todas os exercícios propostos e resolvidos até o momento. Todos os modelos foram elaborados com base nas definições dos fatores ECI (Esforço, Confiança e Independência) encontrados no trabalho desenvolvido por Del Soldato e Du Boulay (1995), revisado em Du Boulay e Del Soldato (2016). A figura 22 apresenta a hierarquia dos modelos citados, considerando o objetivo e o contexto.

Figura 22 – Visão geral dos modelos para reconhecimento da motivação



Fonte: Elaborado pelo autor

Em todos os casos o nível de esforço e confiança é definido como alto ou baixo, além de uma probabilidade associada a cada classe, ou seja, a rede retorna uma probabilidade do estudante ter demonstrado um alto ou baixo esforço, por exemplo. Os modelos para o reconhecimento da motivação no contexto de atividade, recebem como entrada os dados coletados na resolução de cada atividade e a partir destes indicam as probabilidades de cada classe. Já no contexto do estudante os atributos possuem valores que resumem os dados de todas as atividades entregues pelo aluno, ao longo do semestre.

Para identificar esforço e confiança foi definido um conjunto de observáveis relacionados às ações do aluno durante a resolução das atividades. Estes observáveis podem ser categorizados da seguinte maneira:

- Tempos: duração (em minutos) de cada etapa, pontos-chave, hipótese e código fonte. Os tempos são utilizados especialmente na identificação do esforço, uma vez que um tempo maior pode indicar mais dedicação ou persistência e por consequência, maior esforço (Dundunmaya, Banwan e Rodrigo, 2018). No caso da confiança, o tempo é avaliado para verificar se está muito acima dos demais, o que pode indicar uma menor confiança.
- Níveis de detalhamento: o nível de detalhamento das respostas dos estudantes foi usado também como indicativo de maior ou menor esforço. De forma análoga ao número de tentativas usadas por Del Soldato e Du Boulay (1995) é importante para indicar se o estudante tentou, se dedicou para compreender e especificar o problema.

- Acesso às opções de ajuda: o uso das dicas disponibilizadas foi adotado como um dos principais indicativos da confiança demonstrada pelo estudante. Para o esforço foi usado apenas o acesso ao pseudocódigo que é uma “quase solução” para o problema, portanto o acesso a este conteúdo também pode ser indicativo de menor esforço.
- Outros: atributos complementares, que indicam o nível de compreensão demonstrado pelo estudante, um indicativo quanto à execução ou não do código, uma avaliação por parte do aluno quanto à complexidade da atividade e o número de vezes que o aluno interrompe e retoma a tarefa.

O principal critério para definição de esforço neste trabalho é a percepção de que o aluno se dedicou, mostrou disposição para compreender e detalhar o problema antes de iniciar a escrita do código. Sendo assim, as variáveis representam em sua maioria informações sobre as duas primeiras etapas, a descrição dos pontos-chave do problema e da hipótese. Dados sobre a elaboração do código fonte também são considerados, mas com um peso menor. Segundo Del Soldato e Du Boulay (1995, 2016) o esforço pode ser medido mais pelas tentativas, dedicação do estudante do que pelo desempenho ou resultado final, desta forma, a avaliação da qualidade da solução final pode ser apenas um indicativo, mas não é fundamental para indicar os níveis de esforço.

Embora confiança e independência sejam considerados diferentes fatores, alguns atributos apontam tanto para uma maior ou menor confiança, quanto independência. Nesta pesquisa os principais aspectos considerados nos dois fatores (confiança e independência) dizem respeito ao acesso as opções de ajuda disponibilizados no sistema e em uma resolução mais padronizada ou linear do problema, ou seja, estudantes que iniciam e param diversas vezes ou que em alguns casos levam um tempo muito acima do normal podem estar apresentando baixa confiança e indiretamente, menor independência.

A identificação destes fatores individualmente para cada tarefa permite que o professor perceba rapidamente se algum estudante demonstra baixos níveis de esforço e confiança, mesmo que a disciplina ainda esteja no início e poucas atividades tenham sido desenvolvidas. Desta forma é possível perceber e agir para estimular um maior esforço e aumentar a confiança nas tarefas seguintes. A classificação dos níveis associados a cada tarefa também contribui para que o docente verifique a evolução do estudante ao longo da disciplina, se os níveis de esforço e confiança têm aumentado, diminuído ou se estes se mantêm estáveis.

Além do acompanhamento do comportamento dos estudantes, os modelos podem ser usados também para avaliar as atividades propostas, se estas podem ter sido mal formuladas, mal compreendidas ou se não contribuíram para estimular o esforço do estudante. Ao constatar, por exemplo que em um determinado problema a maioria dos estudantes demonstrou baixo esforço ou pouca confiança, pode ser necessário avaliar em conjunto com os docentes os motivos que levaram a esta situação e, eventualmente propor mudanças na elaboração do problema para as turmas seguintes.

2.6.4 Mineração de dados educacionais e sua utilização no modelo afetivo

A mineração de dados é considerada como parte do processo de descoberta de conhecimento em bancos de dados ou conjuntos de dados, desta forma, um dos objetivos da aplicação das técnicas de mineração é encontrar padrões que uma observação humana poderia ignorar ou desconsiderar na análise dos dados. Os resultados da mineração podem se constituir também em modelos que podem ser aplicados em dados futuros para prever ocorrências ou comportamentos, o que contribui para que sejam tomadas ações de forma antecipada para evitar situações indesejadas.

Segundo Tan (2009), as tarefas de mineração de dados podem ser classificadas em diferentes categorias, entre estas está a modelagem de previsão, na qual se enquadra o modelo desenvolvido. Na previsão busca-se construir um modelo para um atributo ou variável alvo como uma função das variáveis explicativas (TAN, 2009).

O foco principal está na identificação da motivação dos estudantes em disciplinas de programação, na tentativa de criar modelos que permitam que professores ou tutores percebam que determinados estudantes apresentam baixos níveis de esforço ou confiança. Tal percepção pode auxiliar na proposição de ações que tem como objetivo melhorar a confiança ou estimular os estudantes a se tornarem mais ativos, mais envolvidos nas tarefas desenvolvidas.

Considerando os aspectos citados, foi possível inferir que técnicas de mineração de dados poderiam ser uma boa alternativa para obter um modelo, a partir dos dados coletados, que auxiliasse na tarefa de identificar os níveis de motivação demonstrados pelo estudante. As técnicas para classificação supervisionada foram utilizadas no contexto deste trabalho devido à possibilidade de aplicação das mesmas sobre um conjunto de dados para que aprendessem a classificar entradas de acordo com os rótulos definidos a priori.

Quando uma técnica supervisionada é utilizada, após o processo de treinamento, é possível observar como foi realizada a classificação, quais as probabilidades ou regras usadas para determinar a classe. Citando como exemplo os métodos baseados em regras ou árvores de

decisão, é possível visualizar uma árvore que se constitui no conjunto de regras usadas para classificar as entradas. Outros métodos, como os que utilizam estratégias probabilísticas, tais como Redes Bayesianas permitem definir as dependências entre as variáveis e as probabilidades associadas aos diferentes atributos.

Embora fosse possível elaborar um modelo baseado somente na observação dos dados e análise estatística feita pelo pesquisador, as técnicas de classificação, contribuiram para obter indicadores quanto à acuracidade e confiabilidade do modelo computacional para reconhecimento da motivação.

É preciso levar em consideração também que o reconhecimento dos níveis de motivação dos alunos nem sempre é algo que pode ser realizado usando regras ou uma abordagem determinística. Em diversas situações o nível de esforço ou confiança, pode ser alto ou baixo, mas com um certo grau de certeza, ou seja, há uma maior ou menor chance de ser maior ou menor. Sendo assim, a aplicação de técnicas para mineração de dados baseadas na distribuição de probabilidades se tornou uma alternativa importante para a determinação não apenas do nível de motivação, mas também da probabilidade de que uma determinada atividade ou estudante esteja apresentando um esforço ou confiança baixo ou alto.

Uma maior probabilidade da confiança ou esforço ser alto, por exemplo, pode indicar uma maior intensidade no estado afetivo, ou seja, uma maior propensão a estar motivado, o que permitiu constatar em algumas situações uma motivação intrínseca do aluno (RYAN e DECI, 2015). Por outro lado, uma maior probabilidade de apresentar índices baixos de esforço ou confiança requer por parte do professor a execução de ações que visam aumentar a motivação extrínseca, tais como, desafios, punições ou o reconhecimento quando este apresentar comportamento desejado.

As técnicas de mineração de dados foram empregadas também como ferramentas para exploração de um conjunto de dados, ainda na fase de preparação dos dados (FAYAD, PIATETSKY-SHAPIRO e SMYTH, 1996). Em um primeiro momento foram adotadas medidas de estatística descritiva para avaliar os dados brutos e para categorizar ou discretizar dados numéricos. Medidas de tendência como média, desvio padrão, ou de análise de faixas de valores como quartil, percentil foram uteis para avaliar a concentração ou distribuição dos dados, embora não tenham sido suficientes para analisar, entre outras questões, a similaridade das ocorrências, além de possíveis dependências entre as variáveis.

Técnicas para agrupamento contribuiram para aumentar o conhecimento sobre as características comuns compartilhadas pelas diferentes entradas, e para a identificação de

variáveis e valores que diferenciam um grupo de registros dos demais (TAN, STEINBACH e KUMAR, 2009). O método para agrupamento contribuiu ainda para uma classificação a priori necessária para o processo de treinamento em métodos supervisionados.

As técnicas de mineração de dados foram usadas com dois objetivos e em diferentes momentos no processo de desenvolvimento e aplicação do modelo afetivo:

- Análise exploratória: durante a preparação dos dados, técnicas baseadas em agrupamento e em regras de associação serão utilizadas explorar o conjunto de dados e para apoiar a seleção e preparação dos mesmos para a aplicação de técnicas de classificação supervisionada.
- Construção e treinamento das redes bayesianas: técnicas baseadas em classificação supervisionada, especialmente Redes Bayesianas foram aplicadas para a obtenção de modelos usados na classificação do esforço e da confiança. Métodos de classificação baseados em árvore de decisão foram usados como técnicas complementares para avaliar a eficiência das RB.

As técnicas para classificação supervisionada foram aplicadas na fase final da elaboração do modelo, após a preparação dos dados e definição das tabelas para classificação a priori, utilizadas para determinar as classes do conjunto de entradas usadas no treinamento. As entradas do processo de treinamento consistem nos dados coletados durante o estudo de caso, nos tempos e nas avaliações feitas pelo professor e pelo pesquisador.

Foram utilizados os dois softwares descritos anteriormente, o Weka e o Netica. A ferramenta Weka foi utilizada devido à disponibilidade de um amplo conjunto de algoritmos e a experiência anterior do pesquisador na utilização desta. O software foi empregado para a aplicação e análise do desempenho dos algoritmos, especialmente das Redes Bayesianas. Os relatórios e indicadores mostrados após a aplicação de um método serão apresentados no capítulo 5 com o objetivo de ilustrar a confiabilidade do modelo.

Já o software Netica foi usado na construção e treinamento das redes que permitem reconhecer esforço e confiança. A facilidade de estruturar e treinar uma rede foi decisiva para adoção desta ferramenta, pois a mesma permite criar e remover dependências entre os atributos, sem a necessidade de alterar os dados de treinamento. As bibliotecas (APIs) do Netica foram usadas na implementação do protótipo do módulo de acompanhamento da motivação que foi utilizado na validação do modelo afetivo. O desenvolvimento e aplicação do referido módulo é apresentado no capítulo 5.

3 PROALG: SISTEMA DE APOIO AO USO DA PROBLEMATIZAÇÃO NO ENSINO DE ALGORITMOS

Neste capítulo serão apresentadas as principais funcionalidades e recursos do sistema PROALG, que foi desenvolvido no contexto desta pesquisa, com o intuito de suportar o método da problematização e permitir a coleta dos dados utilizados na elaboração e validação do modelo para o reconhecimento da motivação dos estudantes.

O desenvolvimento e aplicação do software atende a dois objetivos descritos na seção 1.5, o desenvolvimento e validação de um sistema que permita a resolução de problemas utilizando a problematização e, a automação da coleta de dados para elaboração e validação do modelo para reconhecimento da motivação. O sistema foi implementado também para ser uma ferramenta de apoio em disciplinas de programação, permitindo o gerenciamento das turmas e exercícios que são resolvidos pelos estudantes.

O software foi projetado considerando dois pressupostos fundamentais: i) a problematização como base para o método de apoio ao ensino e aprendizagem na resolução de problemas de programação (BERBEL, 2012) e; ii) a coleta de dados para o reconhecimento do esforço e da confiança (DU BOULAY e DEL SOLDATO, 2016). O método da problematização foi apresentado na seção 2.6.1 e o modelo para o reconhecimento da motivação, que utiliza os dados coletados no sistema, será detalhado no capítulo 5.

Serviram de inspiração para a especificação das funcionalidades, aplicativos existentes, como o URI Online Judge (BEZ, TONIN, RODEGHERI, 2014) e o software apresentado no trabalho desenvolvido por Iepsen (2013). O primeiro apresenta recursos para manutenção de turmas de estudantes e para a publicação e resolução de exercícios de programação. O segundo se constitui em um ambiente para edição de código e coleta de dados que posteriormente são utilizados em um processo de mineração de dados.

A necessidade de implementar uma ferramenta que permitisse gerenciar estudantes e exercícios de programação foi percebida no início da pesquisa, ainda durante o estudo piloto, cujos resultados serão apresentados no capítulo 4. O AVEA Moodle foi utilizado nas atividades desenvolvidas no estudo piloto, momento em que foi possível perceber algumas dificuldades associadas a sua utilização. No estudo piloto, a coleta de dados dependeu dos logs do sistema e nem todas as informações estavam disponíveis, o que aumentou consideravelmente o tempo de análise dos dados.

Além da dificuldade relacionada à coleta de dados, no estudo piloto a etapa de escrita do código fonte não foi realizada de maneira integrada, ou seja, os estudantes elaboravam o código fonte em outra ferramenta, postando o mesmo na lição quando estava finalizado. Esta abordagem não permitiu a obtenção de informações sobre as ações durante a escrita do código, embora tenha sido possível constatar que os dados relacionados a esta etapa seriam importantes para o reconhecimento da motivação. Ficou claro que dados relacionados ao número de compilações, erros, execuções, além de outras ações, como copiar e colar poderiam contribuir para identificar os fatores associados à motivação.

Além da integração de todas as etapas, o sistema implementado possui recursos para que o professor acompanhe os resultados de cada atividade resolvida por cada um dos estudantes. É possível efetuar uma avaliação de cada exercício, além de visualizar estatísticas e informações relacionadas aos tempos de cada etapa, os conteúdos das respostas e os acessos às dicas disponibilizadas.

O sistema foi projetado e desenvolvido no contexto desta pesquisa, mas foi também o tema do Trabalho de Conclusão (TCC) desenvolvido por Hemming (2018). O trabalho citado foi orientado e supervisionado pelo autor da presente tese, sendo assim é possível considerar que o mesmo se constitui em um dos resultados indiretos desta pesquisa, o que demonstra a abrangência deste trabalho. Além do referido trabalho de conclusão, os resultados do desenvolvimento e aplicação do PROALG foram publicados na revista Destaques Acadêmicos (FRANZEN, HEMMING, BERCHT, 2019). O referido software também foi registrado no INPI (Instituto Nacional de Proteção Intelectual) no ano de 2018 (Apêndice D).

Na sequência serão apresentados os requisitos funcionais, a arquitetura e as tecnologias utilizadas na implementação, além das principais interfaces que demonstram o funcionamento dos recursos da ferramenta. Uma ressalva importante diz respeito ao módulo para acompanhamento da motivação que será apresentado com um requisito funcional, mas que será descrito com detalhes no capítulo 5. Como o módulo se baseia no modelo para reconhecimento da motivação, as explicações sobre o mesmo serão inseridas após o capítulo 5, no qual será descrito o referido modelo.

3.1 Requisitos funcionais

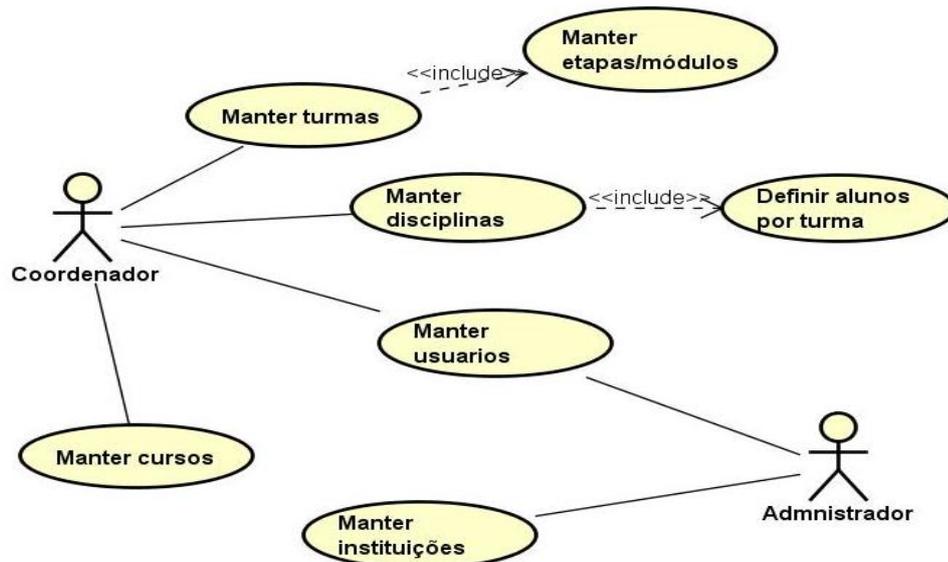
Com o intuito de mostrar os recursos que foram disponibilizados na ferramenta, foram elaborados diagramas de casos de uso (BOOCH, RAUMBAUGH, JACOBSON, 2006). Um caso de uso pode ser definido como uma funcionalidade do sistema, que gera algum resultado

para um ator. O ator pode ser um tipo de usuário em um sistema ou simplesmente uma função exercida por alguém, em uma organização.

Neste sistema são quatro atores principais, os administradores e coordenadores, que tem acesso aos recursos gerais, associados a configuração, administração do sistema, os professores, que gerenciam, propõe e avaliam exercícios de programação e os alunos, que utilizam o sistema para resolver as atividades propostas. A seguir serão apresentados três diagramas, segmentados de acordo com as funcionalidades relacionadas a cada um dos atores.

O diagrama mostrado na figura 23 ilustra principalmente o conjunto de requisitos que apoiam a manutenção dos cadastros essenciais, que permitem que a ferramenta seja utilizada em diferentes instituições, por diversas turmas, professores e estudantes. Tais recursos também contribuíram para a adoção da ferramenta ao longo da pesquisa, nas diferentes etapas do processo.

Figura 23 – Diagrama de casos de uso para a coordenação e administração



Fonte: Elaborado pelo autor

O administrador é responsável somente por definir os usuários e as instituições de ensino nas quais o sistema será utilizado, como ilustrado na Figura 23. O coordenador mantém os cursos, as disciplinas e as turmas que ocorrem a cada um dos semestres. Disciplinas são vinculadas às turmas, que por sua vez são compostas por um conjunto de alunos (usuários) aos quais é concedido acesso as atividades propostas pelo professor.

Figura 24 – Diagrama de casos uso relacionado ao professor



Fonte: Elaborado pelo autor

Na figura 24 é possível observar que cabe ao professor ou os diversos professores que ministram disciplinas de programação, manter um conjunto de exercícios que poderão ser aplicados ao longo das ocorrências das disciplinas. A funcionalidade para manutenção de templates de atividade permite o cadastro destes exercícios.

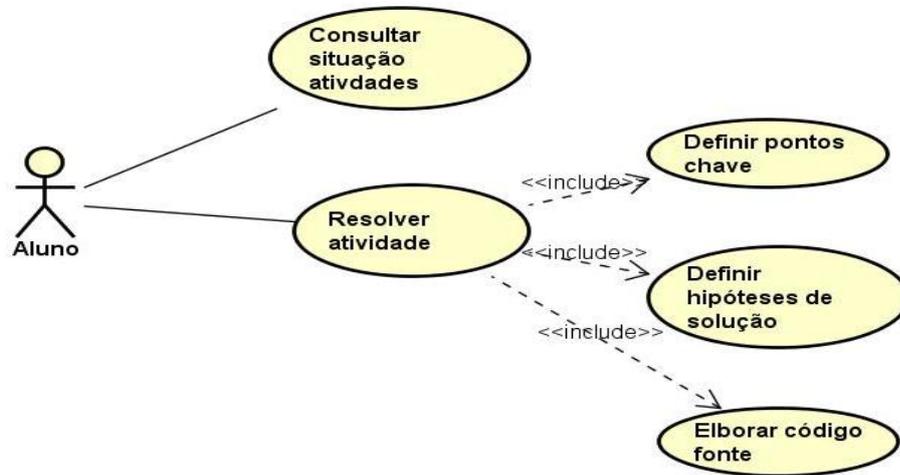
A manutenção de template de atividades é um recurso importante pois contribui para a continuidade do uso do software, uma vez que um *template* pode ser publicado para diferentes turmas. No contexto da presente pesquisa este recurso foi importante para que fossem coletados dados das mesmas atividades em turmas diversas. Um template se constitui em uma descrição, protótipo de uma atividade que será usada posteriormente por um professor para publicar um exercício para os alunos de uma determinada turma.

O acompanhamento da motivação dos estudantes pelo professor baseia-se no modelo para reconhecimento da motivação e tem como principal ferramenta o módulo que foi implementado para utilizar as Redes Bayesianas. Embora o módulo seja um recurso de software relacionado ao sistema PROALG ele será apresentado no capítulo 5, após a explanação sobre a construção do modelo para reconhecimento da motivação, uma vez que a compreensão do modelo é fundamental para o entendimento do módulo.

Em uma determinada turma, tendo como base um *template*, um professor publica uma atividade que deverá ser realizada pelos alunos associados a mesma. O professor terá acesso ainda a recursos para visualização de estatísticas sobre os dados coletados durante as atividades e aos indicativos sobre a motivação demonstrada pelos estudantes durante a realização das

tarefas. Tais recursos se relacionam diretamente aos objetivos deste trabalho, que busca apoiar o ensino de programação, considerando a dimensão afetiva.

Figura 25 – Diagrama de casos de uso das funcionalidades relacionadas ao aluno



Fonte: Elaborado pelo autor

O aluno utiliza o sistema para verificar as atividades que foram propostas pelo professor e para resolver as mesmas (Figura 25). É esta funcionalidade que implementa os passos descritos no método da problematização, descrito na seção 2.6.1. Conforme a configuração do professor na publicação da atividade, o sistema apresenta um fluxo direcionado à resolução baseada na problematização ou não.

3.2 Arquitetura e tecnologias

As tecnologias e a arquitetura do sistema são mostradas na figura 26. Um dos principais requisitos não funcionais foi a definição de que o software deveria ser implementado usando tecnologias para plataforma web. O sistema foi hospedado em um servidor da empresa Digital Ocean (<https://proalg.cf>), o que garantiu a disponibilidade e o uso em qualquer dia e horário.

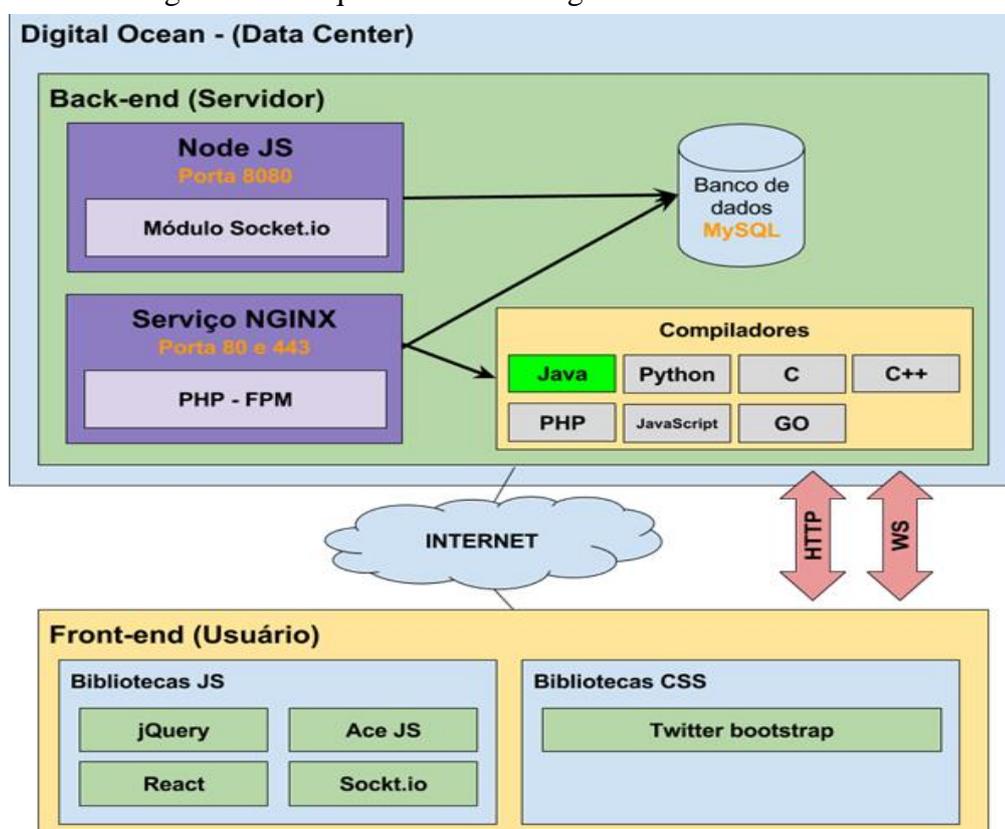
O NGINX, um servidor web que se caracteriza por ser rápido e consumir poucos recursos computacionais foi utilizado para interpretar os códigos fontes escritos na linguagem de programação PHP (*Hypertext Preprocessor*). O NodeJS, é um interpretador para códigos JavaScript que é executado no servidor, sendo responsável por interpretar os códigos fontes escritos nesta linguagem.

No lado cliente foi empregado o JavaScript, na versão ES6 e o CCS (*Cascading Style Sheets*) na versão 3. Para criação da interface do editor de código fonte, foi adotado o framework React na versão 15. Para o desenvolvimento do sistema e elaboração da maioria das telas foi utilizado o gerador de códigos Code Charge Studio na versão 5.1, por ser uma ferramenta que

umenta a produtividade, reduz o tempo de desenvolvimento, a partir da geração de códigos em linguagem PHP.

A coleta dos dados durante a realização das tarefas foi realizado usando *WebSockets*, com a biblioteca *Socket.io* 5. Para o armazenamento dos dados foi utilizado o banco de dados relacional *MySQL* na versão 5.7 a engine de armazenamento *InnoDB*. Para a compilação do código fonte das soluções elaboradas pelos alunos, foi usado o compilador *JAVAC* na versão 8. O projeto do software inclui a possibilidade futura de outras linguagens de programação, porém a atual versão permite somente a compilação de códigos escritos na linguagem *Java*.

Figura 26 – Arquitetura e tecnologias usadas no PROALG

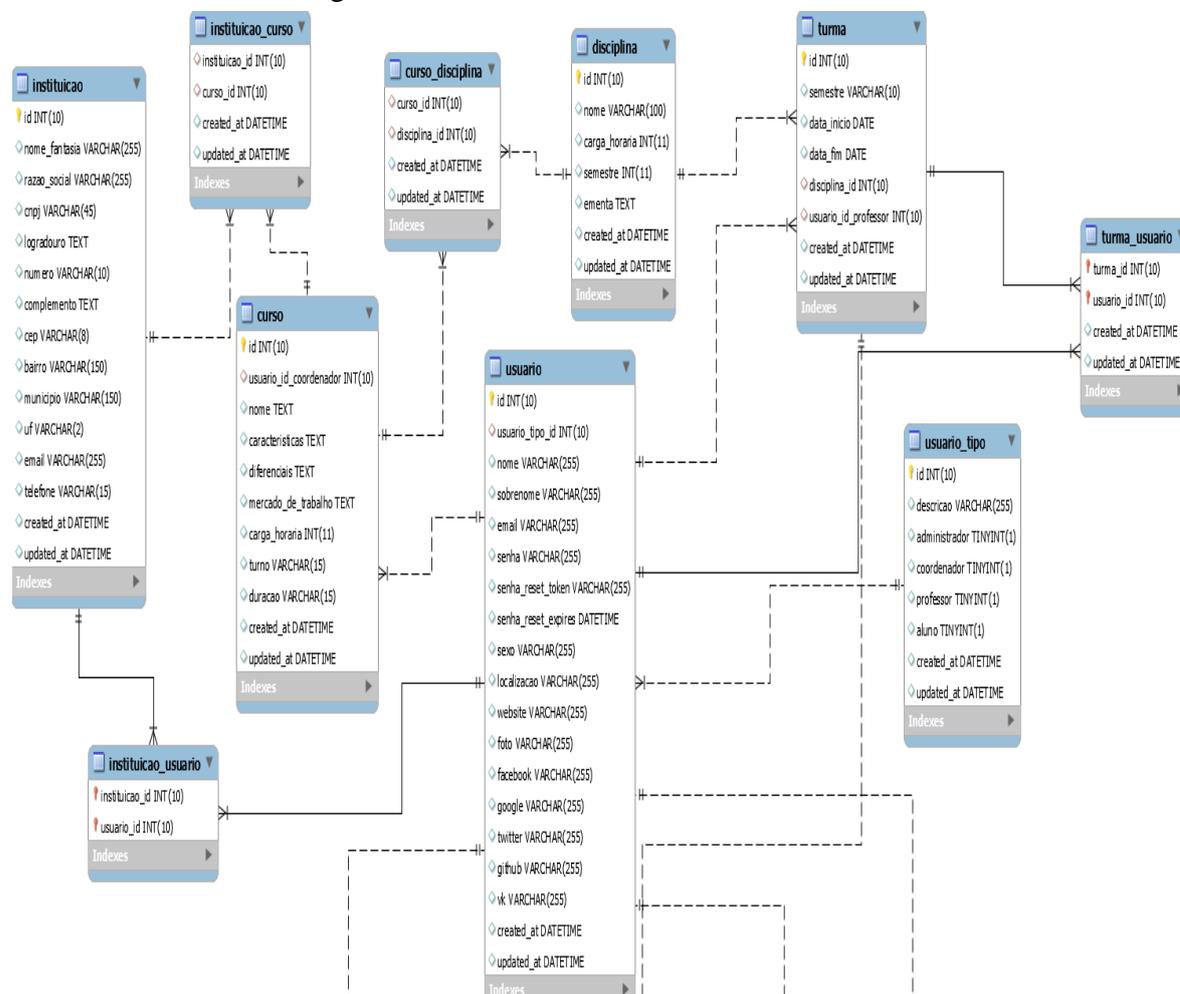


Fonte: Hemming (2017)

3.3 Projeto da base de dados

O modelo ER (Entidade Relacionamento) é uma ferramenta utilizada para demonstrar o projeto da base de dados que suporta um sistema (HEUSER, 2009). O modelo ER relacional apresenta detalhes sobre as tabelas, atributos, além das restrições de integridades (chaves primárias, chaves estrangeiras). Na figura 27 são apresentadas as tabelas básicas, cadastros de usuários (alunos e professores), cursos, disciplinas, turmas, entre outras.

Figura 27 - Modelo ER com as tabelas básicas

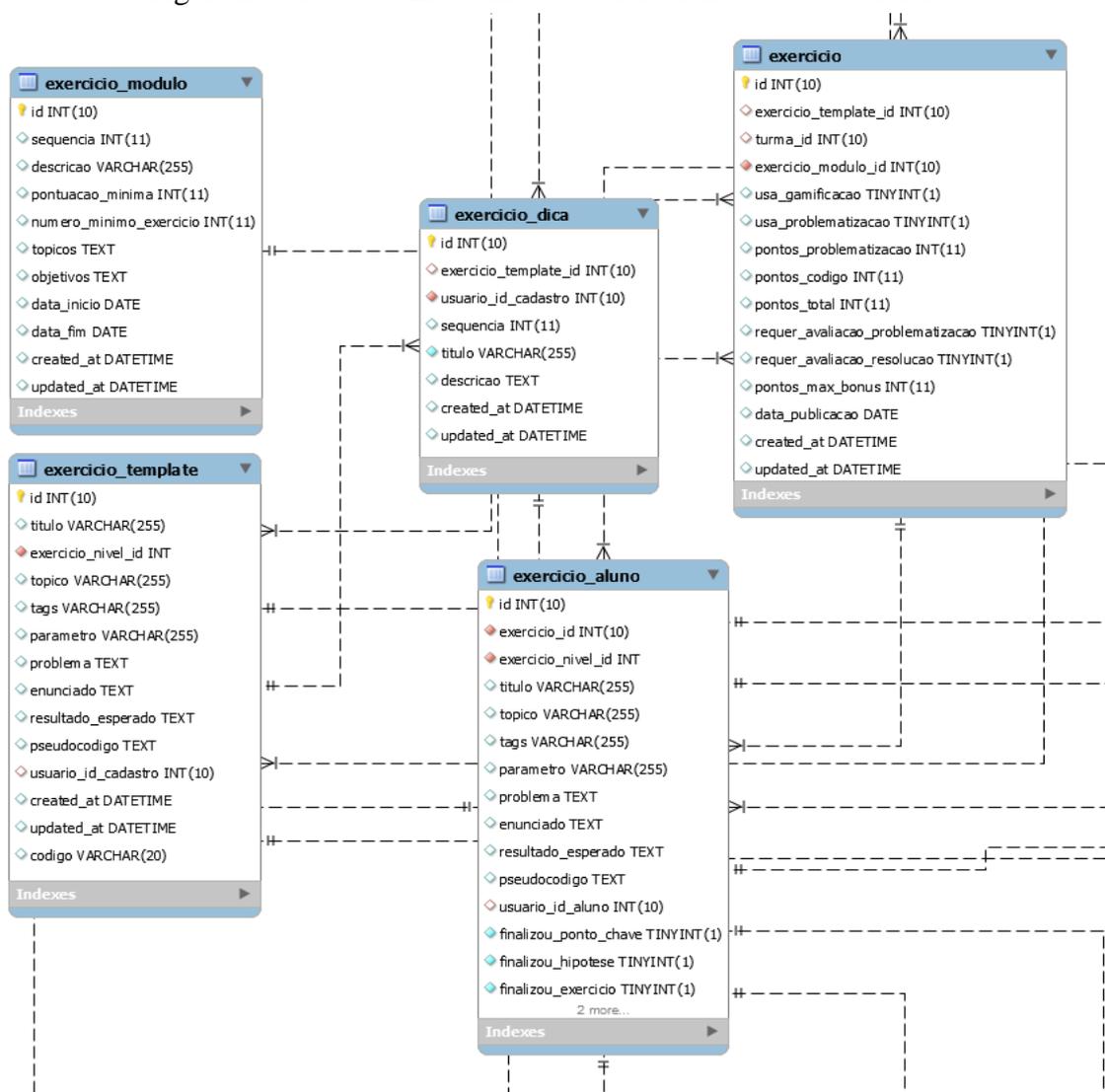


Fonte: Elaborado pelo autor

Foram definidas tabelas para armazenar os exercícios desde os templates (exercicio_template), até o registro de cada atividade que deve ser resolvida pelo estudante (exercicio_aluno). A tabela exercicio_dica, por exemplo, é responsável por guardar as dicas que serão exibidas no momento em que o estudante estiver resolvendo a atividade (Figura 28).

Há ainda um conjunto de tabelas cujo propósito é armazenar todas as ocorrências e ações do estudante enquanto este realiza uma tarefa, sendo a principal delas denominada ocorrencia_resolucao (Figura 29). O conteúdo postado para cada questão, as incidências de entrada e saída da atividade, são registradas nesta tabela, juntamente com a data e a hora em que ocorreram. Foram propostas tabelas para armazenar ações específicas, como o acesso as dicas (log_visualizacao_dica, log_visualizacao_pseudocodigo), além de detalhes sobre a escrita do código fonte, como ações de copiar/colar, edição de trechos da soluções (log_editor_codigo).

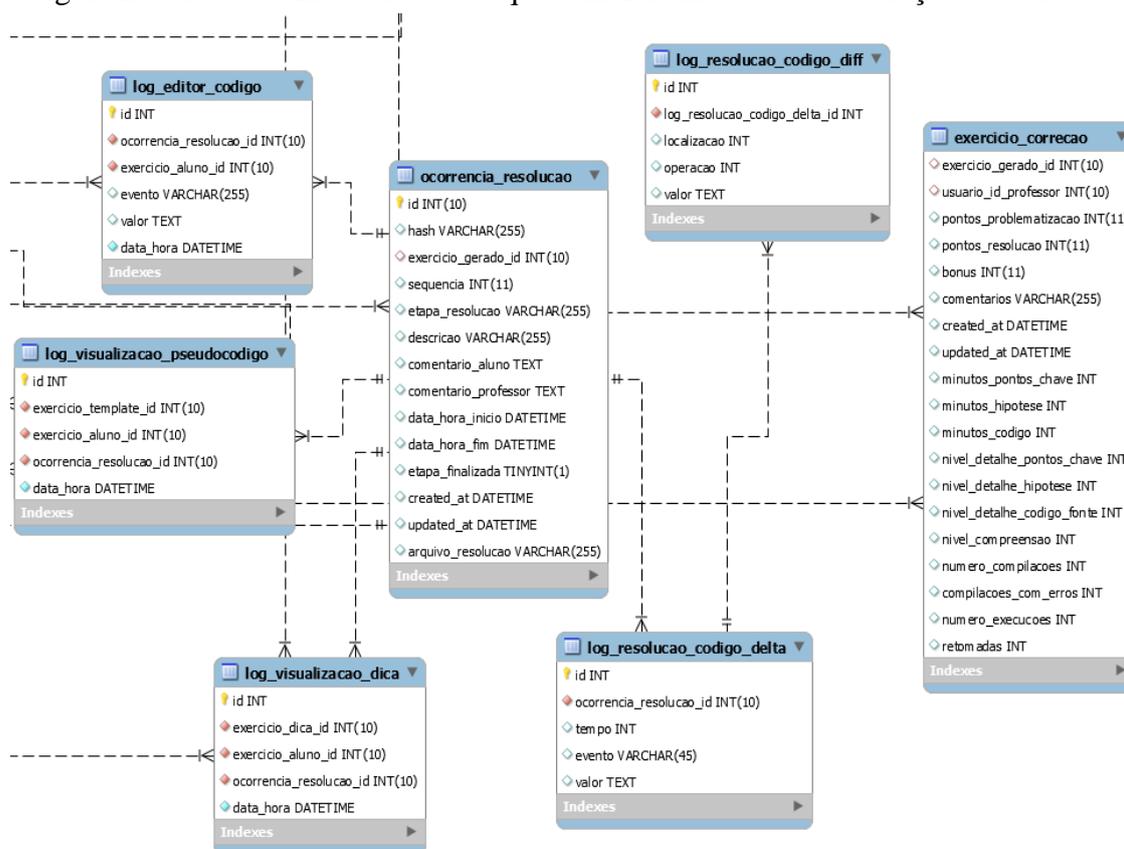
Figura 28 – Modelo ER com tabelas relacionadas aos exercícios



Fonte: Elaborado pelo autor

A tabela `exercicio_correcao` é de fundamental importância, uma vez que na mesma são registrados os dados da avaliação que o professor faz de cada atividade. Os tempos de duração de cada etapa, os níveis de detalhamento e o nível de compreensão que são atribuídos pelo professor na avaliação da tarefa são mantidos nesta tabela. A mesma se constitui na principal fonte dos dados usados na elaboração do modelo para o reconhecimento da motivação do estudante, pois, a partir dos dados gravados nesta tabela são aplicadas as regras de transformação e pré-processamento que geram os dados para as tabelas usadas no módulo de acompanhamento da motivação.

Figura 29 – Modelo ER com tabelas que armazenam dados da resolução das atividades



Fonte: Elaborado pelo autor

3.4 Interfaces do sistema

As interfaces mostradas nesta seção ilustram parte das funcionalidades e a forma de interação dos diferentes atores com o sistema. As telas não mostradas nesta seção podem ser visualizadas no Apêndice A.

Figura 30 - Manutenção de turmas

Fonte: Elaborado pelo autor

A estrutura geral da interface do sistema exibe o título na barra superior, além do menu principal, que foi colocado no lado esquerdo. As interfaces de manutenção seguem um padrão com botões para atualização e exclusão na parte inferior (Figura 30). Os menus mudam de acordo com o usuário, se o mesmo é um professor ou aluno. Usuários do tipo aluno podem visualizar somente os exercícios publicados e que devem ser resolvidos. Um botão é exibido no lado direito da atividade para indicar o status, sendo possível visualizar um exercício finalizado ou resolver a uma atividade ainda não iniciada ou parcialmente resolvida.

No sistema são mantidos templates de atividades, ou seja, definições gerais para exercícios que posteriormente serão alocadas para diferentes turmas. Este recurso atende um dos objetivos do sistema que é permitir a manutenção de um banco de exercícios. A interface mostrada na figura 31 mostra a tela que permite incluir ou alterar os dados de um template.

Figura 31 - Manutenção de template

The screenshot shows a web form titled "Adicionar/Alterar Template" with the following fields and content:

- Título:** Publico e renda
- Nível:** Baixo
- Tópicos:** Entradas, saídas, instruções sequenciais
- Tags:** Entradas, saídas, instruções sequenciais
- Parâmetros:** Públicos em arquibancada e renda.
- Problema:** A final do campeonato regional vai ocorrer no próximo final de semana. Espera-se um grande público. O estádio no qual vai acontecer a final possui dois tipos de acomodações, as arquibancadas e as cadeiras numeradas. As pessoas que compram cadeiras pagam um valor de 80 reais por cada ingresso e quem fica em arquibancada paga 40 reais. O estádio possui um total de 2000 lugares, considerando tanto arquibancada quanto cadeiras. Para facilitar a apuração da renda e da ocupação, crie uma solução para calcular a renda total, a partir do
- Enunciado:** Faça um programa que receba a quantidade de público presente nas cadeiras e nas arquibancadas de um estádio de futebol, em uma final de campeonato. Calcule e mostre o percentual de ocupação e a renda total do jogo, considerando que o público nas cadeiras paga 80 e nas arquibancadas 40 reais. Neste caso deve-se multiplicar o público presente em cada local pelo valor pago para obter a renda total. No estádio cabem 2000 pessoas, sendo assim, é necessário dividir o público total pela capacidade para
- Pseudocódigo:**

```

programa renda_publico
var cadeiras : inteiro
var arquibancadas: inteiro
var renda: real
var ocupacao: real
inicio

```

Fonte: Elaborado pelo autor

Cada template está associado a diferentes tags e tópicos que podem ser usados para indicar os conteúdos e recursos associados ao exercício. Observa-se aqui a definição do problema, que é exibido para o estudante na primeira etapa, a definição dos pontos-chave e o enunciado tradicional, mostrado na etapa final, quando é elaborado o código fonte. Uma

discussão sobre os padrões e aspectos essenciais que devem ser considerados na definição do problema foi apresentada na seção 2.6.

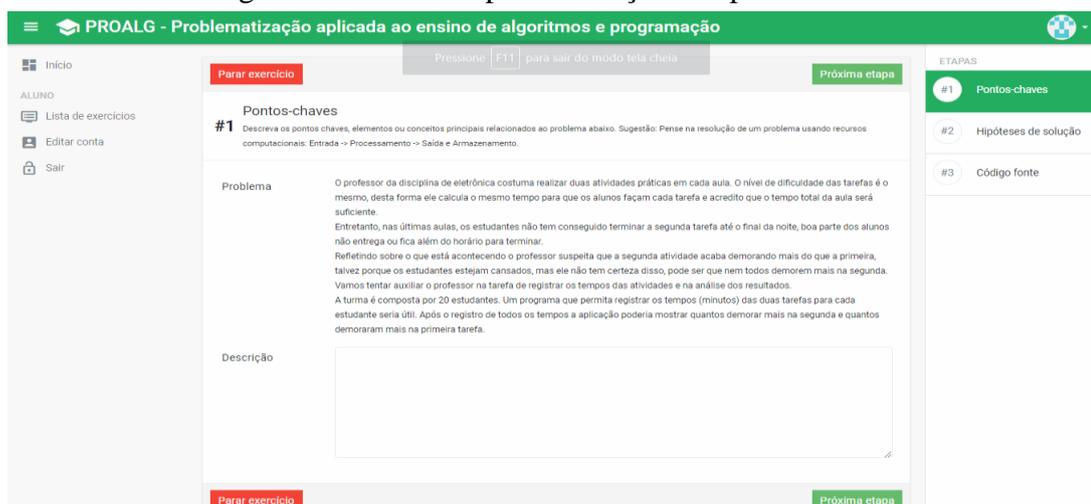
Para cada template são definidas dicas que podem ser acessadas pelo usuário (Apêndice A). A disponibilização de um conjunto de dicas é importante para o reconhecimento da confiança do estudante e atende a um dos fundamentos propostos por Del Soldato e Du Boulay (1995, 2016). Na metodologia proposta pelos autores, os mesmos estabelecem uma relação entre a confiança, a independência e o fato do estudante ter se valido de recursos de ajuda para realizar uma determinada tarefa. De forma análoga o sistema exibe as dicas durante a tarefa e o estudante decide se deseja visualizá-las, ficando registrada uma ocorrência para cada acesso aos recursos de ajuda.

Um template se converte em uma atividade para os estudantes no momento em que o professor seleciona o mesmo para publicar um exercício. Este recurso permite ao professor selecionar um template existente e a partir deste alocar a atividade para os alunos de uma determinada turma resolverem. O próprio aplicativo gera um registro individual de exercício para cada estudante da turma, armazenando este na tabela `exercicio_aluno`. Ao publicar um exercício é possível indicar que será usada a problematização o que implica na resolução usando as etapas da problematização com o arco de Maguerz (BERBEL e GAMBOA, 2011).

A seguir serão apresentadas as interfaces propostas para a resolução das atividades, as diferentes etapas do método pedagógico aplicado. Os passos da problematização implementados são: i) definição dos pontos-chave do problema; ii) descrição de uma hipótese de solução para o problema e; iii) elaboração do código fonte, que se constitui na solução final para o problema.

Os conteúdos das duas primeiras etapas são textuais, não existe um formato ou regras que devam ser seguidas, desta forma, o estudante tem liberdade para descrever da forma que considera mais adequada. Sempre que um estudante inicia uma etapa o sistema registra na tabela `ocorrencia_resolucao`, a data e hora que iniciou e que foi finalizada a mesma, além de registrar o conteúdo postado.

Figura 32 - Interface para definição dos pontos-chave



Fonte: Elaborado pelo autor

As duas primeiras etapas mostradas nas figuras 32 e 33 correspondem à definição dos pontos-chave e à hipótese de solução. Na primeira etapa o estudante não tem acesso a nenhuma dica, pois o objetivo principal é que ele se concentre na compreensão do problema e na busca por conhecimento que lhe permita especificar possíveis soluções. No passo seguinte, os botões com as dicas são exibidos no lado direito, logo abaixo da indicação do andamento da resolução do exercício.

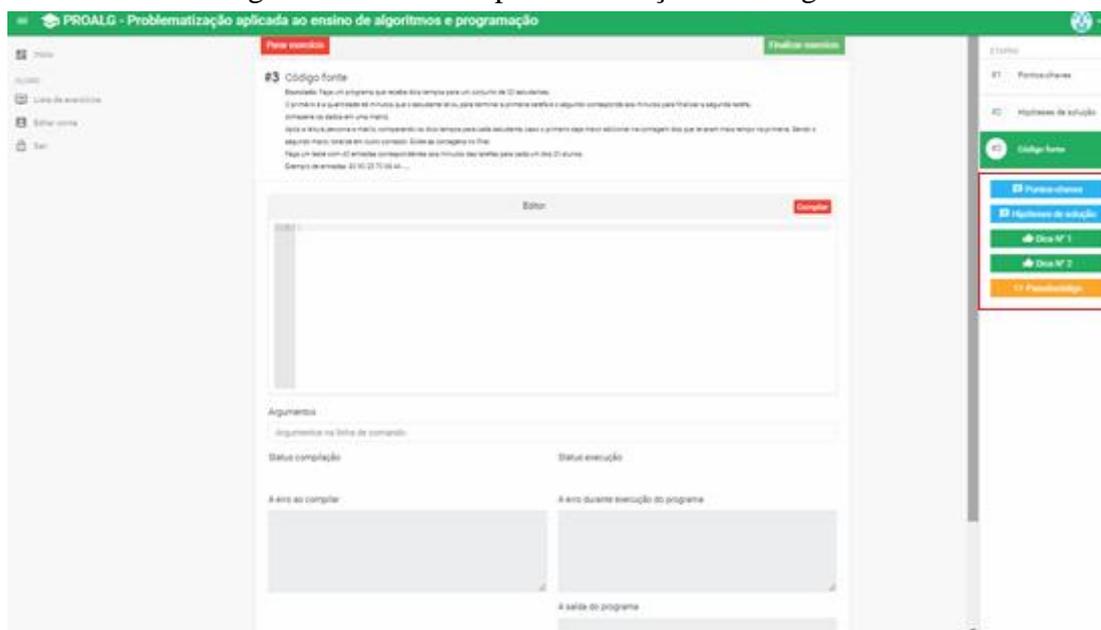
Figura 33 - Interface para definição da hipótese de solução



Fonte: Elaborado pelo autor

Ao finalizar cada uma das etapas, o sistema exibe a interface com o passo seguinte, não sendo possível retornar ao anterior. A última etapa (Figura 34) consiste na elaboração do código fonte da solução e para esta tarefa o sistema exibe um editor que permite a escrita do código, a compilação e a execução de testes, a partir de um conjunto de parâmetros.

Figura 34 - Interface para elaboração do código fonte



Fonte: Elaborado pelo autor

Na figura 34 observa-se que estão disponíveis botões para acesso a todas as dicas, além dos conteúdos postados nas etapas anteriores, onde o estudante pode consultar as definições feitas por ele próprio a respeito do problema. Um aspecto a ser reforçado é que todas as ações realizadas durante as etapas mostradas nas interfaces são registradas, incluindo cancelamentos, retorno ao menu, escrita de código, ações de copiar e colar, compilações, execuções, entre outras.

Após um exercício ter sido entregue pelo estudante, é possível visualizar estatísticas sobre a sua conclusão e efetuar uma avaliação ou correção do mesmo. A avaliação é fundamental para o desenvolvimento do modelo afetivo, uma vez que os dados informados na correção são utilizados para o acompanhamento da motivação exibida pelo aluno.

Inicialmente o sistema mostra a lista das turmas, o número de exercícios publicados e a quantidade de alunos (Apêndice A). Durante a pesquisa foram criadas diversas turmas, algumas apenas com o intuito de testar os recursos da ferramenta, entretanto, boa parte destas fez parte do estudo de caso (2017A e 2017B – T1) e da validação do modelo afetivo (2018B – T1). A figura 35 mostra a listagem dos estudantes que finalizaram uma determinada atividade. É possível verificar as etapas concluídas, além dos resultados da avaliação realizada pelo professor. No lado direito das ocorrências mostradas nesta imagem está o botão que permite avaliar a atividade que foi desenvolvida pelo aluno.

Figura 35 – Listagem dos estudantes que resolveram um determinado exercício

Aluno	Etapas finalizadas	Finalizado	P. problem.	P. resolução	P. bonus	Nível PC	Nível HS	Nível CF	Nível Comp.	Avaliar
[Redacted]	PC - HS - CF	09/08/2018 08:44:33	10 de 10	10 de 10	0	Baixo	Alto	Bom	Alto	[Avaliar]
[Redacted]	PC - HS - CF	09/08/2018 09:31:52	10 de 10	10 de 10	0	Baixo	Médio	Bom	Médio	[Avaliar]
[Redacted]	PC - HS - CF	09/08/2018 08:40:51	10 de 10	10 de 10	0	Baixo	Alto	Bom	Médio	[Avaliar]
[Redacted]	PC - HS - CF	09/08/2018 09:04:20	10 de 10	10 de 10	1	Baixo	Alto	Regular	Médio	[Avaliar]
[Redacted]	PC - HS - CF		0 de 10	0 de 10						

Fonte: Elaborado pelo autor

A interface para avaliação da atividade, mostrada na figura 36 possui um conjunto de recursos, que vão desde a visualização das estatísticas mostradas na parte superior, passando pelo preenchimento da pontuação, níveis de detalhe e compreensão. São exibidas também abas para visualizar os eventos do editor de código, os registros detalhados de acesso aos recursos de ajuda, de compilação e o *code player*, que exhibe uma animação de todo o processo de codificação feito pelo estudante. São mostrados dados sobre os tempos gastos em cada passo da resolução, a avaliação que o estudante fez da complexidade da atividade (médio) e informações sobre compilações, execuções do código fonte. A seguir aparecem as descrições do problema e enunciado que são exibidos quando o estudante acessa a tarefa e na cor verde, os conteúdos para os pontos-chave e hipótese que foram postados pelo aluno (Figura 36).

O código fonte é exibido na segunda parte da tela e após este são mostrados diversos campos preenchidos na correção do exercício, como o nível de compreensão e as pontuações atribuídas pelo professor. É exibido também o comentário feito pelo estudante ao finalizar a atividade e um campo para que o professor escreva um comentário que irá se constituir em um *feedback* para o estudante.

Durante a escrita do código fonte é comum o uso de recursos que incluem copiar e colar conteúdos de outras fontes ou do próprio código que está em construção. Ações de copiar e colar podem ser indicativas interessantes sobre o comportamento do estudante e, por este motivo são registrados na tabela *log_editor_codigo*, sendo disponibilizados para consulta como eventos (Apêndice A).

Figura 36 - Tela de correção de um exercício

Correção

1 Tentativas	6 Ajudas	00:08:22 PONTO-CHAVE	00:13:07 HIPÓTESE	00:44:14 CÓDIGO	01:05:51 TOTAL
Médio Avaliação aluno	0 Compilação abortada	5 Compilação - OK	1 Compilação - ERRO	4 Execução - OK	1 Execução - ERRO

Resultado	Eventos no editor	Ajuda	Compilação	Code player
-----------	-------------------	-------	------------	-------------

Aluno Lucas Germano

Problema
A final do campeonato regional vai ocorrer no próximo final de semana. Espera-se um grande público. O estádio no qual vai acontecer a final possui dois tipos de acomodações, as arquibancadas e as cadeiras numeradas. As pessoas que compram cadeiras pagam um valor de 80 reais por cada ingresso e quem fica em arquibancada paga 40 reais. O estádio possui um total de 2000 lugares, considerando tanto arquibancada quanto cadeiras. Para facilitar a apuração da renda e da ocupação, crie uma solução para calcular a renda total, a partir do número de pessoas que comprou cadeiras e arquibancadas. Além disso, indique o percentual de ocupação total do estádio.

Pontos-chaves
Variáveis → Entradas → Processamento → Saídas

Nível do ponto chave
Baixo

Enunciado
Faça um programa que receba a quantidade de público presente nas cadeiras e nas arquibancadas de um estádio de futebol, em uma final de campeonato. Calcule e mostre o percentual de ocupação e a renda total do jogo, considerando que o público nas cadeiras paga 80 e nas arquibancadas 40 reais. Neste caso deve-se multiplicar o público presente em cada local pelo valor pago para obter a renda total. No estádio cabem 2000 pessoas, sendo assim, é necessário dividir o público total pela capacidade para saber a ocupação.

Hipóteses de solução
Criar variáveis : valor da cadeira, valor da arquibancada, número total de lugares, número de pessoas nas cadeiras, números de pessoas na arquibancada, renda total, porcentagem.
Entradas : fixo = valor da arquibancada e da cadeira; número total de lugares.
leia = número de pessoas na arquibancada e na cadeira, renda total, porcentagem.
Processamento: renda total = (pessoas nas cadeiras X valor da cadeira) + (pessoas nas arquibancadas X valor arquib.)
porcentagem = (pessoas nas cadeiras + pessoas na arquibancada)/número total de lugares
Saída : Mostrar a porra toda

Nível da hipóteses
Alto

Código

```
public class publicoerenda {
    public static void main (String args[]){
        double v1c;
        double v1a;
        double n1l;
        double psc;
        double psa;
        double rnd;
        double pct;
        v1c = Double.parseDouble(args[0]);
        v1a = Double.parseDouble(args[1]);
        n1l = Double.parseDouble(args[2]);
        psc = Double.parseDouble(args[3]);
        psa = Double.parseDouble(args[4]);
        rnd = (v1c*psc)+(v1a*psa);
        pct = ((psc+psa)/n1l)*100.0;
        System.out.printf ("A renda arrecada foi de %.0f e a porcentagem de ocupacao foi de %.2f"
    }
}
```

Nível código fonte
Regular

Pontos problematização
10 de 10 **Pontos resolução** 10 de 10 **Bônus** 1

Nível compreensão
Médio

Avaliação do aluno
Médio

Comentário do aluno

Comentário do professor
Teste

Fonte: Elaborado pelo autor

Um recurso que auxilia o professor é a possibilidade de consultar os acessos aos recursos de ajuda e pseudocódigo disponibilizados ao estudante durante a resolução da atividade. As informações contribuem para analisar a trajetória do aluno, dados que são utilizados no acompanhamento da motivação, especialmente no fator da confiança. Além de consultar o uso das dicas, é possível visualizar os dados sobre cada compilação, quando ocorreu, o tempo, possíveis erros e o código fonte que foi submetido.

O último recurso a ser citado é a opção para reprodução da escrita do código pelo estudante. Enquanto o aluno escreve o código o sistema grava cada alteração nas tabelas `log_resolucao_codigo`, `log_resolucao_codigo_diff` e `log_resolucao_codigo_delta`, o que permite repetir em um formato de vídeo todo o processo de construção da solução. Trata-se de um recurso poderoso que pode contribuir para diversas análises sobre o modo como cada aluno constrói a solução.

Este capítulo apresentou os recursos software PROALG que foi desenvolvido durante a pesquisa com o objetivo de suportar o método da problematização, para coletar dados relacionados à resolução de cada atividade e constituir o modelo afetivo do aluno. Por ter sido uma ferramenta importante durante todas as atividades, especialmente nos estudos de caso, a mesma foi apresentada antes da metodologia da pesquisa e do modelo para o reconhecimento da motivação, entretanto, o desenvolvimento e aplicação do sistema é parte importante dos resultados desta tese. No próximo capítulo serão apresentados os pressupostos metodológicos que guiaram a presente pesquisa, as etapas do trabalho, além das ações e resultados preliminares de cada etapa.

4 METODOLOGIA DA PESQUISA

Neste capítulo será apresentado o plano utilizado na condução das pesquisas. Segundo Creswell (2010), um projeto ou plano de pesquisa envolve três componentes, as concepções filosóficas, as estratégias ou a natureza da investigação e os métodos ou procedimentos. A concepção adotada nesta pesquisa se caracteriza como pós-positivista e pragmática. Segundo Creswell (2010) a concepção pragmática concede ao pesquisador uma liberdade de escolha dos métodos que são considerados mais adequados ao problema e o objetivos propostos.

Esta concepção foi adotada em função de haver uma preocupação com a aplicação e quais técnicas funcionam para a resolução do problema da pesquisa que é “*Como utilizar a Aprendizagem Ativa para o aprendizado de algoritmos e programação, considerando a dimensão afetiva?*”, pois busca-se analisar o cenário atual, o comportamento dos participantes, sem uma análise profunda do contexto social no qual estão inseridos.

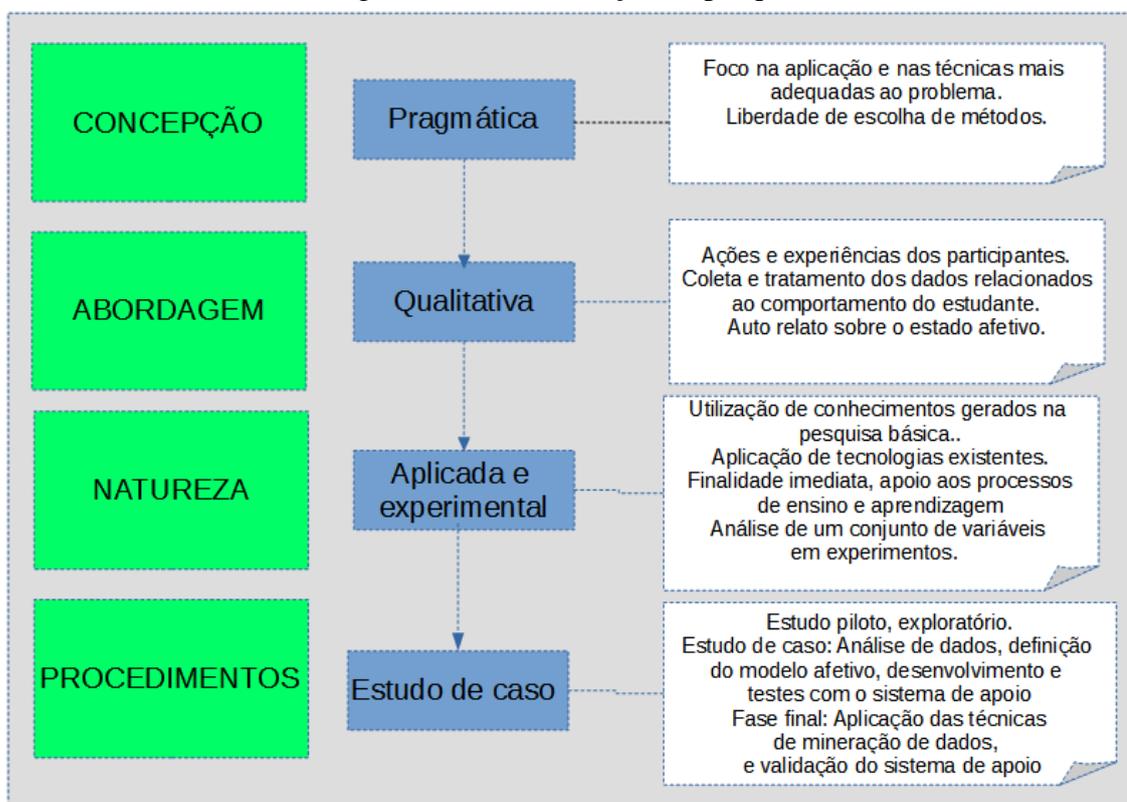
Quanto a natureza da investigação, a pesquisa pode ser definida como aplicada (Figura 37), pois utiliza conhecimentos e tecnologias existentes com o objetivo de propor e aplicar de uma estratégia pedagógica baseada na aprendizagem ativa e um modelo para acompanhar as atividades sob a ótica afetiva. Também é possível definir a pesquisa como experimental, pois a mesma é realizada a partir da definição de um tema ou objeto de estudo e de um conjunto de variáveis que influenciam o estudo (GIL, 2010).

A abordagem da investigação, que pode ser quantitativa, qualitativa ou mista, estabelece diretrizes para os procedimentos que serão utilizados (CRESWELL, 2010). Métodos quantitativos baseiam-se em dados mais estruturados, permitindo análises de variáveis e suas relações. Flick (2009) destaca que a metodologia qualitativa possibilita a interpretação das ações e experiências dos participantes. Estratégias mistas combinam ou associam as abordagens, utilizando procedimentos qualitativos e quantitativos, de forma complementar. A abordagem de investigação adotada neste trabalho foi mista, com predominância qualitativa.

Os métodos ou procedimentos se constituem no terceiro componente da pesquisa, segundo Creswell (2010). Os métodos utilizados nesta pesquisa incluíram um estudo piloto, cujo objetivo foi avaliar a viabilidade do uso do método pedagógico e do reconhecimento da motivação, e um estudo de caso, composto por duas fases, nas quais foram coletados dados sobre as ações dos estudantes. Questionários foram utilizados para conhecer a percepção e o estado afetivo do estudante durante a realização das atividades. Yin (2010) aponta o estudo de caso como método adequado para situações nas quais a questão da pesquisa está relacionada ao

“Como” e “Por quê”. Estudos de caso podem ser usados também em eventos contemporâneos e em contextos nos quais não é necessário um controle sobre os eventos comportamentais.

Figura 37 – Classificação da pesquisa



Fonte: Elaborada pelo autor

Métodos qualitativos e quantitativos foram empregados para a análise dos dados coletados no estudo de caso. Medidas estatísticas quantitativas como média, desvio padrão, quartis foram adotadas em uma análise inicial dos dados e na etapa de pré-processamento para a aplicação das técnicas de mineração de dados. A categorização dos atributos, com destaque para as variáveis relacionadas aos tempos e a classificação prévia do esforço e confiança foram baseadas também nos estudos feitos e na observação do pesquisador sobre os experimentos realizados.

A avaliação dos algoritmos para classificação supervisionada se utilizou de uma combinação de métricas quantitativas, como a acuracidade, precisão e a matriz de confusão, combinadas como uma análise qualitativa sobre a capacidade destes algoritmos de determinar corretamente os níveis de esforço e confiança. A avaliação dos métodos de classificação supervisionada, especialmente as redes bayesianas foi fundamental para a elaboração do modelo.

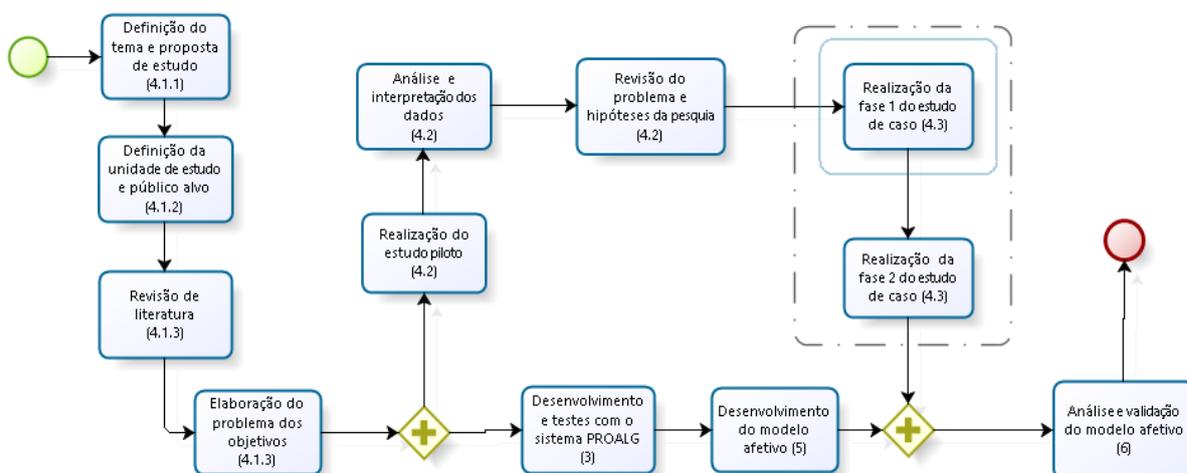
A análise dos questionários utilizados como ferramenta para um autorrelato do estudante sobre o seu estado afetivo e a sua percepção sobre as atividades desenvolvidas foi baseada em uma abordagem mista. Foram utilizadas medidas estatísticas básicas, como a média, desvio padrão e o coeficiente de variação, além de gráficos que indicam a proporção e o percentual das respostas em cada alternativa. Os estudantes foram segmentados de acordo com a classificação do modelo afetivo e as respostas de cada grupo foram comparadas com o intuito de verificar o padrão nas respostas para cada segmento. Técnicas quantitativas foram adotadas de forma complementar para avaliar as respostas de cada grupo de estudantes para as questões propostas na Roda dos Estados Afetivos (REA).

A seguir serão descritas as etapas da pesquisa, com ênfase no estudo de caso, nos objetivos, dados coletados e recursos utilizados. A análise inicial das informações obtidas no estudo piloto e no estudo de caso serão apresentados respectivamente nas seções 4.3 e 4.4

4.1 Etapas da pesquisa

Nesta seção serão apresentadas as atividades realizadas em cada etapa, os objetivos atingidos, os resultados iniciais e um resumo dos dados coletados. A elaboração do modelo ECI (Esforço, confiança e independência) adaptado para o reconhecimento da motivação no contexto deste trabalho e a análise dos resultados da aplicação e validação deste serão apresentados em capítulos posteriores. O fluxo mostrado na figura 38 apresenta as etapas da pesquisa e a sequência em que estas ocorreram. Nas imagem foram inseridos os números das seções que descrevem as atividades executadas.

Figura 38 – Planejamento das etapas da pesquisa



Fonte: Elaborado pelo autor

O estudo piloto e o estudo de caso ocorreram paralelamente ao desenvolvimento do sistema PROALG e a elaboração do modelo para reconhecimento da motivação. O software

desenvolvido passou a ser utilizado já na fase 1 do estudo de caso, tendo sido ajustado e aperfeiçoado durante esta etapa. Dados coletados no estudo piloto e estudo de caso e resultados preliminares foram fundamentais na construção do modelo afetivo.

4.1.1 Definição do tema e proposta de estudo

A definição do tema da pesquisa e a proposta de estudo foi importante para definir as primeiras hipóteses e para a realização do estudo piloto. A premissa inicial foi que a pesquisa abrangeeria o ensino de programação, com ênfase nos componentes curriculares introdutórios.

O tema e a questão de pesquisa estão alicerçados na utilização de um método de aprendizagem ativa, que se constituiu na prática pedagógica. O reconhecimento da dimensão afetiva teve como objetivo contribuir para um acompanhamento e tomada de ações que possam melhorar a aprendizagem. O objetivo é apoiar o professor no desenvolvimento de ações pedagógicas junto aos estudantes.

4.1.2 Definição da unidade de estudo e público-alvo

Para o estudo piloto e estudo de caso foi escolhida uma turma de estudantes da disciplina de Algoritmos e Programação, ministrada nos cursos da área de computação, na Universidade do Vale do Taquari – UNIVATES.

A Univates é uma instituição de ensino de caráter comunitário, localizada no município de Lajeado, no Vale do Taquari, estado do Rio Grande do Sul. A maioria dos estudantes é oriunda de municípios próximos, localizados nos Vales do Taquari e Rio Pardo. São oferecidos 4 cursos na área de computação, Sistemas de Informação, Redes de Computadores, Engenharia de Computação e Engenharia de Software. A disciplina citada consta na matriz dos referidos cursos, no primeiro semestre, sendo desta forma a primeira experiência com a programação no curso.

O número máximo de estudantes matriculados é definido em função da capacidade dos laboratórios, considerando a necessidade de um equipamento por estudante, ou seja, cada aluno tem acesso individual a um computador. Desta forma, a maioria das turmas têm entre 15 e 30 alunos matriculados, dependendo da demanda em cada semestre. Em todos os casos o ensino ocorre na modalidade presencial, com aulas realizadas duas vezes por semana, considerando a carga horária da disciplina que é de 120 horas.

O estudo piloto, descrito na seção 4.2, demonstrou a viabilidade de utilizar turmas de Algoritmos e Programação para aplicação da pesquisa, o que permitiu definir esta disciplina como unidade de pesquisa tanto no estudo de caso, quanto na validação do modelo afetivo. A

disponibilidade de equipamentos (um por estudante) também é um fator importante para a escolha destas disciplinas, pois, permite o desenvolvimento, tanto de atividades individuais, quanto em grupo, onde cada estudante pode expor a sua visão sobre o problema, discutindo com os colegas, posteriormente.

A tabela 6 apresenta um resumo dos participantes e das atividades resolvidas em cada uma das etapas. É possível verificar que o número de estudantes foi menor no estudo piloto e a quantidade de atividades propostas e entregues foi maior na primeira fase do estudo de caso.

Tabela 6 – Participantes por etapa

Etapa	Número de estudantes (participantes)	Número de atividades publicadas	Número de exercícios entregues pelos estudantes
Estudo piloto	14	8	84
Estudo de caso: fase 1	28	11	278
Estudo de caso: fase 2	26	7	147

Fonte: Elaborado pelo autor

4.1.3 Revisão de literatura

A pesquisa bibliográfica foi fundamental para a elaboração da proposta apresentada em 2017 e na condução das pesquisas. Inicialmente foram buscados materiais e publicações que permitissem conhecer melhor o cenário da pesquisa no ensino de algoritmos e programação, uma vez que uma das premissas era qualificar o ensino em uma disciplina introdutória de programação.

Foram pesquisados métodos de Aprendizagem Ativa e resultados da aplicação destes na programação ou em áreas afins. As principais abordagens analisadas foram PBL e a problematização. Estes levantamentos mostraram que a problematização poderia ser uma alternativa promissora para a proposição de uma estratégia pedagógica que focasse na compreensão e definição de um problema antes da escrita do código fonte.

Durante a pesquisa sobre estados afetivos e o reconhecimento destes, foi possível encontrar vários trabalhos que se tornaram referências para a elaboração desta proposta, estes foram detalhados no capítulo 2. Observou-se que as técnicas computacionais mais utilizadas para a identificação de estados afetivos estão relacionadas à mineração de dados educacionais, sendo assim, as pesquisas sobre este tema foram aprofundadas, com o estudo de técnicas que seriam aplicadas nas etapas seguintes.

Os resultados da revisão de literatura permitiram a elaboração da questão da pesquisa e dos objetivos do trabalho. Além disso foram definidas as bases e principais hipóteses a serem

investigadas no estudo piloto que seria realizado na sequência e que, juntamente com novas pesquisas permitiu a atualização dos pressupostos teóricos, face a atualização das pesquisas na área.

4.2 Estudo piloto

A etapa da pesquisa denominada estudo piloto, foi realizada no segundo semestre de 2016, durante os meses de outubro e novembro do referido ano. Esta seção apresenta os principais resultados desta etapa que teve como participantes os alunos de uma disciplina de Algoritmos e Programação, no qual investigou-se a viabilidade de utilizar uma estratégia de ensino baseada na problematização para a resolução das problemas, além de avaliar a possibilidade de identificar a motivação dos estudantes. O estudo piloto e os seus resultados serviram de base para definir os requisitos e implementar o sistema de apoio que foi utilizado no estudo de caso e na validação do modelo, além de contribuir para o planejamento das atividades e da metodologia apresentadas nesta tese.

A disciplina na qual foi realizado o estudo piloto ocorreu na modalidade presencial e o número total de participantes foi 14. Cabe salientar que o número total de matriculados era de 17, mas durante as atividades 3 estudantes desistiram e, desta forma, os dados dos mesmos foram descartados. Foram publicados 8 exercícios, sendo que 84 foram entregues, ou seja, nem todos os estudantes finalizaram todas as atividades, com uma média de 10 entregas por exercício. Considerando o número de entregas de atividades por estudante, em média foram 6 exercícios concluídos, embora alguns estudantes tenham concluído todas as atividades, enquanto outros tenham finalizado metade delas.

Todas as atividades ocorreram em um laboratório de informática, com a alocação de um computador por estudante, fator que facilitou uma orientação e acompanhamento mais direto e próximo por parte do professor. O cenário exposto contribuiu para a pesquisa sobre métodos que estimulassem a reflexão sobre problemas e que demandassem uma análise detalhada do comportamento dos estudantes durante o desenvolvimento das tarefas.

A estratégia adotada seguiu o fluxo mostrado na figura 21 (seção 2.6.1), onde o professor foi responsável por descrever um problema e os estudantes deveriam por descrever os pontos-chave, indicar uma hipótese de solução e, por fim, elaborar a solução sob a forma de um programa de computador. O AVEA Moodle foi usado como plataforma de apoio e coleta de dados. O recurso denominado Lição foi usado para implementar atividades compostas pelas seguintes etapas:

- Definição dos pontos-chave para o problema: inicialmente o estudante foi estimulado a compreender e pensar nos recursos e elementos que usaria para solucionar o problema.
- Definição da hipótese de solução: formalizar uma sequência de passos, blocos ou mesmo um fluxograma indicando uma solução em alto nível antes de iniciar a escrita do código fonte.
- Enunciado e dicas: nesta etapa o estudante visualizava o enunciado detalhado e poderia obter mais dicas a partir dos links para textos de ajuda. O auxílio mais direto é o próprio pseudocódigo, o algoritmo da solução.
- Escrita e postagem do código fonte: finalizando a atividade, o aluno deveria elaborar um programa na linguagem Java e postar, finalizando a lição.

Inicialmente foram propostas questões simples, relacionadas às entradas, saídas e atribuições, culminando com atividades que requeriam o uso de estruturas mais complexas como matrizes. Inicialmente foi apresentada ao estudante uma contextualização do problema a ser resolvido. Além de receber a descrição de uma situação-problema, os estudantes foram estimulados a teorizar e embasar a sua compreensão do problema utilizando os materiais disponibilizados pelo professor ou qualquer outra fonte de pesquisa.

Figura 39 – Definição inicial do problema

Considere um jogo no qual dois dados são jogados ao mesmo tempo diversas vezes. Qual a soma dos valores dos dados?
 Se jogarmos os dados várias vezes, por exemplo 100 vezes, quantas vezes a soma será 2, 3, 4, 5 e assim por diante?
 Vamos criar um programa que jogue os dados um determinado número de vezes, quantas o usuário desejar e que após as jogadas mostre quantas vezes ocorreu cada um dos valores possíveis para a soma dos dados.
 Descreva abaixo os elementos, recursos ou conceitos fundamentais para resolução deste problema.

A sua resposta

Parágrafo B I U

Fonte: Elaborado pelo autor

Diferentemente dos enunciados clássicos que indicam ao aluno ações, como “ler duas variáveis inteiras”, “calcular a divisão de x e y ”, “ler uma matriz de 5×5 ”, os textos iniciais tiveram como proposta contextualizar o problema, associá-lo a uma situação real e estimular o estudante a buscar as informações necessárias para a construção da solução algorítmica. Uma análise das descrições do problema e enunciado foi apresentada na seção 2.6.2, entretanto, cabe ressaltar que as premissas para definição das mesmas foi sendo construída ao longo do estudo piloto e do estudo de caso. Desta forma, as definições mostradas nas figuras 39 e 40 refletiam

as concepções iniciais da pesquisa, por isso são diferentes dos textos mostrados na tabela 5 (seção 2.6.2).

A segunda etapa solicita que o aluno represente o problema em uma solução de alto nível, que pode ser uma sequência de passos ou blocos que compõe a solução. O objetivo é que o estudante formule mentalmente uma representação que posteriormente servirá de base para construção do código fonte. A seguir o enunciado tradicional, que contém indicativos de como resolver o problema, é exibido para o aluno. Nesta página foram disponibilizados botões para acesso às dicas e ao pseudocódigo e outro que leva à resolução da atividade, à postagem do código fonte da solução (Figura 40).

Figura 40 – Enunciado tradicional e dicas

Dicas

Para que o programa funcione adequadamente o usuário deverá informar a quantidade de jogadas que ele deseja. O programa deverá simular, para cada jogada, o lançamento dos dois dados e somar os valores sorteados. O resultado da soma deverá ser contabilizado na posição correspondente do vetor, que possui uma posição para cada resultado possível. Antes de terminar, o vetor deverá ser impresso, informando o resultado das jogadas.

Abaixo você irá encontrar links para dicas que podem auxiliar no desenvolvimento do código fonte. É possível visualizar as dicas, além do pseudocódigo relacionado ao exemplo.

Desenvolva o código fonte para a solução e, após finalizá-lo clique no link "Postar código fonte" para postar o mesmo, encerrando a atividade.



Fonte: Elaborado pelo autor

Neste momento o aluno poderia optar por ir para a etapa final, que é a escrita e postagem do código ou visualizar dicas de auxílio que poderiam contribuir para o desenvolvimento do código fonte. Todas as ações do estudante ficaram registradas no log do ambiente, incluindo início e fim da lição, possíveis paradas e retomadas, instantes em que as questões são respondidas, entre outras informações.

Após o período de realização das atividades, foi aplicado questionário que teve como objetivo coletar dados da percepção que os alunos tiveram sobre as atividades. A análise dos resultados da autoavaliação será apresentada no capítulo 6, em conjunto com os dados das respostas dos participantes do estudo de caso.

4.2.1 Análise dos dados coletados no estudo piloto

Inicialmente os registros obtidos no log do AVEA Moodle foram analisados com objetivo de verificar a relevância de cada ação e quais informações poderiam ser obtidas. A data e a hora de cada uma das ações, bem como as páginas e recursos acessados, constituíram-se em dados essenciais para definição das variáveis propostas. Exemplos destes registros podem ser visualizados na tabela 7.

Nesta tabela percebe-se o instante em que determinado evento ocorre, a descrição do evento e os detalhes que permitem verificar exatamente o que o estudante fez. Os identificadores das páginas ou módulos foram utilizados para calcular o tempo decorrido entre uma resposta e outra e para constatar se o aluno acessou determinado conteúdo, relacionado às dicas.

Tabela 7 – Registros de log coletados do sistema Moodle

Data/hora	Ação	Descrição
10/11/2016 19:20	Lição iniciada	The user with id '771' started the lesson with course module id '558104'.
10/11/2016 19:20	Pergunta visualizada	The user with id '771' has viewed the Dissertação question with id '7188' in the lesson activity with course module id '558104'.
10/11/2016 19:24	Pergunta respondida	The user with id '771' has answered the Dissertação question with id '7188' in the lesson activity with course module id '558104'.
10/11/2016 19:26	Página de conteúdo visualizada	The user with id '771' has viewed the content page with id '7190' in the lesson activity with course module id '558104'.

Fonte: Elaborado pelo autor

Levando em consideração os pressupostos encontrados em (DU BOULAY E DEL SOLDATO, 2016; BERCHT, 2001; LONGHI, 2011), foram propostas variáveis relacionadas ao esforço (tempos, nível de detalhe), confiança e independência (acesso às dicas, retomadas e o acesso anterior à realização da tarefa). Os atributos considerados no estudo piloto foram:

- Tempo para descrição dos pontos-chave do problema: tempo calculado desde o momento em que o aluno iniciou a lição até a submissão da resposta para a primeira questão.
- Tempo para descrição da proposta ou hipótese de solução: tempo decorrido entre a submissão da primeira e a segunda resposta.
- Tempo para a escrita do código fonte: após responder a segunda questão, o aluno tem acesso às dicas e a um enunciado detalhado. Todo o tempo gasto com a leitura do enunciado, dicas e para a escrita e postagem do código fonte é considerado nesta etapa.
- Tempo total da atividade: totalização dos tempos anteriores.
- Acesso às dicas: indica se o estudante olhou o texto das duas dicas e do pseudocódigo. Neste caso não foi computado o tempo que o estudante demandou à leitura das dicas, apenas se acessou ou não.
- Número de retomadas: quantas vezes o aluno abandonou e retomou a atividade. Este dado é importante para verificar alunos que tem dificuldade para manter o foco na resolução do problema. O tempo que o aluno permanece fora do sistema não é considerado para cálculo das variáveis anteriores.

- Acesso anterior: indica se o aluno acessou em determinada data e retornou posteriormente, em outro momento para resolver as questões. Esta informação complementa a anterior, indicando casos em que o retorno a uma atividade ocorre somente no dia seguinte ou alguns dias depois.

Após uma análise dos conteúdos das respostas dos estudantes para as duas primeiras questões, foi definido um conjunto de variáveis que representam a forma que o estudante define os elementos do problema. As variáveis abaixo estão relacionadas à descrição dos pontos-chave do problema.

- Indicação do uso de uma sequência de passos: alguns estudantes podem apontar como pontos-chave uma sequência de passos, uma estrutura lógica ou ações, mesmo que o objetivo na primeira etapa seja somente compreender as características principais do problema.
- Tipo de representação: se foram descritos os dados, operações ou se foram utilizadas instruções de programação.
- Nível de detalhamento: resultado da análise da resposta do aluno, considerando a quantidade de pontos-chave indicados e um maior ou menor detalhamento para cada elemento do problema. A classificação usada foi alto, baixo ou médio.

Foram definidas também variáveis relacionadas à descrição da hipótese.

- Foi utilizado código fonte na hipótese?
- Nível de detalhamento: a análise da solução proposta considera o número de passos indicados e a capacidade do aluno de especificar mais detalhadamente cada um dos passos. Também classificada como alto, médio ou baixo.
- Foi usada uma abordagem sequencial, conjunto de passos ordenado?

O nível de detalhamento nas duas questões é importante para indicar um maior ou menor grau de esforço na resolução da atividade. No caso da identificação dos pontos-chaves do problema, em alguns casos verificou-se respostas que incluíam todas ou maioria das variáveis, com os tipos correspondentes e outras respostas que apenas citavam como ponto chave “variáveis”.

Na definição de uma hipótese ou especificação da solução, um maior detalhamento pode permitir que outra pessoa produza a solução, além de demonstrar a capacidade do estudante para analisar o problema e estruturar uma solução antes de iniciar a escrita do código fonte. Na tabela 8 são apresentados exemplos de descrições dos pontos-chave e da hipótese de solução com diferentes níveis de detalhamento.

Na primeira coluna as respostas são extremamente reduzidas, o que pode ser um indício de pouca dedicação ou baixa capacidade de compreensão do problema. Embora as respostas mostradas na segunda coluna apresentam eventuais inconsistências, demonstram uma tentativa de formalizar mais claramente o problema antes de elaborar o código fonte.

Tabela 8 – Respostas com alto e baixo nível de detalhamento

Questão	Baixo detalhamento	Alto detalhamento
Pontos-chave	-Variáveis -entradas -matriz	criar variáveis uma matriz uma entrada para a matriz, para saber o número de linhas uma entrada para armazenar o tipo laço condicional laço de repetição uma entrada para pedir o tipo armazenado imprimir o pedido
Hipótese	matriz que quando o usuário indicar tais posições salvar a entrada	Gerar uma matriz equivalente a prateleira 12x3 gerar as variáveis lin col fazer dois laços de repetição para preencher a matriz apos preenchida a matriz fazer testes com if para analisar quantos produtos de cada tipo existem em cada prateleira fechar laços pedir ao usuario qual prateleira ele deseja dois laços de repetição informar os valores de tal prateleira fechar laços

Fonte: Elaborado pelo autor

A utilização de instruções de programação que compõe o código fonte na proposta de solução pode indicar que os alunos têm dificuldade em pensar na solução como uma especificação independente de linguagem de programação ou que o raciocínio está diretamente ligado à construção de programas. Ao longo do estudo de caso e da elaboração do modelo para o reconhecimento da motivação algumas variáveis foram descartadas por não estarem diretamente associadas à confiança e ao esforço, como no caso do uso de uma sequência de passos e a forma de representação da solução. Embora se constituíssem em informações interessantes sobre a resolução do problema, foi necessário reduzir o conjunto de atributos do modelo, com o objetivo de torná-lo menos complexo.

A seguir serão apresentadas estatísticas relacionadas aos dados coletados. Foram obtidos registros de 84 atividades, realizadas por 14 alunos diferentes. A tabela 9 apresenta informações agrupados por estudante. Os nomes dos estudantes foram omitidos e definidos como um identificador sequencial.

Tabela 9 – Tempos de resolução das tarefas agrupados por aluno

Aluno	Tempo médio e desvio padrão (minutos)						Proporção de tempo médio	
	Pontos-chave		Hipótese		Código fonte		Pontos-chave + Hipótese	Código fonte
1	10,00	8,88	3,50	2,69	27,83	30,52	32,27%	67,73%
2	7,00	5,45	6,50	6,42	23,75	14,62	36,24%	63,76%
3	17,25	11,11	4,75	6,60	21,12	24,04	51,01%	48,99%
4	16,37	14,29	8,00	4,72	13,25	15,51	64,78%	35,22%
5	30,62	30,66	24,12	25,50	28,00	23,92	66,16%	33,84%
6	6,12	4,05	4,00	3,50	23,87	16,40	29,78%	70,22%
7	13,62	6,09	10,75	6,51	18,62	18,07	56,69%	43,31%
8	18,75	19,46	3,50	2,67	24,00	18,96	49,73%	51,89%
9	9,50	9,98	5,87	7,58	26,25	20,23	36,94%	63,06%
10	17,12	24,19	3,50	3,81	23,50	19,06	47,19%	52,81%
11	30,00	28,01	11,75	13,25	12,25	15,34	76,96%	23,04%
12	9,62	10,54	4,12	4,54	25,00	30,56	35,48%	64,52%
13	19,57	17,43	9,71	6,75	45,42	46,81	39,20%	60,80%
14	8,62	3,99	9,75	6,29	29,37	24,34	35,00%	65,00%

Fonte: Elaborado pelo autor

A média e o desvio padrão para cada estudante foi calculada tendo como base todas as atividades realizadas ao longo do experimento. Observa-se uma grande variação no tempo médio dedicado a cada uma das tarefas, enquanto alguns apresentam médias altas, entre 20 e 30 minutos para a definição dos pontos-chave, outros utilizam menos da metade do tempo para resolver a questão.

Descrever uma hipótese de solução normalmente exige uma compreensão mais profunda do problema e uma reflexão (BERBEL e GAMBOA, 2011), entretanto, o tempo dedicado à primeira etapa é quase sempre superior à segunda e em alguns casos é mais do que o dobro. Observou-se que muitas vezes a estratégia adotada foi identificar os pontos-chave e no mesmo momento estruturar um esboço da solução e só depois postar os conteúdos para as duas respostas no ambiente. Uma hipótese inicial é que valores muito reduzidos poderiam indicar pouco esforço ou que o aluno não considerava importante compreender e formalizar o problema antes de elaborar a solução. Estabeleceu-se desta forma uma analogia dos tempos identificados com o número de tentativas e medidas de perseverança encontradas em Del Soldato e Du Boulay (1995).

Os valores do desvio padrão mostrado indicaram discrepâncias em relação à média. Observou-se uma grande variação entre o tempo dedicado em um exercício e em outro, na mesma etapa. Pode-se citar o exemplo de estudantes que dedicaram 2 minutos para a elaboração do código fonte em determinada atividade e em outras, tempos superiores a uma hora. Tempos muito reduzidos em algumas etapas, principalmente, na escrita do código fonte, geraram questionamentos sobre a possibilidade do aluno ter copiado de forma completa ou parcial uma

solução de alguma fonte ou de um colega. Este questionamento decorre também do fato de que o código não é produzido de forma integrada no ambiente, sendo apenas postado pelo estudante quando finalizado.

Na maioria das atividades foi despendido um tempo maior para a escrita do código fonte. Entretanto, em alguns casos a soma do tempo das duas primeiras atividades representa mais de 60% do tempo total e em um dos casos, é superior a 70%. Analisando os dados da tabela 10, observa-se que estudantes que apresentavam tempos muito maiores nas duas primeiras etapas costumavam acessar frequentemente as dicas ou paravam e retomavam a atividade mais vezes. Uma hipótese possível é que este comportamento estivesse associado à baixos níveis de confiança e independência, além de indicar uma certa dificuldade na compreensão do problema.

A predominância por níveis baixos ou médios no detalhamento do problema foi visível (Tabela 10). Não foi possível estabelecer uma relação direta entre o tempo dedicado e o nível de detalhamento das questões iniciais. Tempos médios altos nas primeiras etapas, como nos exemplos dos alunos 4 e 5, não resultaram necessariamente em maior detalhamento nas respostas. Ao analisar os conteúdos das respostas dos estudantes foi possível constatar que as mesmas apresentam muitas semelhanças nas duas primeiras etapas e, em alguns casos, são praticamente iguais. Tal fato também contribuiu para uma redução dos tempos destinados a descrição da proposta de solução.

Tabela 10 - Estatísticas sobre acessos e detalhamento por aluno

Aluno	Número de acesso às dicas	Número de retomadas	Acesso anterior	Nível de detalhamento					
				Pontos-chave			Hipótese		
				Alto	Médio	Baixo	Alto	Médio	Baixo
1	4	2	1	0	0	5	0	3	2
2	0	4	2	0	2	4	3	2	1
3	0	5	3	1	3	4	1	5	2
4	11	17	5	0	2	5	1	2	4
5	5	21	6	2	2	3	3	3	1
6	3	6	2	0	5	2	0	5	2
7	7	17	5	1	0	5	1	4	1
8	9	9	1	1	5	0	1	4	1
9	0	3	0	3	3	0	0	3	3
10	6	7	2	2	3	1	0	2	4
11	8	4	0	0	4	0	0	4	0
12	13	6	2	1	2	4	0	1	6
13	7	7	3	2	0	3	1	4	0
14	1	4	1	0	1	3	0	1	3

Fonte: Elaborado pelo autor

O número de vezes que o aluno interrompe e atividade e retoma posteriormente combinado com os acessos realizados em uma data anterior podem ser usados como indicativos

para a confiança e independência. É possível constatar que estudantes que acessam constantemente as dicas também apresentam vários acessos anteriores, ao contrário dos estudantes que pouco utilizam as dicas.

A tabela 11 apresenta um resumo sobre os tempos das atividades, que foram classificadas de acordo com o nível de dificuldade (1,2 ou 3). O nível 1 está relacionado ao uso somente de entradas, atribuições, expressões de cálculo e saídas, no segundo nível são empregadas estruturas de repetição e vetores. O nível mais alto inclui a manipulação de matrizes e, eventualmente procedimentos ou funções. As duas primeiras atividades apresentaram um nível de dificuldade baixo, algoritmos que exigiam apenas a leitura de dados, cálculos e saídas. Entretanto, observa-se tempos altos e variáveis para a definição dos pontos-chave e para a hipótese. Durante a resolução das mesmas, percebeu-se que os estudantes demoraram a compreender o que era solicitado e somente após compreender o problema e as questões passaram para o desenvolvimento da solução.

Tabela 11 - Tempos de resolução das tarefas agrupados por atividade

Ativ.	Tempo médio (minutos)			Proporção de tempo médio	
	Pontos-chave	Hipótese	Código fonte	Pontos + Hipótese	Código fonte
1 (1)	16,35	20,50	27,85	56,95%	43,05%
2 (1)	23,53	10,23	23,76	58,69%	41,31%
3 (2)	12,53	4,61	14,84	53,61%	46,39%
4 (2)	11,00	6,58	12,91	57,65%	42,35%
5 (2)	13,41	3,00	6,91	70,36%	29,64%
6 (3)	16,50	5,50	27,64	44,32%	55,68%
7 (3)	20,35	4,57	35,28	41,40%	58,60%
8 (3)	8,69	6,46	49,30	23,51%	76,49%

Fonte: Elaborado pelo autor

Nas atividades seguintes houve uma queda geral no tempo médio, principalmente da segunda etapa, a hipótese da solução. A atividade final mereceu uma análise mais profunda, uma vez que os tempos destinados à análise do problema reduziram, enquanto houve um aumento significativo da codificação. Nesta atividade foi apresentada uma definição bem abrangente do problema, com menos detalhes, se comparada às demais. O objetivo foi verificar se os alunos seriam capazes de pesquisar, buscar subsídios e propor uma solução própria antes de passar para a construção do programa. Os tempos e os baixos níveis de detalhamento observados na tabela 12 indicam uma dificuldade de propor uma hipótese de solução e pouca dedicação para compreender e formalizar o problema.

Tabela 12 - Estatísticas sobre acessos e detalhamento por atividade

Atividade	Número de acesso às dicas	Número de retomadas	Acesso anterior	Nível de detalhamento					
				Pontos-chave			Hipótese		
				Alto	Médio	Baixo	Alto	Médio	Baixo
1	10	17	1	2	3	7	2	9	1
2	3	11	1	3	5	4	1	3	7
3	4	8	2	3	4	4	2	5	5
4	1	14	7	1	3	6	2	5	4
5	1	9	4	0	2	6	0	5	3
6	11	23	7	3	5	3	1	6	4
7	21	23	9	1	4	4	2	2	5
8	23	7	1	0	5	6	1	2	8

Fonte: Elaborado pelo autor

A tabela 12 indica que uma maior complexidade leva a um aumento no acesso às dicas e no número de retomadas. A exceção fica por conta da primeira atividade que apresentou altos índices em virtude de constituir-se na primeira experiência com a estratégia usada.

É possível constatar que o detalhamento na primeira questão é predominantemente baixo ou médio. Na maioria dos casos os estudantes responderam com algumas instruções ou variáveis básicas, sem um detalhamento maior. No caso da proposta de solução houve uma variação um pouco maior, entretanto, o nível médio de detalhamento apresenta contagem igual ou superior à metade em todos os exercícios. Observando-se os conteúdos das respostas, percebe-se que são indicadas etapas para a solução, de forma sequencial, mas na maioria dos casos em alto nível, e em diversas situações alguns passos são negligenciados, aparecendo somente no código fonte.

Com base nas análises dos dados do estudo piloto foram definidas as variáveis que deveriam ser coletadas no estudo de caso, considerando as hipóteses formuladas e as necessidades associadas ao reconhecimento da motivação. Após o estudo piloto foi possível perceber a importância de uma avaliação da compreensão demonstrada pelo estudante, uma avaliação qualitativa, feita pelo professor. Outra questão levantada foi a possibilidade de introduzir uma avaliação do aluno sobre a complexidade do problema, questionamento que foi adicionado no fim da atividade. As variáveis que foram mantidas para a etapa seguinte, foram os tempos (pontos-chave do problema, hipótese e código), os níveis de detalhamento (pontos-chave e hipótese), os registros de acesso aos recursos de ajuda e a quantidade de retomadas.

4.3 Estudo de caso

O estudo de caso foi dividido em duas etapas, denominadas fase 1 e fase 2, pelo fato das atividades terem sido desenvolvidas em duas turmas diferentes, a primeira no semestre A de

2017 e a segunda no semestre B deste mesmo ano. Entretanto, é possível considerar que as duas etapas fazem parte de um estudo caso único em função de terem os mesmos objetivos e de utilizarem o mesmo método para desenvolvimento dos exercícios. O estudo de caso teve como propósito inicial dar continuidade ao uso do método da problematização utilizado no estudo piloto, além da coleta de dados que tiveram como base as mesmas variáveis do estudo piloto.

Na fase 1 foram aplicados 11 exercícios de programação, número superior a quantidade de atividades do estudo piloto. Participaram desta etapa, 28 estudantes, sendo que o número total de exercícios resolvidos foi 278, uma média aproximada de 10 atividades resolvidas por estudante e de 25 resoluções em média por atividade publicada. Ao comparar com os dados das entregas do estudo piloto, constata-se que o percentual de resoluções sobre o total de atividades publicadas foi de aproximadamente 90%, enquanto no estudo piloto ficou em 75%.

O ambiente Moodle ainda foi utilizado na fase 1 do estudo de caso, especialmente nas primeiras atividades, em função do sistema PROALG se encontrar em fase de testes. A partir da metade do semestre as atividades passaram a ser publicadas prioritariamente no sistema de apoio à problematização, embora duas tarefas ainda tenham sido resolvidas no Moodle. O uso paralelo das duas ferramentas foi importante para que fosse possível conduzir os testes e ajustes no PROALG.

A fase 2 contou com a participação de 26 estudantes, com um número de 7 exercícios publicados pelo professor. Em média foram entregues 5,60 atividades por aluno e 21 entregas por atividade. O percentual de exercícios resolvidos em relação ao total de tarefas atribuídas foi de aproximadamente 80%, superior ao estudo piloto, mas inferior à fase anterior do estudo de caso. O número menor de exercícios em relação a primeira parte do estudo de caso se deveu ao uso exclusivo do software PROALG nesta etapa e a um período inicial de validação da ferramenta, durante o qual não foram coletados dados das atividades.

Diferentemente do estudo piloto, que concentrou os exercícios em um período menor, o estudo de caso consistiu em uma utilização sistemática da problematização como estratégia ao longo de todo o semestre, o que permitiu obter resultados mais consistentes para atingir os objetivos relacionados à proposição da metodologia e à definição dos observáveis relacionados a motivação.

4.3.1 Análise dos dados coletados no estudo de caso

Com o intuito de demonstrar a distribuição dos dados coletados em relação aos estudantes e atividades serão apresentadas na sequência as tabelas com os dados da fase 1 do estudo piloto. Serão apresentados as mesmas variáveis mostrados no estudo piloto, o que

permite estabelecer uma comparação entre as etapas, além de avaliar a evolução do método proposto e a inserção do sistema PROALG na pesquisa.

Tabela 13 – Tempos de resolução das tarefas por aluno na fase 1 do estudo de caso

Aluno	Tempo médio e desvio padrão (minutos)						Proporção de tempo médio	
	Pontos-chave		Hipótese		Código fonte		Pontos-chave Hipótese	+Código fonte
1	12,64	6,68	9,36	9,19	17,36	19,69	55,89%	44,11%
2	29,73	22,35	7,82	10,02	22,64	26,00	62,39%	37,61%
3	24,55	23,61	21,00	19,67	17,09	15,09	72,71%	27,29%
4	15,73	18,45	12,64	12,82	28,27	22,06	50,08%	49,92%
5	24,50	17,84	4,90	3,48	22,10	29,63	57,09%	42,91%
6	26,22	21,80	12,44	11,54	21,88	24,61	66,54%	33,46%
7	38,55	24,27	17,55	16,08	4,36	10,83	92,78%	7,22%
8	18,20	14,86	4,50	2,07	27,20	16,71	45,49%	54,51%
9	22,71	25,77	4,57	3,21	11,00	12,23	71,27%	28,73%
10	38,00	34,29	5,00	4,53	5,00	2,71	97,73%	2,27%
11	28,27	26,74	10,00	8,89	21,00	24,71	64,57%	35,43%
12	20,64	21,29	12,00	19,65	23,90	21,05	60,03%	39,97%
13	21,09	13,63	24,55	41,20	34,45	25,09	56,98%	43,02%
14	5,75	5,50	4,50	5,74	43,25	20,39	19,16%	80,84%
15	13,18	5,44	20,00	36,49	38,55	25,83	46,26%	53,74%
16	35,36	22,35	4,45	3,42	11,55	17,64	77,52%	22,48%
17	37,80	22,96	9,00	5,60	25,40	33,79	64,82%	35,18%
18	24,71	19,68	17,43	24,16	13,00	25,75	76,42%	23,58%
19	14,50	5,56	9,20	10,80	24,10	21,18	49,58%	50,42%
20	24,40	21,34	6,80	5,51	13,70	14,41	69,49%	30,51%
21	35,00	21,95	7,90	8,89	14,00	27,49	75,40%	24,60%
22	31,64	41,15	6,09	3,51	9,45	9,56	79,96%	20,04%
23	27,45	27,97	7,45	3,78	25,73	26,87	57,57%	42,43%
24	17,00	23,29	13,10	16,81	28,10	26,89	51,72%	48,28%
25	14,91	10,93	16,82	25,01	42,64	23,79	42,67%	57,33%
26	9,70	9,91	3,40	4,93	18,80	18,83	41,07%	58,93%
27	13,29	8,52	4,29	4,50	29,57	23,24	37,27%	62,73%
28	24,20	17,73	5,70	3,50	44,30	27,48	40,30%	59,70%

Fonte: Elaborado pelo autor

A tabela 13 mostra as médias, desvios dos tempos dedicados à descrição dos pontos-chave, hipótese e elaboração do código por cada estudante. Percebe-se que há uma variação nas médias, especialmente no tempo dedicado aos pontos-chave do problema, com alguns alunos apresentando médias inferiores a 10 minutos, enquanto outros apresentam tempos médios próximos a 40. A ampla variação nos tempos foi uma característica percebida já no estudo piloto e se manteve no estudo de caso.

Como já havia ocorrido no estudo piloto, os tempos dedicados à elaboração da hipótese são em média menores do que ao problema. Durante o estudo piloto foi possível inferir que muitos estudantes descrevem tanto os pontos-chave quanto a hipótese no início do exercício ou respondem de forma muito parecida as duas questões. Esta constatação permanece no estudo

de caso, embora tenha sido possível perceber uma variação maior nos conteúdos postados, e um leve aumento no tempo médio da hipótese.

A proporção de tempo gasto na duas primeiras etapas em relação ao tempo total destinado à tarefa apresentou um padrão semelhante ao verificado no estudo piloto. Grande parte dos estudantes destina um tempo maior nos passos anteriores à escrita do código, entretanto, observa-se que em muitos casos o tempo destinado ao código é muito reduzido, o que indica que o aluno já havia escrito o programa antes de iniciar o passo 3 da tarefa. Chamou a atenção os casos dos estudantes 7 e 10 cuja proporção é superior a 90% , o que confirma a hipótese de que o estudante não segue os passos da forma que é proposto no método.

Tabela 14 – Estatísticas sobre acessos e detalhamento por aluno na fase 1 do estudo de caso

Aluno	Número de acessos às dicas	Número de retomadas	Nível de detalhamento			Hipótese		
			Alto	Médio	Baixo	Alto	Médio	Baixo
1	22	3	1	7	3	2	6	3
2	0	3	2	2	7	2	4	5
3	0	7	1	2	8	4	6	1
4	0	5	0	7	4	3	7	1
5	7	10	0	2	6	1	4	3
6	4	4	0	3	5	2	6	0
7	0	3	2	6	2	3	4	3
8	6	11	2	2	6	1	6	3
9	1	0	2	3	4	2	4	3
10	0	8	2	2	3	2	3	2
11	0	9	2	5	4	1	10	0
12	1	13	1	8	2	2	5	4
13	7	7	2	4	4	2	3	5
14	2	6	2	3	1	1	3	2
15	5	3	2	8	1	2	8	1
16	4	10	3	8	0	2	8	1
17	15	5	2	5	4	1	8	2
18	1	7	2	3	3	1	6	1
19	0	1	1	7	2	1	5	4
20	1	2	1	6	3	1	8	1
21	2	2	1	1	9	2	7	2
22	0	11	1	9	1	1	9	1
23	6	17	0	3	8	0	7	4
24	0	13	1	8	1	1	8	1
25	0	8		5	5	2	6	2
26	0	2	1	7	3	2	8	1
27	6	7	1	4	3	1	6	1
28	7	3	3	4	4	0	8	3

Fonte: Elaborado pelo autor

A tabela 14 apresenta um resumo sobre o número de vezes que alguma dica ou pseudocódigo foi acessado, o número de retomadas, além da distribuição dos níveis de detalhe

entre as classes alto, médio e baixo. Há uma clara tendência de concentração nos níveis de detalhamento médio, tanto no problema, quanto na hipótese, entretanto, comparando os dados com os resultados do estudo piloto, há uma pequena tendência de aumento do nível alto e redução do baixo.

Sobre o acesso as dicas, é possível verificar que há casos de alunos que acessam com frequência, várias vezes, enquanto outros quase nunca ou jamais acessam. De forma semelhante alguns apresentam um grande número de retomadas, enquanto outros tendem a apresentar um comportamento mais linear, iniciando a tarefa e finalizando a mesma sem interromper, na grande maioria das atividades.

Analisando os tempos das atividades é perceptível a influência da complexidade do exercício nos tempos que os estudantes dedicam à resolução, especialmente nas tarefas que envolvem matrizes, vetores e procedures. O incremento no tempo não ocorre apenas na elaboração do código fonte, mas também na compreensão e especificação do problema.

As duas últimas atividades propostas no estudo de caso não foram publicadas no estudo piloto e a complexidade das mesmas foi superior às demais, levando a tempos médios maiores. Os tempos demonstram que proposição de atividades com diferentes níveis de complexidade é importante para o reconhecimento da motivação, uma vez que implicam em tempos e níveis de detalhamento diferentes das mais simplificadas.

Tabela 15 - Tempos de resolução das tarefas por atividade na fase 1 do estudo de caso

Ativ.	Tempo médio (minutos)			Proporção de tempo médio	
	Pontos-chave	Hipótese	Código fonte	Pontos + Hipótese	Código fonte
1 (1)	16,60	9,57	16,64	61,17%	38,83%
2 (1)	25,80	8,56	12,67	73,61%	26,39%
3 (1)	18,15	8,22	34,89	43,05%	56,95%
4 (2)	27,89	24,84	33,32	61,28%	38,72%
5 (2)	21,48	5,04	12,76	67,52%	32,48%
6 (2)	18,08	7,71	33,96	43,17%	56,83%
7 (2)	19,38	10,08	23,00	56,16%	43,84%
8 (3)	21,69	9,19	20,20	62,94%	37,06%
9 (3)	11,25	5,58	6,00	73,72%	26,28%
10 (3)	32,42	11,46	28,46	60,66%	39,34%
11 (3)	65,70	17,78	30,42	67,01%	32,99%

Fonte: Elaborado pelo autor

Um fato que chama a atenção na tabela 15 é que a proporção do tempo dos dois primeiros passos em relação ao tempo para escrita do código não é influenciada diretamente pela complexidade. Embora o tempo para compreensão do problema apresente um incremento significativo nos exercícios mais complexos, a proporção dos tempos é semelhante às demais atividades.

Tabela 16 - Estatísticas sobre acessos e detalhamento por atividade na fase 1 do estudo de caso

Ativ.	Número de acessos às dicas	Número de retomadas	Nível de detalhamento			Hipótese		
			Alto	Médio	Baixo	Alto	Médio	Baixo
1 (1)	4	11	5	17	6	7	19	2
2 (1)	6	12	4	15	8	7	12	8
3 (1)	10	27	3	14	10	3	17	7
4 (2)	14	48	5	11	9	5	13	7
5 (2)	2	12	3	14	9	5	19	2
6 (2)	5	12	3	9	12	2	15	7
7 (2)	10	12	4	12	9	2	19	4
8 (3)	9	1	3	10	9	2	12	8
9 (3)	3	9	2	12	11	2	17	6
10 (3)	17	23	3	11	10	6	14	4
11 (3)	17	19	3	9	13	4	14	7

Fonte: Elaborado pelo autor

A distribuição da complexidade por atividade também concentra-se no no nível médio, na grade maioria das atividades, seguindo um padrão demonstrado na distribuição por estudante e uma tendência verificada no estudo piloto. O número de retomadas e o acesso às dicas apresenta uma tendência clara de alta nas duas atividades finais, que são as mais complexas. Este aumento corrobora com a visão de que exercícios mais complexos são necessários para avaliar a confiança e o esforço do estudante, uma vez que influenciam diretamente na demanda por dicas e pela consulta do pseudocódigo.

As tabelas com os resultados da segunda fase do estudo de caso não serão apresentadas em função de apresentaram dados que refletem os mesmos padrões da primeira parte do estudo de caso. Uma análise dos dados coletados mostrou que os dados seguem as mesmas tendências o que permite concluir que não existem grandes variações no comportamento das diferentes turmas e que o conjunto de dados coletados nas etapas da pesquisa se constitui em uma boa fonte para a aplicação dos métodos de mineração de dados descritos no capítulo 6.

Como o objetivo de complementar as variáveis usadas no modelo afetivo, na segunda fase do estudo de caso foi proposto um novo atributo para avaliar o nível de compreensão do estudante a respeito do problema. Os níveis de detalhamento foram baseados mais no tamanho e nos aspectos quantitativos da resposta, não refletindo uma visão mais qualitativa, um estudante poderia ter detalhado mais, sem apresentar necessariamente um entendimento correto.

Ao final do estudo de caso todas as respostas dos estudantes para os exercícios entregues tanto no estudo piloto, quanto estudo de caso foram reavaliadas para atribuir um nível de compreensão. Nesta avaliação foram avaliadas as respostas para os pontos-chave, hipótese

e o próprio código fonte, com o objetivo de verificar se as duas primeiras respostas apresentam uma especificação coerente, em relação ao código.

O nível de entendimento foi classificado como alto, médio ou baixo, seguindo o mesmo padrão dos níveis de detalhe de pontos-chave e hipótese. A tabela 17 apresenta um resumo da distribuição do nível de compreensão dos exercícios entregues em cada uma das etapas da pesquisa.

As duas fases do estudo de caso apresentaram um número significativamente maior de atividades com níveis altos de compreensão, ao contrário do estudo piloto no qual mais de 85% dos exercícios teve níveis médio ou baixo. Entretanto, mesmo com o aumento do nível mais alto de compreensão no estudo de caso, em todas as predomina o nível médio de compreensão. Uma das hipóteses para o incremento na compreensão pode ser o estímulo para uma maior reflexão sobre o problema e uma sistematização do método ao longo do semestre e não apenas em um período da disciplina, como ocorreu no estudo piloto.

Tabela 17 – Estatísticas sobre o nível de compreensão em cada etapa da pesquisa

Etapa	Nível de compreensão					
	Alto	%	Médio	%	Baixo	%
Estudo piloto	12	14,30%	43	51,20%	29	34,50%
Estudo de caso: fase 1	59	21,30%	140	50,50%	78	28,20%
Estudo de caso: fase 2	38	25,90%	66	44,90%	43	29,20%

Fonte: Elaborado pelo autor

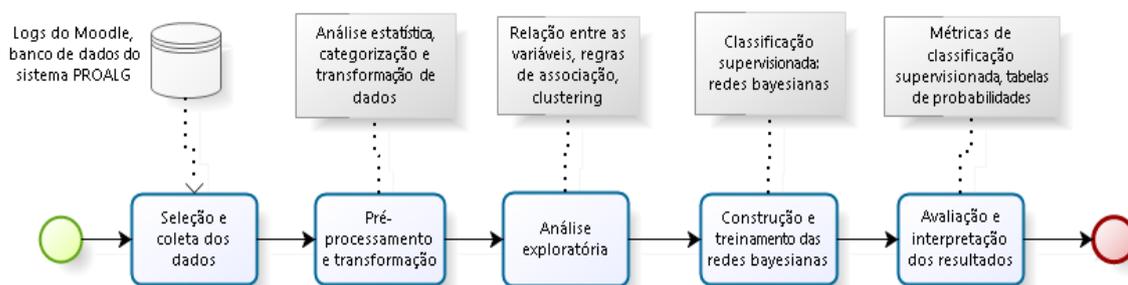
Outro aspecto a ser ressaltado é que no estudo de caso, o questionário de autorrelato do estudante foi ampliado, foram incluídas as questões propostas na REA (Roda dos Estados Afetivos) como ferramenta para conhecer melhor o estado afetivo do estudante. Os resultados das respostas dos estudantes para estas questões serão apresentados no capítulo 7, em conjunto com as demais questões, já aplicadas no estudo piloto.

5 RECONHECIMENTO DA MOTIVAÇÃO: DESENVOLVIMENTO DO MODELO AFETIVO

Este capítulo descreve o desenvolvimento do modelo afetivo que foi criado para responder a questão problema apresentada na seção 1.3, que propôs a investigação sobre a possibilidade de apoiar o processo de ensino e aprendizagem identificando os fatores associados à motivação dos estudantes. O modelo apresentado constitui-se portanto em um dos resultados principais do trabalho, sendo essencial para a implementação do módulo para acompanhamento da motivação descrito na seção 5.5.

As atividades desenvolvidas para elaborar e implementar o modelo para reconhecimento da motivação seguem o fluxo para descoberta de conhecimento proposto por Fayad, Piatetsky-Shapiro e Smyth (1996), que inclui a seleção dos dados, o pré-processamento e transformação, a aplicação das técnicas de mineração de dados e a interpretação e avaliação do conhecimento obtido para a elaboração do modelo composto pelas RB. As atividades desenvolvidas estão demonstradas no processo representado na figura 41.

Figura 41 – Processo de desenvolvimento do modelo afetivo



Fonte: Elaborado pelo autor

A seleção e coleta dos dados iniciou durante o estudo piloto, tendo continuidade no estudo de caso (seção 4.2 e 4.3). As principais variáveis foram definidas ainda no estudo piloto, a partir da análise da viabilidade de reconhecer o estado afetivo do estudante, e complementadas no estudo de caso, com o objetivo de responder as principais perguntas e hipóteses sobre como reconhecer o estado afetivo da motivação.

Inicialmente os logs do AVEA Moodle se constituíram na principal fonte de dados, sendo necessário gerar os logs no ambiente e analisar os registros para obter os tempos, dados sobre acesso e conteúdos das respostas. Com a adoção do sistema PROALG foi possível executar consultas diretamente na base de dados para obter os dados necessários.

O pré-processamento e a transformação é uma atividade fundamental para a aplicação dos algoritmos de mineração de dados. Medidas estatísticas, como média, desvio padrão,

distribuições de frequência, combinadas com técnicas de mineração para agrupamento e extração de regras foram usadas com o objetivo de analisar, categorizar os dados e encontrar relações entre as entradas, ocorrências. As atividades desenvolvidas e o resultado da etapa de pré-processamento e transformação serão descritas nas próximas seções.

No estudo piloto foi possível formular hipóteses, como, por exemplo a de que estudantes que apresentam tempos muito elevados na definição dos pontos-chave do problema e tempos muito reduzidos ou mínimos na especificação da solução e elaboração do código podem estar demonstrando um baixo nível de esforço para compreender e refletir sobre o problema. O estudo de caso contribuiu para aprimorar e adequar as hipóteses e descobertas feitas no estudo piloto.

Tempos muito reduzidos nas duas primeiras etapas e superiores na construção do código fonte também podem evidenciar um menor nível de esforço. A aplicação das técnicas de mineração em uma análise exploratória buscou relacionar estes tempos a um menor detalhamento das respostas para os pontos-chave e para a hipótese de solução, o que poderia indicar uma dependência entre estes atributos.

Com relação a confiança e independência, foram formuladas hipóteses relacionando o acesso às dicas e tempos muito reduzidos ou muito elevados à uma maior ou menor confiança. Os estudantes que apresentaram tempos fora do padrão, muito acima da média podem estar relacionados a um maior acesso às opções de ajuda, o que poderia caracterizar uma falta de confiança, por exemplo.

Na etapa definida como análise exploratória estas hipóteses e relações foram avaliadas, utilizando para isso técnicas baseadas em agrupamento e para extração de regras. O agrupamento foi empregado para indicar quais valores de atributos apareceram associados em determinados grupos. A extração de regras buscou estabelecer relações entre os valores dos atributos, buscando verificar se um valor específico em um atributo aparecia associado outro valor em um atributo diferente. Os resultados da aplicação destas técnicas serão apresentados posteriormente nas seções que descrevem os modelos específicos para esforço e confiança.

A última etapa do processo foi a construção e o treinamento das redes bayesianas que permitiram reconhecer o esforço e a confiança de cada atividade e do estudante ao longo das atividades. Complementarmente foi aplicado método *BayesNet* para classificação supervisionada com o intuito de verificar a eficiência e robustez deste método. Métricas para análise de classificadores foram empregadas na interpretação e análise dos resultados obtidos.

A seguir serão apresentados os modelos para o reconhecimento da motivação e os resultados do processo de desenvolvimento. Como mostrado na seção 2.6.3 o modelo para reconhecimento da motivação é dividido em quatro componentes, de acordo com o contexto (atividade ou geral do estudante) e fator (esforço e confiança).

5.1 Reconhecimento do esforço em cada atividade

O primeiro componente do modelo afetivo é utilizado para definir os níveis de esforço do estudante em cada tarefa. O objetivo é perceber se o estudante apresenta um nível alto ou baixo de esforço. O foco principal sob o ponto de vista do professor é observar os estudantes que demonstram baixos níveis de esforço, principalmente se isso ocorre reiteradamente. A combinação de níveis baixos em mais atividades também pode indicar a necessidade de uma intervenção junto ao estudante.

Para composição do modelo foram definidas diferentes variáveis, sendo parte delas relacionadas aos tempos coletados durante a resolução dos problemas. Outro grupo de atributos corresponde à avaliação feita pelo pesquisador em conjunto com o professor quanto ao conteúdo das respostas dos estudantes em cada etapa, os pontos-chave do problema, a hipótese e o código.

5.1.1 Pré-processamento e transformação: variáveis relacionadas ao esforço

As variáveis temporais relacionadas à resolução de cada atividade são quantitativas, sendo definidas como os minutos dedicados a cada uma das etapas, além dos somatórios de alguns destes tempos. Utilizou-se como padrão a unidade de tempo minuto, pois não é necessário especificar a quantidade de segundos. Como posteriormente ocorre uma classificação dos tempos em categorias (alto, médio e baixo) um detalhamento maior, em segundos, por exemplo, torna-se excessivo. Neste caso o tempo mínimo registrado é de 1 minuto, mesmo nos casos em que o aluno dedicou somente alguns segundos para resolução.

Tanto no sistema PROALG, quanto no Moodle foi possível obter a data e hora inicial sempre que o estudante inicia a etapa. No PROALG quando a tela relacionada ao problema, hipótese ou código é exibida, o registro da data e hora é incluído na tabela de ocorrências, com a identificação da etapa. A finalização do passo atual e conseqüentemente o início do passo seguinte gera um novo registro de término nesta mesma tabela, o que permite calcular exatamente o tempo de cada tarefa.

A utilização de técnicas de mineração de dados em muitos casos demanda uma transformação dos dados para a sua aplicação. A maioria dos algoritmos têm dificuldades para

agrupar ou classificar entradas baseadas em variáveis que possuem um grande número de valores possíveis, ou tipos de dados com valores contínuos. Técnicas de Redes Bayesianas, por exemplo apresentam melhor desempenho com valores discretizados, pois desta forma conseguem associar as probabilidades relacionadas aos valores das entradas com as possíveis classes de cada atributo dependente (FLORES e BARONE, 2003).

Após uma análise da distribuição dos valores, foi tomada a decisão de categorizar as variáveis que indicam os tempos em três classes diferentes: “ALTO”, “MEDIO” ou “BAIXO”. Embora em cada atividade, os tempos sejam diferentes e apresentem variações, foi necessário adotar um padrão para os atributos temporais, evitando um número excessivo de classes.

A discriminação dos tempos em mais classes, tais como muito baixo, muito alto não contribuiria significativamente para indicar que há uma maior probabilidade do esforço ser baixo ou alto e acarretaria em maiores dificuldades de processamento para os métodos de classificação.

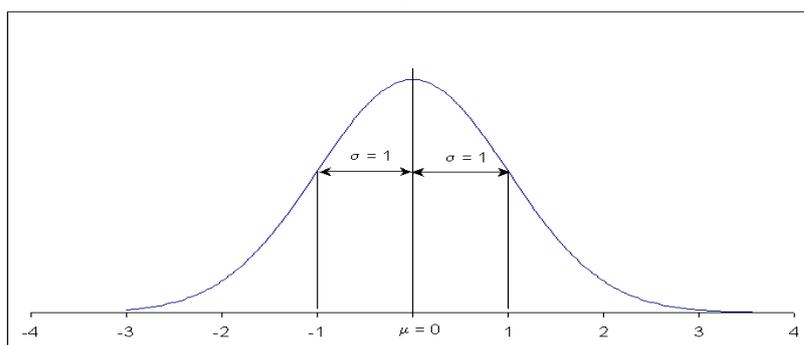
Os atributos temporais foram categorizados levando em consideração a distribuição dos tempos em cada amostra, considerando que cada atividade é uma amostra diferente. Ou seja, o enquadramento na categoria leva em conta a distribuição dos tempos em cada atividade distinta. Esta abordagem foi usada em virtude dos tempos variarem de uma atividade para a outra, principalmente pela complexidade e maior necessidade de análise e pesquisa sobre o problema.

O principal objetivo ao categorizar é indicar quais tempos estão abaixo do esperado para a atividade e quais estão acima. Os valores que estão dentro de limites considerados normais são categorizados como médio e os demais são definidos como baixo ou alto. A principal questão que emergiu ao longo da pesquisa esteve relacionada aos limites usados para esta classificação, ou seja, abaixo de qual valor pode ser considerado baixo e acima de quanto seria alto.

Uma das possibilidades avaliadas inicialmente foi a utilização da média e do desvio padrão como balizadores, considerando que os valores que estivessem abaixo da média subtraída do desvio padrão seriam baixos e os que estivessem um acima da média mais o desvio, poderiam ser considerados altos. Esta estratégia poderia permitir a identificação dos valores discrepantes, tanto para cima, quanto para baixo. Entretanto, segundo (Larson e Farber, 2004), o uso destas medidas para identificação das categorias é indicado quando a amostra se constitui em uma distribuição próxima do normal, na qual a maior parte dos valores está próxima da média, dentro dos limites do desvio calculado.

Analisando a distribuição dos valores coletados em diferentes tarefas, observou-se que as amostras em muitos casos não seguem uma distribuição normal e medidas como a média ou o desvio padrão em muitos casos são pouco relevantes para indicar a concentração dos valores. Uma distribuição normal, segundo Larson, Farber e Patarra (2004) apresenta um formato de sino e é simétrica em torno da média, além de aproximar-se mais do eixo x à medida que se afasta da média em ambos os lados (Figura 42).

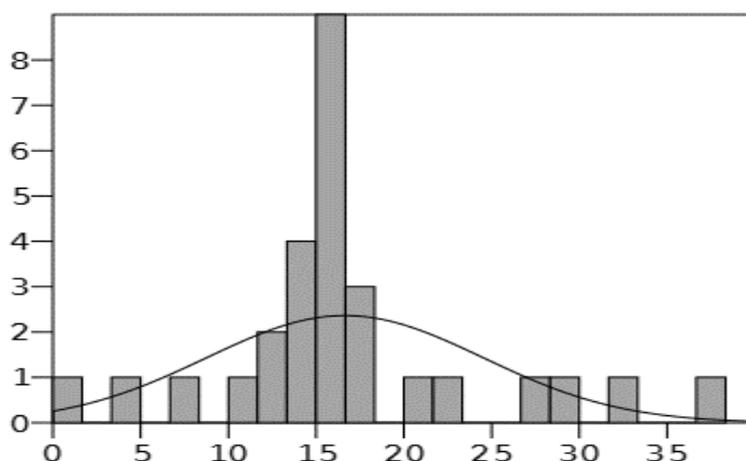
Figura 42 – Exemplo de curva normal



Fonte: Larson, Farber e Patarra (2004)

Em alguns exercícios, é possível identificar uma distribuição mais equilibrada, próxima a uma curva normal e nestes casos média é significativa como no exemplo dos tempos do problema da atividade 1, do estudo de caso, mostrados na figura 43. Neste exemplo, a média é 16,60 e o desvio padrão 7,80, medidas que indicam uma variação relativa em relação à média.

Figura 43 - Gráfico com histograma e curva para os tempos do problema da atividade 1 do estudo de caso



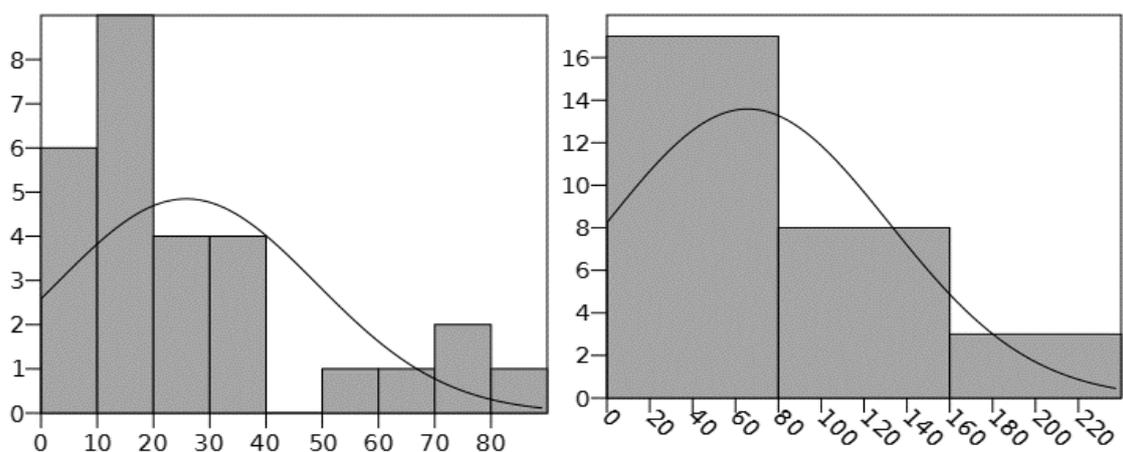
Fonte: Elaborado pelo autor

Em vários exercícios, porém, o desvio padrão aproxima-se do valor da média, indicando que há uma variação grande dos tempos, o que dificulta o uso destas medidas para definir a

categorização. Pode-se utilizar como exemplo as atividades 2 e 11 do estudo de caso, mostradas na figura 44. No primeiro caso, a média é de 25,80 e o desvio padrão de 23,00. Já na segunda série, há uma variação ainda maior, com média de 65,70 e desvio apresentado quase o mesmo valor.

Em ambos os casos os valores da moda e mediana são inferiores à média, o que caracteriza, de acordo com Spiegel e Stephens (2009) uma curva inclinada à direita. É possível constatar que as curvas não seguem um padrão normal, com uma concentração de valores nos intervalos iniciais, embora existam valores altos, que influenciam tanto a média, quanto o desvio padrão. O uso da média menos o desvio padrão nestes casos implicaria na quase não existência de valores baixos, em ambos os casos. Ambas as amostras são fortemente influenciadas pela existência de valores muito altos, se comparados com a maioria, gerando um desvio acima do normal.

Figura 44 – Gráficos com histograma e curva para atividades com maior variabilidade nos tempos



Fonte: Elaborado pelo autor

Definir limites e categorias para conjuntos de dados como os que estão representados na figura 44 é uma tarefa que envolve uma análise da distribuição em relação ao rol de valores. Para o presente trabalho, categorizar um valor como baixo ou alto é fundamental para a determinação do esforço e para tanto foi necessário definir um método que pudesse ser aplicado às diferentes amostras, mesmo que estas apresentassem valores de amplitudes muito diversas.

A partir da análise da distribuição dos valores e das diferentes formas possíveis para classificação, foram avaliadas as medidas denominadas percentis e quartis, que consideram um conjunto de dados ordenados. Segundo Spiegel e Stephens (2009) o percentil, quartil são medidas que permitem dividir o conjunto de dados em partes iguais, sendo que os quartis de uma amostra se referem a quatro partes.

Os quartis podem ser considerados indicadores de localização, que mostram se um valor encontra-se na primeira, segunda, terceira ou quarta parte dos dados (Spiegel e Stephens, 2009), desta forma, torna-se possível encontrar os limites de cada uma das faixas e utilizar estes como forma de categorização. Ao utilizar estas medidas foi possível adotar uma forma padrão, ou regra para categorizar qualquer variável de tempo, em qualquer amostra, independente da amplitude dos dados, uma vez que os limites das faixas serão calculados de acordo com a lista ordenada de valores.

As regras para categorização das variáveis temporais foram as seguintes:

- Para cada amostra, calcular o primeiro e o terceiro quartil.
- Avaliar cada tempo, se for inferior ou igual ao valor do primeiro quartil, a categoria é “BAIXO”.
- Se o valor for maior ou igual ao valor do terceiro quartil, a categoria é “ALTO”.
- Os demais valores são classificados como “MEDIO”.

As regras permitiram classificar como baixos e altos aqueles valores que apresentam maior discrepância, os mais próximos do limite mínimo e do limite máximo. Valores iguais aos limites dos quartis calculados serão incluídos nas classes alto e baixo, o que pode em alguns casos fazer com que um percentual superior a 25% seja verificado nestas classes. A tabela 18 apresenta os quartis, os dados e as classes definidas para cada valor, em uma amostra.

Os valores apresentados indicam duas atividades com diferenças na distribuição dos valores, no primeiro caso o maior valor corresponde a 82 minutos, mas a maioria dos valores fica entre 10 e 30 minutos. No segundo, existem alunos que levaram até 180 minutos para realização da atividade, o que mostra uma variação bem maior na amplitude dos dados. Estas amostras foram mostradas pelo fato de apresentarem desvios altos e curvas assimétricas, demonstrando a dificuldade para determinação da classe dos tempos.

Nos casos de uma quantidade tão grande de tempo é possível questionar se os alunos não estariam já resolvendo todas as etapas, inclusive escrevendo o código no início da atividade. Entretanto, não é possível descartar que os tempos maiores indiquem realmente uma tentativa do estudante de compreender o problema, de pesquisar a respeito do mesmo com o objetivo de analisar melhor a situação. Desta forma, optou-se por não alterar ou descartar tempos muito altos, foram mantidos os tempos originais coletados, até mesmo em função da amostra para cada atividade ser reduzida.

As médias e os desvios calculados também demonstraram, especialmente na segunda amostra uma alta variância nos dados, o que demonstra que utilizar a média para a discretização

dos dados nem sempre é possível. O primeiro e o segundo quartil foram os limites usados para categorizar como baixo, alto e médio. Na segunda amostra os valores ficam bastante distantes, desta forma, apenas valores realmente altos ou baixos ficarão nestas classes. Embora alguns tempos que foram classificados como médios estejam bastante acima do esperado, como no caso de 131 minutos, o padrão padrão para a discretização dos tempos se mostrou adequado, embora as regras usadas não tenham sido perfeitas para todas as situações.

Tabela 18 – Tempos para definição dos pontos-chave do problema nas atividades 2 e 11 do estudo de caso.

Aluno	Atividade 2		Atividade 11	
	Tempo	Classe	Tempo	Classe
Aluno 1	4	BAIXO	9	BAIXO
Aluno 2	17	MEDIO	90	MEDIO
Aluno 3	79	ALTO	120	MEDIO
Aluno 4	6	BAIXO	3	BAIXO
Aluno 5	11	MEDIO	7	BAIXO
Aluno 6	10	BAIXO	135	MEDIO
Aluno 7	61	ALTO	6	BAIXO
Aluno 8	36	ALTO	42	MEDIO
Aluno 9	1	BAIXO	13	MEDIO
Aluno 10	19	MEDIO	160	ALTO
Aluno 11	22	MEDIO	157	ALTO
Aluno 12	10	BAIXO	180	ALTO
Aluno 13	10	BAIXO	9	BAIXO
Aluno 14	1	BAIXO	10	BAIXO
Aluno 15	8	BAIXO	12	MEDIO
Aluno 16	30	MEDIO	160	ALTO
Aluno 17	72	ALTO	52	MEDIO
Aluno 18	7	BAIXO	1	BAIXO
Aluno 19	20	MEDIO	15	MEDIO
Aluno 20	32	MEDIO	39	MEDIO
Aluno 21	50	ALTO	141	ALTO
Aluno 22	29	MEDIO	148	ALTO
Aluno 23	19	MEDIO	150	ALTO
Aluno 24	82	ALTO	14	MEDIO
Aluno 25	15	MEDIO	9	BAIXO
Aluno 26	33	ALTO	14	MEDIO
Aluno 27	24	MEDIO	15	MEDIO
Aluno 28	16	MEDIO	131	MEDIO
Primeiro quartil	10		9,75	
Terceiro quartil	32,25		136,50	
Média	25,80		68,78	
Desvio padrão	23,00		65,77	

Fonte: Elaborado pelo autor

A seguir serão apresentadas as variáveis definidas para a classificação do esforço associado aos dados de cada atividade, realizada por algum estudante. Os atributos são divididos em três grupos, o primeiro relacionado aos tempos e o segundo composto pelos resultados da avaliação feita pelo professor do conteúdo postado pelos estudantes na resolução da tarefa. O registro da execução e acesso ao pseudocódigo foram apresentados no terceiro grupo, por não se caracterizarem como tempos, nem como avaliações de conteúdo.

A categorização adotada para os primeiros quatro atributos foi descrita nos parágrafos anteriores, em todos os casos foram usados os quartis e as categorias “ALTO”, “MEDIO” e “BAIXO”. Apenas o atributo correspondente à proporção, adotou outra regra que será descrita abaixo. Os nomes entre parênteses são usados para identificar as variáveis nas redes bayesianas, além de serem iguais aos nomes dos campos nas tabelas que serão mostradas no módulo para acompanhamento da motivação no PROALG. As variáveis relacionadas aos tempos são as seguintes:

- Tempo para definição dos pontos-chave do problema (classe_tp_prob_ef): quantidade de minutos decorridos desde que o usuário iniciou a tarefa, até o momento em que a resposta foi postada.
- Tempo para definição da hipótese de solução (classe_tp_hip_ef): segue o mesmo cálculo do tempo anterior e é categorizado também como alto, médio e baixo.
- Tempo para a elaboração do código fonte (classe_tp_cod_ef): tempo que o estudante levou para escrever, compilar, testar e postar o código. Nas situações em que o estudante já possui o código pronto este tempo pode ser bem reduzido, o que será perceptível na variável seguinte. Este tempo é categorizado como alto, médio e baixo, da mesma forma que os anteriores.
- Proporção de tempo das duas primeiras etapas em relação a escrita do código (classe_prop_ph): é calculado o percentual da soma das etapas iniciais (pontos-chave e hipótese) em relação ao tempo total.

Sobre o último atributo descrito, uma análise dos dados coletados indicou que percentuais inferiores a 20% ou 80% estão fora do padrão, pois a grande maioria dos estudantes apresenta proporções dentro destes limites. Existem duas hipóteses, a primeira é que o estudante demandou um tempo insuficiente para compreender e especificar o problema e a segunda é que a escrita do código já foi feita nas primeiras etapas, o que também poderia indicar menor esforço na problematização. Desta forma, a variável assume dois valores: “PROPORCIONAL”,

“DESPROPORCIONAL”, o primeiro indicando que o percentual ficou dentro dos limites e o segundo que ficou abaixo ou acima.

A seguir serão apresentadas os níveis de detalhamento e de compreensão, que foram atribuídos pelo professor e pelo pesquisador na avaliação dos conteúdos. Estas variáveis foram definidas em função da necessidade de considerar não somente os tempos, mas também dados associados à qualidade ou até mesmo o tamanho das soluções produzidas pelo aluno. Na seção que apresentou os resultados do estudo piloto foi apresentada uma definição dos critérios adotados para atribuir a classe correspondente ao nível de detalhe nestes atributos.

Os níveis de detalhamento e a compreensão foram categorizados como “ALTO”, “MEDIO” e “BAIXO”, mantendo a distinção em 3 categorias, como nos tempos. As variáveis são as seguintes:

- Nível de detalhe dos pontos-chave (*classe_nivel_detalhe_prob*): classificação como alto, médio e baixo da descrição dos pontos-chave do problema. Esta classificação foi realizada pelo professor e pelo pesquisador a partir da análise qualitativa do conteúdo postado.
- Nível de detalhe da hipótese (*classe_nivel_detalhe_hip*): classificação como alto, médio e baixo da descrição da hipótese de solução para a atividade.
- Nível de compreensão (*classe_nivel_compreensao*): avaliação da compreensão demonstrada pelo aluno, a partir da descrição dos pontos-chave e da hipótese. Segue o padrão de classificação, alto, médio e baixo. Os níveis de detalhe consideram principalmente o fato do aluno ter apresentado uma descrição mais completa, mais detalhada e o nível de compreensão indica se o estudante demonstrou claramente como pretende elaborar a solução, se ficou explícito que ele analisou e compreendeu claramente o problema antes da elaboração do código. Esta análise é também qualitativa e embora apresente um grau de subjetividade é importante por incluir um aspecto relacionado ao desempenho do estudante, uma avaliação da qualidade dos conteúdos postados.

Os dois últimos atributos que formam o modelo para reconhecimento do esforço em cada atividade são apresentados abaixo:

- Acesso ao pseudocódigo (*visualizou_pseudo*): indica se o usuário acessou a ajuda que exibe o código fonte. O acesso às demais dicas não é considerado como indicativo de menor ou maior esforço, porém, o pseudocódigo representa uma especificação detalhada do problema, a sua visualização, conclui-se, portanto, que

pode ser um indicador também relacionado ao esforço. O atributo possui os valores SIM ou NÃO, indicando apenas se foi acessado pelo estudante.

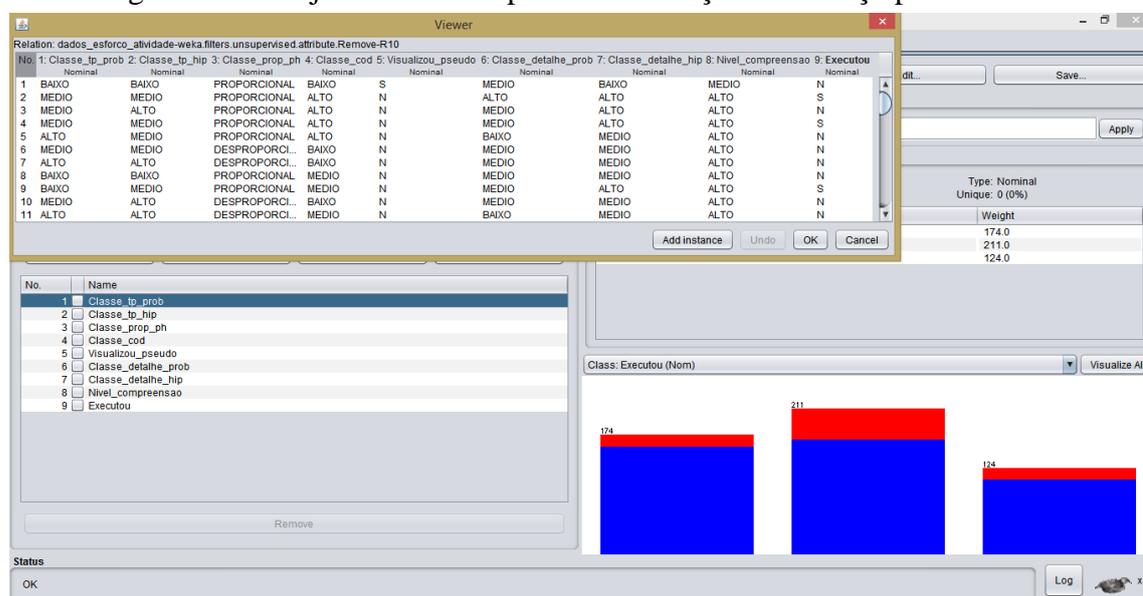
- Executou código (execucao_codigo): indica se o estudante conseguiu remover os erros de compilação e executar, pelo menos uma vez o código fonte. Na maioria dos casos é necessário um certo grau de esforço para chegar a uma solução executável, portanto, o uso desta variável é importante na classificação do esforço demonstrado na atividade. Os valores possíveis são SIM ou NÃO.

5.1.2 Reconhecimento do esforço por atividade: análise exploratória

Após definir as variáveis que seriam utilizadas para o esforço, o próximo passo consistiu em uma análise exploratória que visou encontrar relações entre as mesmas no conjunto de dados e subsídios para uma classificação a priori, ou seja, para determinar a qual classe cada entrada seria associada para posteriormente realizar o treinamento das redes bayesianas.

Para a aplicação dos algoritmos foi necessário converter os dados de entrada em um arquivo no formato arff, uma vez que a ferramenta escolhida para esta aplicação foi o Weka. A figura 45 mostra o arquivo carregado na ferramenta e uma amostra dos dados e valores dos atributos.

Figura 45 – Conjunto de dados para determinação do esforço por atividade



Fonte: Elaborado pelo autor

Durante a análise exploratória foram usados diferentes algoritmos de agrupamento e extração de regras e os testes envolveram diversas configurações em relação ao número de grupos, quantidade de regras, entre outros parâmetros. Serão apresentados os principais

resultados que sintetizam os demais e que permitiram estabelecer algumas conclusões sobre os dados coletados.

A exploração do conjunto de dados iniciou pela execução de técnicas baseadas em agrupamento, com destaque para o algoritmo *SimpleKmeans*, que é um dos métodos mais conhecidos e utilizados. A ideia inicial era separar o conjunto de entradas em 3 grupos distintos, que seriam as classes “ALTO”, “MEDIO”, “BAIXO” e por este motivo o parâmetro que indica o número de clusters foi configurado para gerar três grupos. A figura 46 mostra os grupos gerados e os centroides destes.

Figura 46 – Resultados da aplicação do *SimpleKmeans* nos dados relacionados ao esforço por atividade

```

Number of iterations: 3
Within cluster sum of squared errors: 1302.0

Initial starting points (random):

Cluster 0: MEDIO,BAIXO,DESPORCIONAL,BAIXO,N,BAIXO,BAIXO,BAIXO,N
Cluster 1: MEDIO,MEDIO,PROPORCIONAL,ALTO,N,MEDIO,MEDIO,MEDIO,S
Cluster 2: MEDIO,MEDIO,PROPORCIONAL,MEDIO,N,BAIXO,MEDIO,ALTO,N

Missing values globally replaced with mean/mode

Final cluster centroids:

```

Attribute	Full Data (509.0)	Cluster#		
		0 (198.0)	1 (203.0)	2 (108.0)
Classe_tp_prob	MEDIO	BAIXO	MEDIO	MEDIO
Classe_tp_hip	MEDIO	BAIXO	MEDIO	MEDIO
Classe_prop_ph	PROPORCIONAL	DESPORCIONAL	PROPORCIONAL	PROPORCIONAL
Classe_cod	MEDIO	BAIXO	MEDIO	MEDIO
Visualizou_pseudo	N	N	N	N
Classe_detalhe_prob	BAIXO	BAIXO	MEDIO	BAIXO
Classe_detalhe_hip	MEDIO	BAIXO	MEDIO	MEDIO
Nivel_compreensao	MEDIO	BAIXO	MEDIO	MEDIO
Executou	N	N	N	N

Fonte: Elaborado pelo autor

Ao observar os resultados mostrados na figura 46, verifica-se que os dois últimos grupos apresentam valores muito semelhantes para a maioria dos atributos. Foram feitos testes com um número maior de grupos, porém, a principal constatação foi que um número maior de clusters não contribuía para uma visualização mais apurada do perfil dos diferentes grupos. Uma das hipóteses formuladas neste momento foi de que um número de 3 grupos talvez fosse excessivo, porém, o número de 3 classes para o esforço foi mantido neste momento, com o objetivo de realizar mais testes posteriormente.

Em relação a configuração dos grupos percebe-se que em um deles há uma combinação ou correlação das categorias correspondentes a BAIXO, que são considerados negativos para determinar o esforço. Tal constatação corroborou com a ideia de estabelecer pontuações positivas e negativas para a classificação a priori, que será explicada posteriormente.

Na sequência o algoritmo Apriori foi empregado para extrair regras que contribuíssem para indicar relações ou dependências entre os atributos. A figura 47 mostra as principais regras que emergiram do processo de mineração. São apresentadas as 10 regras mais relevantes, embora tenham sido feitos testes com um número maior de regras.

Figura 47 – Resultados da aplicação do Apriori nos dados relacionados ao esforço por atividade

```

1. Classe_prop_ph=DESPROPORCIONAL Classe_detalhe_prob=BAIXO Classe_detalhe_hip=BAIXO 86 ==> Nivel_compreensao=BAIXO 84 <conf:(0.98)> lift:(2.62) lev:(0.1) [51] cov:
2. Classe_detalhe_prob=BAIXO Classe_detalhe_hip=BAIXO 110 ==> Nivel_compreensao=BAIXO 107 <conf:(0.97)> lift:(2.61) lev:(0.13) [65] conv:(17.23)
3. Visualizou_pseudo=N Classe_detalhe_prob=BAIXO Classe_detalhe_hip=BAIXO 92 ==> Nivel_compreensao=BAIXO 89 <conf:(0.97)> lift:(2.59) lev:(0.11) [54] conv:(14.41)
4. Classe_prop_ph=DESPROPORCIONAL Classe_cod=BAIXO Classe_detalhe_prob=BAIXO 99 ==> Visualizou_pseudo=N 93 <conf:(0.94)> lift:(1.12) lev:(0.02) [10] conv:(2.31)
5. Classe_tp_prob=MEDIO Classe_cod=MEDIO Visualizou_pseudo=N 87 ==> Classe_prop_ph=PROPORCIONAL 81 <conf:(0.93)> lift:(1.81) lev:(0.07) [36] conv:(6.03)
6. Classe_cod=BAIXO Classe_detalhe_hip=MEDIO 85 ==> Visualizou_pseudo=N 79 <conf:(0.93)> lift:(1.11) lev:(0.02) [7] conv:(1.98)
7. Classe_tp_prob=MEDIO Classe_cod=MEDIO 99 ==> Classe_prop_ph=PROPORCIONAL 92 <conf:(0.93)> lift:(1.81) lev:(0.08) [41] conv:(6.01)
8. Classe_cod=BAIXO Classe_detalhe_prob=BAIXO 112 ==> Visualizou_pseudo=N 104 <conf:(0.93)> lift:(1.11) lev:(0.02) [10] conv:(2.03)
9. Classe_cod=BAIXO Visualizou_pseudo=N Nivel_compreensao=BAIXO 82 ==> Classe_prop_ph=DESPROPORCIONAL 76 <conf:(0.93)> lift:(1.91) lev:(0.07) [36] conv:(6.03)
10. Classe_prop_ph=DESPROPORCIONAL Classe_cod=BAIXO Nivel_compreensao=BAIXO 82 ==> Visualizou_pseudo=N 76 <conf:(0.93)> lift:(1.11) lev:(0.01) [7] conv:(1.91)

```

Fonte: Elaborado pelo autor

Os resultados demonstram que existe uma relação entre um baixo nível de compreensão e baixos níveis de detalhamento para a descrição do problema e da hipótese. Estas relações suscitaram a possibilidade de existir uma influência das variáveis citadas no nível de compreensão, embora a relação fosse apenas para os casos de baixa compreensão. Outra regra interessante indica que um tempo baixo no código combinado com médio ou baixo detalhamento da hipótese estaria relacionado ao não acesso ao pseudocódigo.

5.1.3 Reconhecimento do esforço por atividade: construção e treinamento das redes bayesianas

Para poder classificar os registros utilizados no processo de treinamento dos algoritmos de classificação supervisionada foi necessário definir a classe a qual pertence cada uma das entradas. Esta classificação, a priori foi baseada nos resultados da aplicação dos algoritmos de agrupamento e regras de associação e também em uma análise empírica dos valores das variáveis nos dados coletados durante o estudo de caso.

Para cada atributo, as diferentes categorias foram associadas a pontuações positivas, negativas ou neutras. As classes que indicam tempos e níveis de detalhamento baixos receberam pontuações negativas, enquanto os valores altos receberam pontuação positiva. Níveis médios foram considerados neutros, pois considera-se que não exercem grande influência no esforço demonstrado. É importante ressaltar que foram feitos testes com diferentes pontuações, avaliando em cada caso quantas entradas ficaram em cada classe. A tabela 19 mostra as pontuações finais, usadas para determinação do nível de esforço.

Tabela 19 – Pontuações para determinação da classe correspondente ao esforço por atividade

Variável	Regra e pontuação
Classe_tp_prob_ef	BAIXO = -1; MÉDIO = 0; ALTO = 1
Classe_tp_hip_ef	BAIXO = -1; MÉDIO = 0; ALTO = 1
Classe_tp_cod_ef	BAIXO = -1; MÉDIO = 0; ALTO = 1
Classe_prop_ph	PROPORCIONAL = 1; DESPROPORCIONAL = -1
Visualizou_pseudo	SIM = -1; NÃO = 1
Execucao_codigo	SIM = 1; NÃO = -1
Classe_nivel_detalhe_prob	BAIXO = -1; MÉDIO = 0; ALTO = 1
Classe_nivel_detalhe_hip	BAIXO = -1; MÉDIO = 0; ALTO = 1
Classe_nivel_compreensao	BAIXO = -1; MÉDIO = 0; ALTO = 1

Fonte: Elaborado pelo autor

A tabela acima foi aplicada a cada uma das entradas e para cada uma delas foi calculado o total de pontos, de acordo com o valor das variáveis. A partir desta tabela foi definida uma regra geral para classificar os registros. Se o valor calculado após aplicar as regras for negativo e inferior a -1, considera-se que o esforço geral associado a atividade é baixo. Em caso de um valor positivo, superior a 1, a mesma é categorizada como alto, e nos demais valores, considerou-se médio. A categoria MEDIO indica uma neutralidade, ou seja, não é possível afirmar que o aluno se esforçou pouco, mas também não houve um esforço considerável.

Com a definição da pontuação e das regras para determinação da classe, o passo seguinte foi aplicar métodos de classificação supervisionada, com destaque para o algoritmo *ByesNet* (redes bayesianas) com o intuito de verificar se o método era capaz de aprender a classificar corretamente as entradas. Em um primeiro momento o objetivo era classificar em três grupos, porém os resultados mostrados abaixo justificam a utilização de apenas duas classes.

A figura 48 mostra os resultados da aplicação do algoritmo em um conjunto de dados com três classes para o esforço (“ALTO”, “MEDIO”, “BAIXO”). Avaliando as métricas e a matriz de confusão é possível constatar que tanto a acuracidade (84,67), quanto os demais índices não foram totalmente satisfatórios. As principais dificuldades estão relacionadas à determinação da classe “MEDIO”, cuja precisão ficou abaixo das demais (0,476).

Outros algoritmos para classificação, como o J48, baseado em árvore de decisão apresentou resultados semelhantes ao método *BayesNet*. A partir destes resultados, verificou-se a dificuldade de separar as entradas em três classes e foi tomada a decisão de utilizar somente duas classes, alto ou baixo, uma vez que o principal objetivo é identificar os estudantes que

apresentam baixos níveis de esforço. Além disso, o uso das redes bayesianas contribui para indicar a intensidade ou probabilidade de ser baixo ou alto, desta forma, uma classe intermediária tornou-se desnecessária.

Figura 48 – Resultados da classificação do esforço por atividade com 3 classes

```

Correctly Classified Instances      431          84.6758 %
Incorrectly Classified Instances    78           15.3242 %
Kappa statistic                    0.7443
Mean absolute error                 0.1475
Root mean squared error             0.2663
Relative absolute error             37.8673 %
Root relative squared error        60.3888 %
Total Number of Instances         509

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,875   0,039   0,965     0,875   0,918     0,831   0,984    0,987    BAIXO
                0,942   0,071   0,854     0,942   0,896     0,849   0,983    0,961    ALTO
                0,541   0,101   0,476     0,541   0,506     0,417   0,867    0,406    MEDIO
Weighted Avg.   0,847   0,058   0,860     0,847   0,851     0,777   0,967    0,895

=== Confusion Matrix ===

  a  b  c  <-- classified as
245  0  35 |  a = BAIXO
  0 146  9 |  b = ALTO
  9  25  40 |  c = MEDIO

```

Fonte: Elaborado pelo autor

A regra para determinação a priori do esforço foi ajustada para que as entradas cuja soma de pontos fosse inferior a zero fosse classificada como baixo e as demais como alto. Desta forma, as entradas foram segmentadas em duas classes, que passaram a determinar o nível de esforço em cada atividade. O arquivo arff foi novamente gerado, agora com duas classes para o atributo equivalente ao nível de esforço, e o método *BayesNet* foi novamente aplicado para verificar a capacidade do mesmo em classificar corretamente as entradas. Os resultados com apenas duas categorias é mostrado na figura 49.

Figura 49 – Resultados da classificação do esforço por atividade com 2 classes

```

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      469          92.1415 %
Incorrectly Classified Instances    40           7.8585 %
Kappa statistic                    0.8371
Mean absolute error                 0.1134
Root mean squared error             0.2244
Relative absolute error             24.1217 %
Root relative squared error        46.2884 %
Total Number of Instances         509

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,896   0,036   0,976     0,896   0,934     0,842   0,988    0,993    BAIXO
                0,964   0,104   0,849     0,964   0,902     0,842   0,988    0,983    ALTO
Weighted Avg.   0,921   0,062   0,928     0,921   0,922     0,842   0,988    0,989

=== Confusion Matrix ===

  a  b  <-- classified as
284  33 |  a = BAIXO
  7 185 |  b = ALTO

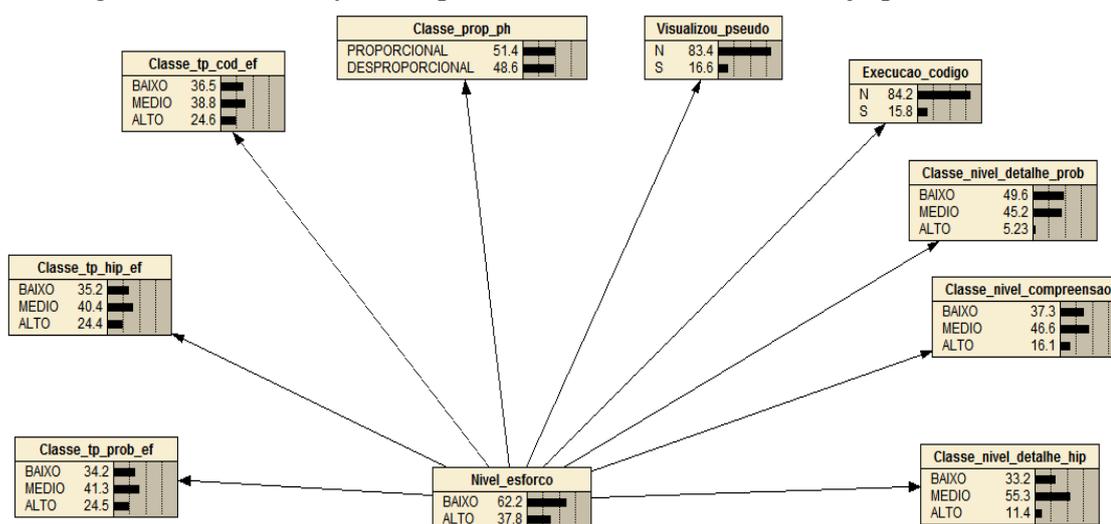
```

Fonte: Elaborado pelo autor

Após os ajustes da tabela e da regra utilizada para classificação inicial dos dados, a técnica de Redes Bayesianas foi aplicada com o objetivo de obter uma tabela de probabilidades que será utilizada como base para a implementação do módulo para o reconhecimento do esforço do estudante. Os dados para o treinamento foram os mesmos utilizados na ferramenta Weka, porém, foi preciso criar um arquivo em formato texto.

O software Netica foi utilizado para construção da rede bayesiana, na qual foram definidas as dependências entre os atributos. A arquitetura da Rede Bayesiana é mostrada na figura 61. A estrutura é equivalente a que foi gerada pela aplicação do método *BayesNet* no Weka, na qual todas as variáveis têm dependência do nível de esforço. Esta estratégia foi adotada considerando o objetivo principal que é a determinação do esforço a partir dos valores dos demais atributos.

Figura 50 – Rede Bayesiana para o reconhecimento do esforço por atividade



Fonte: Elaborada pelo autor

A imagem mostra os percentuais que indicam a distribuição dos valores para cada atributo no conjunto de dados de entrada, sendo que a grande maioria dos registros utilizados no processo foi classificada como baixo esforço (62,2%). Nas demais variáveis, tempos e níveis de detalhe a predominância é da classe neutra (MEDIO), com uma maior quantidade de baixos, em relação ao número de valores altos.

O principal objetivo do processo de treinamento e aprendizado da rede bayesiana é a obtenção da tabela de probabilidades que permite classificar o nível de esforço e inferir sobre as probabilidades de todos os atributos em relação aos valores deste atributo. Os percentuais são calculados pelo software Netica e armazenados em cada uma das variáveis, formando o modelo bayesiano para classificação do esforço.

As tabelas de probabilidades são exibidas na figura 51, que exibe um quadro no qual aparece o nome do atributo (node) e uma tabela que exibe uma coluna inicial com os valores das classes do nível de esforço, seguida de três colunas com as probabilidades relacionadas aos valores do próprio atributo (node). Os números indica a chance dos valores aparecerem combinados em cada entrada, ou seja, uma maior ou menor chance do esforço ser alto ou baixo, dado um valor específico da variável.

Chama a atenção as probabilidades associadas ao nível de detalhamento e compreensão altos e de um nível de esforço baixo. Há uma chance mínima do esforço ser determinado como alto nas situações em que os detalhes ou compreensão são avaliados como baixo, o que evidencia a importância de uma avaliação criteriosa por parte do professor na correção dos exercícios. Um alto nível de esforço naturalmente está relacionado a altos percentuais nas categorias alto ou médio.

Observa-se um maior equilíbrio nos percentuais das variáveis que representam os tempos das tarefas. A probabilidade do tempo ser alto nos casos de nível de esforço baixo, por exemplo, é aproximadamente 20% no atributo que indica a classe para especificação dos pontos-chave (classe_tp_prob_ef), o que evidencia uma menor influência desta na determinação do nível de esforço.

Figura 51 – Tabelas de probabilidades da rede bayesiana para o reconhecimento do esforço por atividade

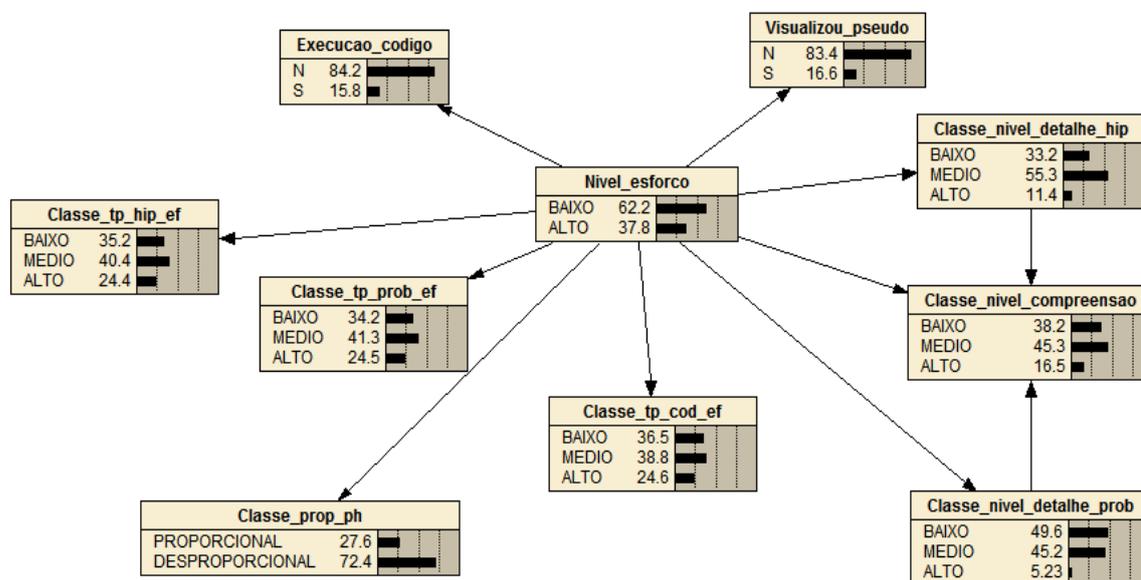
Node	Classe	BAIXO	MEDIO	ALTO
Classe_tp_cod_ef	BAIXO	47.813	33.437	18.75
	MEDIO	17.949	47.692	34.359
	ALTO			
Classe_tp_hip_ef	BAIXO	48.438	37.813	13.75
	MEDIO	13.333	44.615	42.051
	ALTO			
Classe_tp_prob_ef	BAIXO	42.5	36.875	20.625
	MEDIO	20.513	48.718	30.769
	ALTO			
Classe_prop_ph	BAIXO	38.245	61.755	
	MEDIO	73.196	26.804	
	ALTO			
Visualizou_pseudo	BAIXO	80.251	19.749	
	MEDIO	88.66	11.34	
	ALTO			
Execuciao_codigo	BAIXO	95.925	4.075	
	MEDIO	64.948	35.052	
	ALTO			
Classe_nivel_detalhe_prob	BAIXO	66.875	32.5	0.625
	MEDIO	21.026	66.154	12.821
	ALTO			
Classe_nivel_compreensao	BAIXO	58.75	40.938	0.313
	MEDIO	2.051	55.897	42.051
	ALTO			
Classe_nivel_detalhe_hip	BAIXO	50.625	47.813	1.563
	MEDIO	4.615	67.692	27.692
	ALTO			

Fonte: Elaborado pelo autor

Considerando a relação entre o nível de compreensão e os níveis de detalhamento mostrados na análise exploratória, foram testadas outras arquiteturas para a rede, sendo uma delas apresentada na figura 52, na qual existe uma dependência entre os níveis de detalhe e a

compreensão. As simulações com diferentes combinações para os valores dos atributos mostraram pequenas diferenças nas probabilidades, mas em todas as situações a classe para o esforço foi a mesma.

Figura 52 – Rede Bayesiana alternativa para o reconhecimento do esforço



Fonte: Elaborado pelo autor

Com a nova arquitetura o impacto da variável que determina o nível de compreensão foi reduzido, especialmente nas situações em que o valor era médio ou alto, entretanto, a influência na definição do nível de esforço não foi acentuada. Após os testes com as diferentes arquiteturas, optou-se pela utilização da primeira, na qual são calculadas as probabilidades dos atributos em relação ao esforço. A estrutura simplificada foi escolhida devido ao fato de não haver uma convicção sobre a existência de uma dependência entre as variáveis citadas e em função da arquitetura ser equivalente aquela que é gerada automaticamente na ferramenta Weka.

5.2 Reconhecimento da confiança em cada atividade

O segundo componente do modelo geral é utilizado para definir os níveis de confiança do estudante em cada tarefa. O objetivo é perceber se o estudante apresenta um nível alto, baixo ou médio de confiança e atuar para encorajar, estimular o estudante a participar ativamente, o que pode evitar, por exemplo, uma desistência ou abandono do componente curricular.

O modelo para o reconhecimento da confiança relacionada a cada atividade complementa o reconhecimento do esforço, permitindo que o docente acompanhe simultaneamente os dois fatores. Em alguns casos, um aluno pode apresentar, por exemplo, um esforço reduzido, mas uma confiança e indiretamente uma independência alta, o que poderia

indicar que ele acredita que não precisa se dedicar tão intensamente e não necessita auxílio durante as atividades.

Enquanto o esforço foca nos tempos, detalhamento das respostas e compreensão, a confiança considera os acessos aos recursos de ajuda, o número de vezes que o aluno abandona e retoma a atividade e também o nível de compreensão. Os tempos do problema são utilizados, com uma categorização diferente, uma vez que a confiança não é refletida sempre por tempos maiores. Um exemplo que pode ser citado, é um tempo alto, como por exemplo, 180 minutos, que é classificado como alto, representando mais esforço, mas que é considerado fora do padrão, o que pode caracterizar uma menor confiança.

Embora os fatores de confiança e independência não representem a mesma noção, podem estar neste caso, na resolução das tarefas, muito próximos e desta forma, serão utilizadas neste trabalho em uma noção única. As variáveis utilizadas neste trabalho podem indicar tanto a confiança, quanto uma certa independência demonstrada pelo estudante. Ao não acessar as dicas, ao realizar a tarefa dentro de um limite de tempo esperado, por exemplo, é possível inferir que o aluno apresentou bom grau de independência. Tempos muito altos, fora do padrão, por exemplo, poderiam indicar que o aluno está buscando apoio ou que de certa forma está confuso quanto ao que deve ser feito, o que indica baixa independência e confiança ao mesmo tempo. Sendo assim, foi proposto somente um modelo para confiança, uma vez que não foram constatadas diferenças sensíveis entre nível de confiança e independência.

5.2.1 Pré-processamento e transformação: variáveis relacionadas à confiança

Para a identificação da confiança os atributos temporais utilizados foram o tempo para a definição dos pontos-chave do problema e para elaboração da hipótese. Entretanto, a forma de categorização foi diferente da utilizada no modelo relacionado ao esforço, descrito na seção 5.1.1. No caso da confiança, busca-se verificar se o estudante apresentou um tempo considerado normal ou se este foi muito acima dos demais, do que poderia ser considerado normal para a atividade em questão.

É possível inferir que ao exibir um tempo muito acima do esperado, o aluno demonstra insegurança, estar confuso, indeciso na resolução ou não entender o que é proposto, situações que podem indicar uma baixa confiança para elaborar e postar o conteúdo solicitado. Outra hipótese é que um tempo excessivo pode indicar uma dificuldade de agir autonomamente, caracterizando assim, uma baixa independência, enfatizando a relação entre confiança e independência citada anteriormente. Considerando que a problematização envolve uma mudança na forma de resolver a atividade, um estímulo para um comportamento ativo, tempos

acima do normal podem indicar que o estudante está pouco confiante na sua capacidade apresentar uma especificação coerente e completa, ou não assimilou totalmente o método adotado.

Considerando estas hipóteses, optou-se por classificar os tempos em duas categorias, NORMAL e ACIMA, sendo esta última a classe que indica menor confiança. Analisando as amostras de tempos de cada atividade foram analisadas duas possibilidades para categorização, o uso da média ou da mediana de cada atividade. Foi possível verificar que em algumas atividades os valores destas duas medidas eram próximos, enquanto em outras estavam distantes.

Como o principal objetivo é capturar valores que estejam fora do padrão, muito acima do que é esperado, a estratégia adotada foi calcular o valor dobrado da média, ou seja, calcula-se a média dos tempos da atividade e compara-se o tempo de cada aluno com este valor, sendo caracterizados como tempos acima do normal os que forem maiores do que o dobro da média.

Avaliações empíricas e testes com amostras de dados demonstraram que a média poderia ser usada, mas que seria necessária uma regra adicional. Foi possível constatar que em algumas amostras o valor da média era bastante alto, em função de alguns alunos apresentarem tempos altíssimos. Para evitar que alguns destes registros fossem classificados como normais, foi criada uma regra complementar, que determinou que valores acima de 60 (uma hora) fossem sempre categorizados como acima do normal. A determinação deste limite foi feita de forma empírica, observando os dados coletados e levando em consideração a complexidade dos problemas, que é bem menor em disciplinas introdutórias como Algoritmos e Programação, na qual os dados foram coletados.

A maioria das atividades apresentou um número baixo de tempos acima do normal, entretanto, problemas mais complexos apresentaram um número um pouco superior, como mostra a tabela 20, que representa os tempos de dois exercícios do estudo de caso. No primeiro exercício, menos de 5% das atividades tiveram tempos acima, enquanto na atividade 11, aproximadamente 40% ficou nesta categoria.

Para a confiança foram utilizados ainda alguns atributos que já foram relacionados e detalhados no modelo para reconhecimento do esforço por atividade, como, por exemplo, o nível de compreensão e a proporção do tempo gasto nas duas primeiras etapas em relação ao tempo total da atividade.

Tabela 20 – Tempos e determinação da classe para reconhecimento da confiança

Aluno	Exercício 1		Exercício 11	
	Tempo	Classe	Tempo	Classe
Aluno 1	7	NORMAL	9	NORMAL
Aluno 2	17	NORMAL	90	ACIMA
Aluno 3	15	NORMAL	120	ACIMA
Aluno 4	15	NORMAL	3	NORMAL
Aluno 5	21	NORMAL	7	NORMAL
Aluno 6	16	NORMAL	135	ACIMA
Aluno 7	23	NORMAL	6	NORMAL
Aluno 8	13	NORMAL	42	NORMAL
Aluno 9	12	NORMAL	13	NORMAL
Aluno 10	15	NORMAL	160	ACIMA
Aluno 11	28	NORMAL	157	ACIMA
Aluno 12	14	NORMAL	180	ACIMA
Aluno 13	15	NORMAL	9	NORMAL
Aluno 14	1	NORMAL	10	NORMAL
Aluno 15	14	NORMAL	12	NORMAL
Aluno 16	15	NORMAL	160	ACIMA
Aluno 17	38	ACIMA	52	NORMAL
Aluno 18	11	NORMAL	26	NORMAL
Aluno 19	16	NORMAL	15	NORMAL
Aluno 20	16	NORMAL	39	NORMAL
Aluno 21	17	NORMAL	141	ACIMA
Aluno 22	18	NORMAL	148	ACIMA
Aluno 23	33	NORMAL	150	ACIMA
Aluno 24	14	NORMAL	14	NORMAL
Aluno 25	15	NORMAL	9	NORMAL
Aluno 26	14	NORMAL	14	NORMAL
Aluno 27	4	NORMAL	15	NORMAL
Aluno 28	29	NORMAL	131	ACIMA
Média	16,6		68,1	

Fonte: Elaborado pelo autor

Com relação aos acessos às dicas e ao pseudocódigo, a estratégia adotada no pré-processamento foi apenas registrar se ocorreu acesso ou não, não sendo adotada uma medida do número de vezes que a dica foi lida. A conversão envolve simplesmente gerar um caractere (“S” ou “N”) a partir dos registros das ações dos usuários na base do sistema.

Outro dado coletado para indicar a confiança diz respeito ao número de vezes que um estudante inicia, para e retoma a atividade, o que no caso dos logs do ambiente Moodle gera um registro específico. No PROALG o estudante pode clicar no botão para parar o exercício,

sair da página ou retornar ao menu. Estes registros foram usados para gerar o atributo que inicialmente foi denominado de retomadas.

Assim como nos demais atributos, foi definida uma regra para discretização dos valores desta variável. Foram definidas três classes possíveis, que indicaram que não houve retomada (NENHUMA), que houve uma ou duas retomadas (POUCAS), ou que o estudante parou e retornou mais de duas vezes (MUITAS). Os valores utilizados para definição das categorias foram definidos empiricamente, a partir de uma análise dos dados coletados, considerando que uma maior linearidade, menor número de entradas e saídas pode ser considerada como uma característica associada a um maior grau de confiança.

Ao final da atividade foi solicitado que o aluno avaliasse a dificuldade da atividade, a partir de três opções possíveis, fácil, difícil ou médio. Esta avaliação foi inserida no sistema PROALG como uma ação obrigatória para finalizar a tarefa, portanto, todos os estudantes devem selecionar uma das alternativas para entregar a atividade. Este atributo consiste em um autorrelato do estudante sobre a sua percepção a respeito de cada um dos exercícios realizados, o que pode indicar se ele está mais confiante na sua capacidade de resolução, por ter achado fácil ou se a confiança é menor, por ter encontrado dificuldades durante a resolução.

Segundo Fayad e (2006) o pré-processamento inclui em muitos casos o preenchimento de dados faltantes, em algumas entradas. Durante o estudo piloto não foi solicitada uma avaliação da complexidade da atividade, embora os estudantes pudessem se manifestar em fóruns criados para este fim. Em algumas atividades foi possível inferir o grau de dificuldade na opinião do aluno pela sua manifestação, porém em outros não havia informações disponíveis para determinar o valor desta variável. Nestes casos foi utilizada uma distribuição proporcional, compatível com os registros coletados no estudo de caso, como estratégia para preencher os valores faltantes para esta variável. Atividades cuja complexidade era menor na maioria dos casos foi avaliada como fácil pelos discentes, ao contrário das atividades de maior complexidade.

A seguir serão apresentadas as variáveis usadas no reconhecimento da confiança em cada atividade:

- Tempo para definição dos pontos-chave do problema (classe_tp_prob_cf): classificação indicando se o tempo foi NORMAL ou ACIMA.
- Tempo para a definição da hipótese (classe_tp_hip_cf): segue o mesmo padrão de classificação do tempo para o problema.

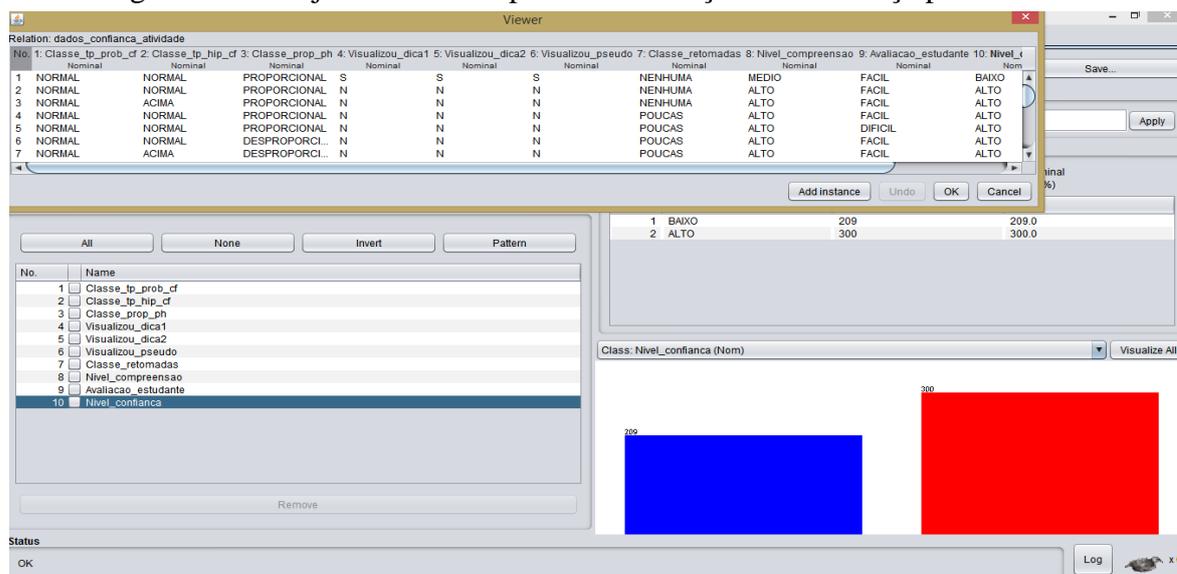
- Proporção de tempo das duas primeiras etapas em relação a escrita do código (classe_prop_ph): utiliza-se a mesma classificação definida para a identificação do esforço, indicando se o tempo é “PROPORCIONAL” ou “DESPROPORCIONAL”.
- Acesso à dica 1 (visualizou_dica1): indica se o aluno acessou a primeira dica (S) ou não (N).
- Acesso à dica 2 (visualizou_dica2): indica se o aluno acessou a segunda dica (S) ou não (N).
- Acesso ao pseudocódigo (visualizou_pseudo): indica se o aluno acessou o pseudocódigo (S) ou não (N).
- Nível de compreensão (classe_nivel_compreensao): avaliação da compreensão demonstrada pelo aluno, a partir da descrição dos pontos-chave e da hipótese. É utilizado o mesmo o padrão de classificação, alto, médio e baixo, que é empregado na identificação do esforço. É uma avaliação qualitativa do resultado da análise e descrição do problema e da hipótese e um nível mais baixo de compreensão pode indicar um grau de confiança menor do estudante.
- Avaliação do grau de dificuldade feito pelo estudante (classe_avaliacao_aluno): resposta do estudante para cada atividade, indicando se achou fácil, médio ou difícil.
- Número de retomadas (classe_retomadas): indica se forma muitas, poucas ou nenhuma retomada.

5.2.2 Reconhecimento da confiança por atividade: análise exploratória

Após definir as variáveis que seriam utilizadas para a confiança, o passo seguinte consistiu em uma análise exploratória para encontrar relações entre as variáveis no conjunto de dados e subsídios para uma classificação a priori, de forma análoga ao que foi feito no reconhecimento do esforço por atividade. Os dados associados à confiança de todas as atividades (509 registros) foram convertidos em um arquivo do tipo arff para uso na ferramenta Weka. A figura 53 mostra o arquivo carregado na ferramenta e uma amostra dos dados e valores dos atributos.

A exploração do conjunto de dados seguiu o mesmo roteiro utilizado no reconhecimento do esforço por atividade, com a aplicação do método de agrupamento, seguido do algoritmo para extração de regras. Os resultados mostrados na figura 54 mostram informações sobre os três grupos gerados pelo algoritmo SimpleKmeans.

Figura 53 – Conjunto de dados para determinação da confiança por atividade



Fonte: Elaborado pelo autor

É possível perceber uma concentração de registros nos grupos 1 e 2 que apresentam os valores associados a uma maior confiança, especialmente para os atributos de visualização das dicas e pseudocódigo (N). Os centroides do grupo 0 apresentam dados que apontam para uma menor confiança, o que é percebido também no nível de compreensão (BAIXO ou MEDIO) e na avaliação do estudante (DIFICIL ou MEDIO).

Figura 54 - Resultados da aplicação do SimpleKmeans nos dados relacionados à confiança por atividade

```
Number of iterations: 3
Within cluster sum of squared errors: 1058.0

Initial starting points (random):

Cluster 0: NORMAL, NORMAL, PROPORCIONAL, S, S, S, POUCAS, MEDIO, FACIL, BAIXO
Cluster 1: NORMAL, NORMAL, DESPROPORCIONAL, N, N, N, NENHUMA, BAIXO, MEDIO, ALTO
Cluster 2: NORMAL, NORMAL, PROPORCIONAL, N, N, N, NENHUMA, ALTO, MEDIO, ALTO

Missing values globally replaced with mean/mode

Final cluster centroids:

Attribute                Full Data          Cluster#
                          (509.0)           0              1              2
                          (74.0)           (202.0)        (233.0)

=====
Classe_tp_prob_cf        NORMAL             NORMAL          NORMAL          NORMAL
Classe_tp_hip_cf        NORMAL             NORMAL          NORMAL          NORMAL
Classe_prop_ph          PROPORCIONAL      PROPORCIONAL   DESPROPORCIONAL PROPORCIONAL
Visualizou_dica1        N                 S              N              N
Visualizou_dica2        N                 S              N              N
Visualizou_pseudo       N                 S              N              N
Classe_retomadas        NENHUMA           POUCAS         NENHUMA         NENHUMA
Nivel_compreensao       MEDIO             MEDIO          BAIXO           MEDIO
Avaliacao_estudante     MEDIO             DIFICIL        MEDIO           FACIL
Nivel_confianca         ALTO              BAIXO          BAIXO           ALTO
```

Fonte: Elaborado pelo autor

A alta concentração de registros em clusters com valores de atributos semelhantes apontou para uma tendência de classificação em duas classes, o que já havia sido percebido e adotado no reconhecimento do esforço por atividade. Esta distribuição decorre ainda do fato de que grande parte das entradas possui valores normais nos primeiros atributos e iguais a não (N) nos acessos às opções de ajuda.

Durante a aplicação do método Apriori para extração de regras foram adotados diferentes parâmetros, como um número maior de regras (10 a 50), um grau de confiança mínimo de 0,90 e posteriormente de 0,80, além de tentativas de remover um ou mais atributos do conjunto durante os testes. As relações em sua maioria foram estabelecidas entre os atributos que indicam a visualização dos recursos de ajuda e, em alguns casos as classes para os tempos (figura 55).

Tais resultados contribuíram para demonstrar que os acessos as dicas geralmente ocorrem simultaneamente, ou seja, quem acessa uma dica, também acessa a outra ou mesmo o pseudocódigo. Sobre os demais atributos não foi possível estabelecer muitas conclusões derivadas das regras extraídas, em função de poucos deles terem aparecido nas regras mais frequentes.

Figura 55 - Resultados da aplicação do Apriori nos dados relacionados à confiança por atividade

Best rules found:

```

1. Visualizou_dical=N Visualizou_pseudo=N 396 ==> Visualizou_dica2=N 390 <conf:(0.98)> lift:(1.14) lev:(0.09) [46] conv:(7.54)
2. Visualizou_dical=N 431 ==> Visualizou_dica2=N 422 <conf:(0.98)> lift:(1.13) lev:(0.1) [48] conv:(5.75)
3. Visualizou_pseudo=N 418 ==> Visualizou_dica2=N 404 <conf:(0.97)> lift:(1.12) lev:(0.08) [41] conv:(3.72)
4. Visualizou_dica2=N Visualizou_pseudo=N 404 ==> Visualizou_dical=N 390 <conf:(0.97)> lift:(1.14) lev:(0.1) [48] conv:(4.17)
5. Visualizou_dica2=N 442 ==> Visualizou_dical=N 422 <conf:(0.95)> lift:(1.13) lev:(0.1) [48] conv:(3.26)
6. Visualizou_pseudo=N 418 ==> Visualizou_dical=N 396 <conf:(0.95)> lift:(1.12) lev:(0.08) [42] conv:(2.82)
7. Visualizou_pseudo=N 418 ==> Visualizou_dical=N Visualizou_dica2=N 390 <conf:(0.93)> lift:(1.13) lev:(0.09) [44] conv:(2.49)
8. Visualizou_dical=N Visualizou_dica2=N 422 ==> Visualizou_pseudo=N 390 <conf:(0.92)> lift:(1.13) lev:(0.09) [44] conv:(2.31)
9. Visualizou_dical=N 431 ==> Visualizou_pseudo=N 396 <conf:(0.92)> lift:(1.12) lev:(0.08) [42] conv:(2.16)
10. Visualizou_dica2=N 442 ==> Visualizou_pseudo=N 404 <conf:(0.91)> lift:(1.12) lev:(0.08) [41] conv:(2.04)

```

Fonte: Elaborado pelo autor

Embora tenham sido desenvolvidas diversas atividades e testes para exploração dos dados para o reconhecimento da confiança, apenas as principais considerações e resultados foram apresentados. Muitas das conclusões e testes foram semelhantes aos apresentados no reconhecimento do esforço e por isso não foram repetidos nesta seção.

5.2.3 Reconhecimento da confiança por atividade: construção e treinamento das redes bayesianas

Considerando os resultados da análise exploratória e a estratégia usada no reconhecimento do esforço por atividade, a elaboração da rede para determinação da confiança para cada atividade incluiu também a elaboração de uma tabela para classificação a priori dos

dados que foram usados no processo de treinamento. Novamente foram adotadas duas classes possíveis, alto ou baixo, o que permitiu uma padronização na representação para os dois fatores, o esforço e a confiança. A tabela 21 mostra as pontuações finais, usadas para determinação do nível de confiança.

Tabela 21 – Pontuações para determinação da classe correspondente à confiança por atividade

Variável	Regra e pontuação
Classe_tp_prob_cf	NORMAL=1; ACIMA=-1
Classe_tp_hip_cf	NORMAL=1; ACIMA=-1
Classe_prop_ph	PROPORCIONAL = 1; DESPROPORCIONAL = -1
Visualizou_dica1	SIM=-1; NÃO=1
Visualizou_dica2	SIM=-1; NÃO=1
Visualizou_pseudo	SIM= -2; NÃO = 1
Classe_nivel_compreensao	BAIXO = -1; MÉDIO = 0; ALTO = 1
Classe_avaliacao_aluno	FÁCIL=1;MÉDIO=0;DIFÍCIL=-1
Classe_retomadas	NENHUMA=1;POUCAS=-1;MUITAS=-2

Fonte: Elaborado pelo autor

Alguns atributos, como o acesso ao pseudocódigo e a avaliação do estudante receberam um peso maior e, portanto, influenciaram mais a determinação da confiança. Assim como no modelo do esforço por atividade, a tabela é aplicada a cada uma das entradas e para cada uma delas é calculado o total de pontos, de acordo com o valor das variáveis. A regra geral para classificar, a priori cada um dos registros é a seguinte: se o valor é superior a 1, categoriza-se como alto, e nos demais casos, será considerado como nível baixo.

Com a determinação da classe para cada entrada do conjunto de dados de treinamento, foi possível aplicar o método *BayesNet* para avaliar as métricas e a matriz de confusão resultante do treinamento. Cabe salientar novamente que a análise destes resultados contribuiu para indicar a viabilidade de adotar uma estratégia baseada na classificação supervisionada, o que já havia sido verificado no reconhecimento do esforço. Os resultados da aplicação do algoritmo são mostrados na figura 56.

Figura 56 - Resultados da classificação da confiança por atividade com 2 classes

```

Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances      479           94.1061 %
Incorrectly Classified Instances    30            5.8939 %
Kappa statistic                    0.8768
Mean absolute error                 0.1257
Root mean squared error             0.218
Relative absolute error             25.9654 %
Root relative squared error         44.3223 %
Total Number of Instances          509

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0,890   0,023   0,964     0,890   0,925     0,879   0,989    0,985    BAIXO
                0,977   0,110   0,927     0,977   0,951     0,879   0,989    0,992    ALTO
Weighted Avg.   0,941   0,074   0,942     0,941   0,941     0,879   0,989    0,989

=== Confusion Matrix ===

  a  b  <-- classified as
186 23 |  a = BAIXO
 7 293 |  b = ALTO

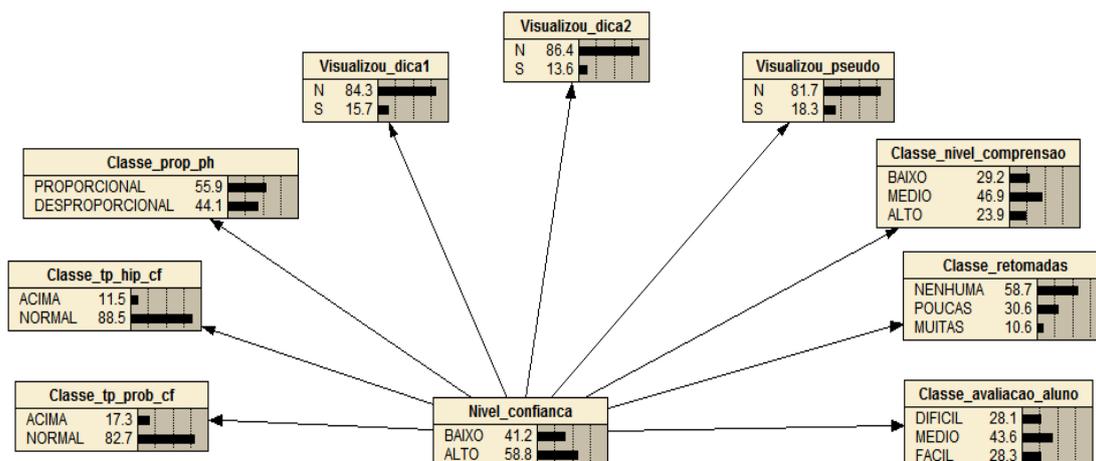
```

Fonte: Elaborado pelo autor

A acuracidade foi satisfatória (94%), bem como as demais medidas, como, por exemplo, a precisão para baixo (0,964) e alto (0,927). As taxa de verdadeiros positivos (TP) e *recall* indicam uma dificuldade um pouco maior para determinar os casos de baixa confiança, o que deriva do fato de que existem menos entradas para esta classe (209). Além disso, os valores associados à baixa confiança na maioria das variáveis também aparecem com menor frequência. A dificuldade para determinar uma baixa confiança não invalida o modelo, embora demonstre a necessidade de refazer o processo de treinamento das redes bayesianas em trabalhos futuros.

O passo seguinte consistiu em desenhar e efetuar o treinamento da rede bayesiana no software Netica. A arquitetura da rede proposta para a confiança por atividade é mostrada na figura 57. A arquitetura proposta segue os princípios da rede para reconhecimento do esforço, com dependências apenas da variável do nível de confiança. Da mesma forma que ocorreu nos testes do modelo para o esforço, a aplicação do método *BayesNet* no Weka também gerou a estrutura apresentada na figura 57, o que caracteriza um padrão para a classificação nos dois modelos.

Figura 57 - Rede Bayesiana para o reconhecimento da confiança por atividade



Fonte: elaborado pelo autor

Na figura 57 são exibidos os percentuais das ocorrências dos dados utilizados no processo de treinamento da rede, sendo que a grande maioria dos registros foi classificada com nível alto de confiança (62,2%). Uma das características verificadas nas variáveis usadas para identificar a confiança é a concentração de valores em alguns atributos, nos quais determinadas categorias ocorrem em mais de 80% das entradas (visualizou_dica1, visualizou_dica2 e visualizou_pseudo).

Figura 58 – Tabelas de probabilidades da rede bayesiana para o reconhecimento da confiança por atividade

Node	Chance	% Probability	Reset	Close
Classe_tp_prob_cf Table (in Bayes net reco...)	Classe_tp_prob_cf	% Probability	Reset	Close
Nivel_confianza	ACIMA	NORMAL		
BAIXO	35.849	64.151		
ALTO	4.305	95.695		
Classe_tp_hip_cf Table (in Bayes net reco...)	Classe_tp_hip_cf	% Probability	Reset	Close
Nivel_confianza	ACIMA	NORMAL		
BAIXO	18.868	81.132		
ALTO	6.291	93.709		
Classe_prop_ph Table (in Bayes net reco...)	Classe_prop_ph	% Probability	Reset	Close
Nivel_confianza	PROPORCIONAL	DESPROPORC...		
BAIXO	31.604	68.396		
ALTO	72.848	27.152		
Visualizou_dica1 Table (in Bayes net reco...)	Visualizou_dica1	% Probability	Reset	Close
Nivel_confianza	N	S		
BAIXO	65.566	34.434		
ALTO	97.351	2.649		
Visualizou_dica2 Table (in Bayes net reco...)	Visualizou_dica2	% Probability	Reset	Close
Nivel_confianza	N	S		
BAIXO	69.34	30.66		
ALTO	98.344	1.656		
Visualizou_pseudo Table (in Bayes net reco...)	Visualizou_pseudo	% Probability	Reset	Close
Nivel_confianza	N	S		
BAIXO	58.491	41.509		
ALTO	98.013	1.987		
Classe_nivel_compensao Table (in Bayes n...)	Classe_nivel_compen...	% Probability	Reset	Close
Nivel_confianza	BAIXO	MEDIO	ALTO	
BAIXO	46.009	43.192	10.798	
ALTO	17.492	49.505	33.003	
Classe_retomadas Table (in Bayes net re...)	Classe_retomadas	% Probability	Reset	Close
Nivel_confianza	NENHUMA	POUCAS	MUITAS	
BAIXO	46.009	31.925	22.066	
ALTO	67.657	29.703	2.64	
Classe_avaliacao_aluno Table (in Bayes net ...)	Classe_avaliacao_aluno	% Probability	Reset	Close
Nivel_confianza	DIFICIL	MEDIO	FACIL	
BAIXO	51.643	39.437	8.92	
ALTO	11.551	46.535	41.914	

Fonte: Elaborado pelo autor

Uma análise das tabelas de probabilidades, obtidas no treinamento, mostradas na figura 58 indica que a visualização das dicas, pseudocódigo, tempos acima do normal e um número alto de retomadas apresentam baixa associação com altos níveis de confiança. Isso ocorre

devido ao número reduzido de entradas com estas características, o que enfatiza a ideia de que interromper a atividade, acessar frequentemente opções de ajuda são indicativos de baixa confiança, especialmente quando aparecem combinados nos mesmos registros.

O atributo que indica o nível de compreensão apresenta um maior equilíbrio na distribuição das probabilidades, o que indica uma menor influência na confiança, se comparado ao modelo para determinação do esforço por atividade. Sobre a avaliação do estudante, como era de se esperar, os casos em que o exercício é considerado fácil possuem probabilidade reduzida para baixa confiança e de forma oposta, uma avaliação difícil apresenta pouca chance de aparecer associada e um alto nível de confiança.

A inserção desta variável se mostrou importante para verificar se a percepção que o estudante tem da atividade está de acordo com os dados derivados do seu comportamento na resolução do exercício. Uma situação em que as demais variáveis apontam para baixa confiança e este atributo apresenta valor correspondente a fácil, por exemplo, pode acarretar em uma indicação que a confiança não é totalmente baixa, nem alta. Cabe ressaltar que esta é uma das principais justificativas e vantagens do uso da abordagem bayesiana, a possibilidade de indicar uma classe com um fator de certeza.

Os modelos apresentados até aqui descrevem a visão por atividade, contribuindo para que o professor possa acompanhar cada exercício e o estado afetivo do estudante durante a realização das tarefas. É possível comparar, por exemplo duas atividades ou ainda verificar se baixos níveis são frequentes ou mais comuns em determinados exercícios ou estudantes.

Em um contexto diferente, é necessário reconhecer a motivação de cada estudante ao longo das atividades, o que implica em uma síntese do que o estudante fez e como se comportou durante os exercícios desenvolvidos. Na visão por estudante também foram desenvolvidos dois modelos específicos, um para o reconhecimento do esforço e outro para identificar o nível de confiança.

5.3 Reconhecimento do esforço geral do estudante.

O contexto relacionado ao estudante baseia-se no princípio de que a motivação geral do estudante emerge a partir dos dados coletados durante as atividades, sendo necessário estabelecer uma forma de resumir ou agrupar os valores obtidos em cada exercício, por discente. A definição do conjunto de atributos que permitiu identificar o esforço do estudante na disciplina passou pelo resumo ou síntese dos valores dos atributos em cada uma das atividades e pela análise novas variáveis relacionadas ao número de atividades não realizadas e a semelhança dos conteúdos postados.

Uma questão essencial no contexto do aluno é a possibilidade de classificar o esforço e a confiança a qualquer momento, independentemente do número de exercícios que foram propostos pelo professor e finalizados pelos estudantes. Além disso, na medida em que a disciplina evolui, o professor terá proposto mais atividades e alguns alunos terão finalizado todas, enquanto outros apenas parte das tarefas e espera-se que em qualquer situação o modelo seja eficiente para determinar os fatores da motivação.

Uma das primeiras possibilidades avaliadas para o reconhecimento da motivação por estudante foi a adoção de medidas estatísticas simples, tais como, contagens de valores, somas, médias como forma de obter pontuações por variável. A partir das pontuações seria possível classificar o nível geral de esforço e confiança, não sendo necessário neste caso usar métodos de classificação supervisionada, como as redes bayesianas, por exemplo. Entretanto, esta estratégia implicaria no uso apenas das variáveis já apresentadas no contexto da atividade, dificultando a inserção de novos atributos, além de não manter o padrão de classificação probabilística, que permite determinar o esforço com um grau de certeza associado.

Com o objetivo de adotar o mesmo padrão dos modelos específicos por atividade e permitir a inserção de novos atributos, a análise do perfil do estudante não se resumiu à aplicação de medidas estatísticas, tais como percentual ou número de atividades cujo tempo ou o detalhe foi classificado como alto ou baixo. Uma simples análise estatística sobre os dados das atividades realizadas desconsideraria, por exemplo, tarefas não entregues, o que é um indicativo importante do esforço ao longo do tempo.

Além disso, uma avaliação dos conteúdos coletados mostrou que alguns estudantes tendem a postar descrições para o problema e para a hipótese muito semelhantes em diferentes atividades, o que pode indicar que ele está usando a mesma solução, fazendo pequenas adaptações, para diferentes problemas. Embora esta forma de expressão não possa ser totalmente desqualificada, um grau de semelhança muito alto pode indicar um menor esforço e deve ser levado em consideração no reconhecimento do nível geral de esforço.

As próximas seções descrevem a síntese dos valores para gerar os atributos usados no reconhecimento da motivação geral do estudante, além da elaboração das redes bayesianas. Nesta etapa não foram utilizados métodos de agrupamento ou extração de regras sobre os dados agrupados, uma vez que os atributos empregados foram derivados das variáveis já descritas na identificação da motivação por atividade. Sendo assim, a relação entre estes, as possíveis dependências entre as variáveis já eram conhecidas, o que contribuiu para descartar a necessidade de uma análise exploratória.

5.3.1 Reconhecimento do esforço por estudante: síntese das variáveis relacionadas às atividades

A maioria das variáveis usadas no reconhecimento do esforço associado a cada atividade apresenta valores categorizados como alto, médio ou baixo. Tal padrão é observado tanto nos tempos, quanto nos níveis de detalhe e na compreensão, o que permite a utilização de um método padrão para resumir os dados por estudante, para estes atributos.

A análise dos valores e a síntese deve levar em consideração o número de atividades realizadas e deve ser robusta o suficiente para que seja possível aplicar o modelo em qualquer contexto, independente do número de exercícios realizados. Não importa que tenha sido feitas duas, cinco, dez ou mais atividades, deve ser possível sintetizar as variáveis e aplicar o modelo de forma consistente. Tal requisito é fundamental para permitir que o professor acompanhe os níveis de motivação a qualquer momento, de forma constante, verificando, por exemplo se as ações ou intervenções feitas por estão contribuindo para aumentar o nível de motivação ao longo do tempo.

Considerando as necessidades expostas, foi definido um método baseado na proporção de cada categoria ao longo das atividades, com uma atribuição de um peso ou pontuação para cada valor, sendo 0 para a classe baixo, 2 para médio e 3 para alto. A seguir é calculada a pontuação máxima possível, de acordo com o número de exercícios publicados pelo professor. Supondo que tenham sido publicada 5 atividades, a pontuação máxima seria 15, se o estudante tivesse apresentado tempos sempre altos. A pontuação real obtida e a proporção em relação à pontuação máxima são determinadas e indicam o valor da categoria do atributo.

Para os atributos usados no modelo para determinar o esforço do estudante foram definidas quatro categorias possíveis para os atributos temporais e níveis de detalhe, “BAIXO”, “MEDIO_BAIXO”, “MEDIO_ALTO” e “ALTO”. O nível médio foi dividido em duas categorias, em função da necessidade de verificar se predominância foi de médio com tendência de baixa ou de alta. Em boa parte dos casos, estudantes apresentam várias vezes o nível médio como predominante na atividade, mas este vem acompanhado de baixos ou altos em outras, o que demonstra que existe uma tendência, para cima ou para baixo.

A categorização segue a seguinte regra: percentual menor ou igual a 25% é considerado baixo, entre 25 e 50% (excluindo o limite) é medio_baixo, entre 50 (inclusive) e 75, medio_alto e acima de 75% é alto. A tabela 22 mostra o exemplo de um estudante que finalizou 11 atividades. Cada linha da tabela exhibe os dados de uma atividade resolvida pelo aluno, com valores para quatro variáveis.

Na penúltima linha foram mostradas as fórmulas usadas para calcular a proporção de cada atributo. Observa-se, por exemplo, que no tempo da hipótese a proporção foi inferior a 0,25, acarretando na categorização como baixo, enquanto no nível de detalhe da hipótese, o valor de 0,78 foi classificado como alto.

Inicialmente o valor proposto para a categoria baixo era 1, entretanto, utilizando este valor nenhum dos estudantes apresentava categoria geral baixo para os atributos, pois ao somar um terço dos pontos (33%) a categorização já era definida como media_baixa. Optou-se por diferenciar mais os casos de baixo esforço dos demais, zerando a pontuação. Com isso, os casos que o baixo esforço predomina, mas que em alguns exercícios o aluno demonstrou um médio esforço já passam a não ser definidos simplesmente como baixo e sim, com um médio com tendência de baixo.

Tabela 22 – Síntese das variáveis para o reconhecimento do esforço por estudante

Tempo do problema classe_tp_prob_ef		Tempo da hipótese classe_tp_hip_ef		Detalhe do problema classe_nivel_detalhe_prob		Detalhe da hipótese classe_nivel_detalhe_hip	
Classe atividade	Pontuação	Classe atividade	Pontuação	Classe atividade	Pontuação	Classe atividade	Pontuação
MEDIO	2	MEDIO	2	BAIXO	0	ALTO	3
ALTO	3	MEDIO	2	BAIXO	0	ALTO	3
MEDIO	2	BAIXO	0	MEDIO	2	MEDIO	2
MEDIO	2	BAIXO	0	BAIXO	0	ALTO	3
MEDIO	2	BAIXO	0	MEDIO	2	ALTO	3
MEDIO	2	BAIXO	0	MEDIO	2	MEDIO	2
MEDIO	2	BAIXO	0	MEDIO	2	MEDIO	2
ALTO	3	BAIXO	0	MEDIO	2	MEDIO	2
BAIXO	0	BAIXO	0	MEDIO	2	MEDIO	2
MEDIO	2	BAIXO	0	MEDIO	2	MEDIO	2
MEDIO	2	BAIXO	0	BAIXO	0	MEDIO	2
22/33 = 0,66		4/33 = 0,12		14/33 = 0,42		26/33 = 0,78	
MEDIO_ALTO		BAIXO		MEDIO_BAIXO		ALTO	

Fonte: Elaborado pelo autor

Ao analisar o primeiro atributo (tempo do problema) da tabela, é possível perceber que 8 das 11 atividades apresentaram tempo médio, o que representa aproximadamente 72% dos casos, enquanto em torno de 18% foram altos e 10% baixos. Constata-se que o esforço geral é médio com uma leve tendência de alto, portanto, foi classificado como medio_alto. Em outras situações, a tendência para o alto ou baixo é mais clara, como no detalhe do problema que apresenta em torno de 60% de médios e o restante de baixos, caracterizando um nível médio com clara tendência de baixa.

O caso do tempo da hipótese apresenta um nível geral considerado baixo, pois em 80% das atividades este estudante apresentou tempos que ficaram no primeiro quartil, ou seja, tempo menor ou igual aos 25% menores. Isso demonstra que o tempo dedicado à elaboração da hipótese na grande maioria das atividades é inferior aos demais colegas, o que neste caso contribui para a determinação de um baixo esforço.

Os atributos que indicam se foi acessado o pseudocódigo e se o código foi executado apresentam valores do tipo sim ou não em cada atividade. Neste caso também foi adotado um cálculo baseado na proporção, no qual é feita uma contagem de exercícios com valor igual a sim, dividindo o resultado pelo número total de atividades finalizadas. Os dois atributos para o modelo específico do estudante apresentam três classes, nenhuma, poucas, muitas, que representam o número de vezes que ocorreu acesso ao pseudocódigo e quantas vezes foi feita a execução do código.

As regras para enquadramento do pseudocódigo são: percentual de vezes que foi acessado igual a 0, “NENHUMA”, percentual superior a 0 e inferior a 50%, “POUCAS” e acima de 50%, “MUITAS”. Neste caso é possível considerar que o ideal seria nenhum acesso e que uma contagem superior a metade das atividades (50%) é excessivo e pode indicar menor esforço.

Quanto ao atributo de execução do código, o desejado sob o ponto de vista do esforço é que o estudante sempre finalize a atividade, com, pelo menos uma execução. Desta forma utilizou-se a seguinte regra: percentual de vezes que foi executado igual a 0, “NENHUMA”, percentual superior a 0 e inferior a 80%, “POUCAS” e acima de 80%, “MUITAS”. A diferença em relação ao pseudocódigo é que nesta variável a classe que corresponde a muitas vezes representa o caso mais positivo.

O número de atividades não realizadas ou realizadas parcialmente é um atributo fundamental para definir o esforço geral do estudante. A determinação da classe foi feita usando a seguinte regra: se mais de 30% das atividades não foram finalizadas, a categoria foi definida como “MUITAS”, se todas foram finalizadas, o valor é correspondente a “NENHUMA”, caso contrário é definido como “POUCAS”.

A análise das respostas dos estudantes durante o estudo piloto indicou que em muitas situações as aoluções entregues pelos estudantes eram muito parecidas para todos os problemas, especialmente na descrição dos pontos-chave e da hipótese. A partir desta constatação foi adicionado uma nova variável que indica a semelhança das respostas de cada estudante ao longo

das atividades. A categorização seguiu o padrão alto, médio e baixo usado em grande parte dos atributos nos demais modelos.

A seguir é apresentado um resumo das variáveis usadas no reconhecimento do esforço geral demonstrado pelo estudante.

- Tempo para definição dos pontos-chave do problema (*classe_tp_prob_ef*): proporção dos tempos para descrição dos pontos-chave do problema. Classificado como “BAIXO”, “MEDIO_BAIXO”, “MEDIO_ALTO” ou “ALTO”.
- Tempo para elaboração da hipótese (*classe_tp_hip_ef*): proporção dos tempos para descrição da hipótese. Tanto esta, quanto a próxima variável utilizam os mesmos valores de classe da anterior.
- Tempo para o código fonte (*classe_tp_cod_ef*): tempos dedicados ao desenvolvimento do código fonte.
- Nível de detalhe do problema (*classe_nivel_detalhe_prob*): os níveis de detalhe e de compreensão seguem a definição de classes usadas nos tempos.
- Nível de detalhe da hipótese (*nivel_detalhe_hip*).
- Nível de compreensão (*classe_nivel_compreensao*).
- Número de execuções (*classe_exec_cod*): categorizado como nenhuma, pouca ou muitas. Considera o atributo que indica se foi executado (S) ou não (N).
- Número de acessos ou visualizações do pseudocódigo (*classe_vis_pseudo*): quantas vezes foi acessado o pseudocódigo, categorizado da mesma forma que o número de execuções.
- Atividades não realizadas (*classe_ativ_ao_realizadas*): proporção (muitas, poucas ou nenhuma) de tarefas não resolvidas.
- Semelhança entre as definições do problema e hipótese (*grau_semelhanca*): avaliação feita pelo professor, que indica se houve um alto, medio ou baixo grau de semelhança, nos conteúdos das atividades resolvidas.

5.3.2 Reconhecimento do esforço por estudante: construção das redes bayesianas

Nesta etapa seguiu-se os mesmos passos adotados na elaboração das redes bayesianas dos modelos por atividade, com uma determinação inicial da classe de cada uma das entradas, a aplicação do método *BayesNet* na ferramenta Weka, seguida da construção e treinamento da rede. Cada registro nos arquivos de treinamento correspondem aos dados de um estudante.

O número de registros disponíveis para o processo de treinamento, foi 68, o que corresponde ao número total de participantes do estudo piloto e estudo de caso. Embora o

número de registros para o treinamento possa ser considerado reduzido, os mesmos permitiram obter um conjunto de probabilidades e um modelo inicial, que pode ser aperfeiçoado futuramente com um treinamento baseado em um número maior de participantes. Esta é uma característica importante dos modelos propostos nesta pesquisa, a possibilidade de recriar as tabelas de probabilidades, refazer o processo de treinamento das redes a partir da disponibilidade de novos dados.

Para determinar a classe de cada entrada para o aprendizado da rede, foram atribuídos pontos aos valores de cada atributo (tabela 23). Após calcular a pontuação total do estudante, foi aplicada a seguinte regra: se a soma for menor ou igual a 0 (zero) o nível de esforço é baixo, caso contrário, é alto.

Tabela 23 - Pontuações para determinação da classe correspondente ao esforço por atividade

Variável	Regra e pontuação
Classe_tp_prob_ef	BAIXO = -2; MEDIO_BAIXO = -1; MEDIO_ALTO=1 ; ALTO = 2
Classe_tp_hip_ef	BAIXO = -2; MEDIO_BAIXO = -1; MEDIO_ALTO=1 ; ALTO = 2
Classe_tp_cod_ef	BAIXO = -2; MEDIO_BAIXO = -1; MEDIO_ALTO=1 ; ALTO = 2
Classe_vis_pseudo	NENHUMA=1; POUCAS=0;MUITAS=-1
Classe_exec_cod	NENHUMA=-1; POUCAS=0;MUITAS=1
Classe_nivel_detalhe_prob	BAIXO = -2; MEDIO_BAIXO = -1; MEDIO_ALTO=1 ; ALTO = 2
Classe_nivel_detalhe_hip	BAIXO = -2; MEDIO_BAIXO = -1; MEDIO_ALTO=1 ; ALTO = 2
Classe_nivel_compreensao	BAIXO = -2; MEDIO_BAIXO = -1; MEDIO_ALTO=1 ; ALTO = 2
Classe_ativ_nao_realizadas	NENHUMA=1; POUCAS=-1;MUITAS=-3
Grau_semelhanca	ALTO=-1; MEDIO=0;BAIXO=1

Fonte: Elaborado pelo autor

As variáveis associadas aos tempos e nível de detalhe tem o mesmo impacto na identificação do esforço do estudante, mas o número de atividades não realizadas é decisivo, por isso, o peso deste atributo é maior, ao deixar de realizar algumas atividades o estudante demonstra claramente um menor nível de esforço. Se forem diversas atividades se pode concluir

que o esforço é menor, mesmo que durante os exercícios feitos ele tenha demonstrado maior dedicação.

Analisando as métricas para classificação supervisionada mostrados na figura 59, é possível perceber que o método *BayesNet* apresenta boa capacidade de classificar corretamente o conjunto de entradas, tanto nos casos do nível alto, quanto baixo. Além da acuracidade que foi superior a 95%, os indicadores *recall* e *precision* apresentam valores altos, o que confere um alto grau de robustez ao modelo.

Figura 59 – Resultados da aplicação do método BayesNet para determinar o esforço por estudante

```

Correctly Classified Instances      65          95.5882 %
Incorrectly Classified Instances    3           4.4118 %
Kappa statistic                    0.9094
Mean absolute error                0.0484
Root mean squared error            0.1683
Relative absolute error             9.9762 %
Root relative squared error        34.1725 %
Total Number of Instances         68

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall  F-Measure  MCC      ROC Area  PRC Area  Class
                0,950   0,036   0,974     0,950   0,962     0,910   0,996    0,997    BAIXO
                0,964   0,050   0,931     0,964   0,947     0,910   0,996    0,996    ALTO
Weighted Avg.   0,956   0,042   0,957     0,956   0,956     0,910   0,996    0,997

=== Confusion Matrix ===

  a  b  <-- classified as
38  2  |  a = BAIXO
 1 27  |  b = ALTO

```

Fonte: Elaborado pelo autor

A estrutura da rede bayesiana (Figura 71), elaborada na ferramenta Netica segue o padrão mostrado no reconhecimento por atividade, com dependência apenas entre a variável do nível de esforço para as demais. Além disso, o atributo que determina o nível de esforço segue a mesma categorização (“ALTO”, “BAIXO”) usada no reconhecimento do esforço em cada atividade.

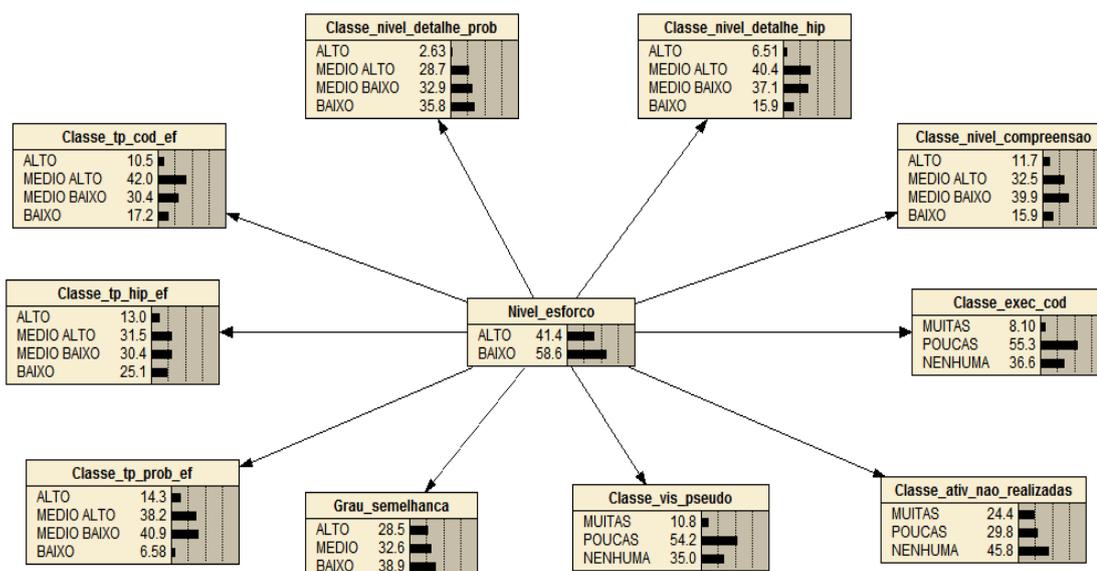
É possível verificar que em algumas classes há categorias que aparecem mais raramente, como no caso dos níveis de detalhamento alto e a classe de muitas execuções de código. Tal situação foi verificada também nas redes para reconhecimento da motivação por atividade, embora com menos intensidade. O problema do desequilíbrio de classes é discutido por diversos autores, como Tan (2009) que argumenta que distribuições de dados em classes desequilibradas são comuns em muitas aplicações reais. O autor apresenta exemplos como o número de defeitos em produtos que podem ocorrer muito raramente em uma empresa ou ainda fraudes em cartões cuja ocorrência é de 1 em 100. Nestes casos mesmo que poucas entradas

apresentem as características indicadas, esta informação é relevante para o método de aprendizado de máquina e para o modelo que será obtido.

Em alguns situações as medidas de precisão que avaliam os classificadores podem não se mostrar plenamente adequadas, sendo necessário ajustar as mesmas e o modelo para lidar com as particularidades verificadas nos dados (TAN, 2009). No caso de um modelo probabilístico é necessário verificar as probabilidades associadas, com o objetivo de analisar a influência de cada um dos valores da classe no processo de classificação.

No contexto do trabalho, mesmo que a ocorrência de determinada classe seja rara, ela pode ter influência na determinação do nível de confiança ou esforço, além disso o uso de uma quantidade maior de registros, em trabalhos futuros pode alterar o contexto verificado, com uma incidência maior das classes com baixa representação nos dados usados no treinamento. Diferentes grupos de alunos podem exibir perfis diversos o que pode gerar uma mudança na incidência das categorias o que justifica a manutenção das classes citadas.

Figura 60 – Rede bayesiana para reconhecimento do esforço do estudante



Fonte: Elaborado pelo autor

Na figura 60 são exibidas as probabilidades a priori, obtidas no processo de treinamento da rede, onde é possível verificar uma concentração nas classes que indiquem níveis médio alto e médio baixo. No caso das variáveis de execução do código, visualização e de atividades não realizadas, a maioria dos valores indica poucas ou nenhuma ocorrência. A distribuição para o grau de confiança é mais equilibrada, aproximadamente um terço dos estudantes em cada uma das categorias. Em relação ao nível de esforço, observa-se um número maior de alunos com baixo esforço, o que foi verificado também nos dados do esforço por atividade.

Uma análise das tabelas de probabilidades mostradas na figura 61 indica que a variável correspondente ao nível de compreensão apresenta altas probabilidades nas categorias alto (21.87) e médio alto (68.75) para o nível alto de esforço, enquanto os valores baixo (25) e médio baixo (63.63) estão mais associados a um nível baixo de esforço, o que confirma a influência desta na determinação do esforço. Esta situação já havia sido verificada no reconhecimento por atividade, tendo permanecido no modelo geral do estudante.

Com relação aos atributos inseridos neste modelo, o grau de semelhança e o número de atividades não realizadas, percebe-se que a finalização de todas as atividades têm alta probabilidade para o nível de esforço alto (67.74), da mesma forma um baixo grau de semelhança (70.96). Em ambos os casos, os percentuais associados ao nível de esforço baixo são equilibrados, embora exista uma tendência de maior semelhança e muitas ou poucas atividades não realizadas nos registros de baixo esforço.

Figura 61 – Tabela de probabilidades da rede para reconhecimento do esforço do estudante

The figure displays eight probability tables from a Bayesian network. Each table shows the conditional probabilities for a specific node, with rows representing the effort level (ALTO or BAIXO) and columns representing the variable categories. The tables are as follows:

Node	ALTO	MEDIO AL...	MEDIO BA...	BAIXO
Classe_tp_cod_ef	15.625	53.125	18.75	12.5
Classe_nivel_detalle_prob	3.125	53.125	34.375	9.375
Classe_nivel_detalle_hip	12.5	71.875	12.5	3.125
Classe_nivel_compreensao	21.875	68.75	6.25	3.125
Classe_ativ_nao_realizadas	12.903	19.355	67.742	32.558
Classe_vis_pseudo	6.452	41.935	51.613	13.953
Grau_semelhanca	12.903	16.129	70.968	39.535
Classe_tp_prob_ef	31.25	34.375	28.125	6.25
Classe_tp_hip_ef	25	40.625	21.875	12.5

Fonte: Elaborado pelo autor

5.4 Reconhecimento da confiança geral do estudante

O modelo para o reconhecimento da confiança geral demonstrada pelo estudante é formado por atributos que estão relacionados às variáveis definidas para o reconhecimento da confiança em cada atividade. Da mesma forma que ocorre no reconhecimento do esforço do

estudante, os valores dos atributos são obtidos a partir de um resumo ou síntese do conjunto de valores associados a estes atributos nas atividades, seguindo o mesmo princípio do modelo geral do esforço do estudante.

5.4.1 Reconhecimento da confiança por estudante: síntese das variáveis relacionadas às atividades

As variáveis relacionadas à confiança em cada tarefa possuem em sua maioria, duas categorias, no caso dos tempos, cujos valores indicam se foi acima ou normal, se o total das duas primeiras etapas foi proporcional ou ainda valores do tipo sim ou não, no acesso às dicas. Em todos estes atributos foi adotada uma categorização padrão, com os valores resumidos indicando as categorias nenhuma, muitas ou poucas, correspondendo ao número de vezes que o tempo foi acima, quantas vezes foi desproporcional ou ainda o número de vezes que foi acessada a dica ou pseudocódigo.

Utiliza-se portanto uma contagem para as ocorrências de cada valor, calculando a proporção desta em relação ao número total de atividades finalizadas pelo aluno. Neste contexto, para indicar maior confiança é desejável que em nenhuma ou poucas vezes o tempo seja acima do normal, seja desproporcional e em nenhuma ou em poucas atividades ocorra acesso às dicas.

A regra geral para determinação da categoria dos atributos citados é a seguinte: se a proporção é superior a 50%, é definido como “MUITAS”, se é 0 (zero), o valor é correspondente a “NENHUMA” e nos demais casos, é atribuída a classe “POUCAS”. A exceção é a variável que indica acesso ao pseudocódigo, neste caso, se mais de 30% das vezes ocorreu o acesso já é considerado na categoria de muitos acessos, em função de ser fortemente indicado que o estudante evite olhar o pseudocódigo durante a realização da tarefa.

A abordagem descrita na seção 5.3.1 foi adotada também no modelo para confiança para definir o atributo que corresponde ao nível de compreensão para o modelo geral da confiança, desta forma, adota-se o mesmo padrão para sintetização dos dados.

A mesma estratégia, que associa uma pontuação a cada registro de atividade é usada igualmente para o atributo que representa a avaliação feita pelo estudante em cada exercício, na qual ele indica se foi fácil, médio ou difícil. No modelo geral da confiança esta variável possui as categorias: “BAIXO”, “MEDIO_BAIXO”, “MEDIO_ALTO” e “ALTO”, de forma análoga ao que ocorre com a variável do nível de compreensão. Uma ressalva sobre este atributo é que um número maior de avaliações do tipo fácil contribui para que a categoria seja alto ou

médio alto, enquanto, avaliações com conteúdo difícil levam a categorização como baixo ou médio baixo.

A síntese do atributo que corresponde ao número de retomadas, segue a mesma lógica do nível de compreensão e avaliação, com a seguinte pontuação: nenhuma=3; poucas=2; muitas=0. Desta forma quanto menos retomadas, mais registros de atividades sem retomadas, maior a chance de ocorrer um valor baixo ou médio baixo e de forma oposta, mais ocorrências de muitas retomadas acarretam em categorização com alto ou médio alto. Conclui-se portanto que um valor alto na categoria geral de retomadas representa mais retomadas e portanto uma menor confiança. Este atributo se constitui no único caso em que valores negativos são atribuídos às categorias alto ou médio alto (Tabela 24).

5.4.2 Reconhecimento da confiança por estudante: construção das redes bayesianas

Esta seção descreve o conjunto de atributos, e a rede bayesiana elaborada para determinar o nível de confiança do estudante. A tabela 24 descreve as variáveis e as pontuações utilizadas para uma classificação a priori do nível de confiança do estudante.

Tabela 24 - Pontuações para determinação da classe correspondente à confiança por atividade

Variável	Regra e pontuação
Classe_tp_prob_cf	NENHUMA=1;POUCAS=0;MUITAS=-1
Classe_tp_hip_cf	NENHUMA=1;POUCAS=0;MUITAS=-1
Classe_prop_ph_cf	NENHUMA=1;POUCAS=0;MUITAS=-1
Classe_vis_dica1	NENHUMA=1;POUCAS=0;MUITAS=-1
Classe_vis_dica2	NENHUMA=1;POUCAS=0;MUITAS=-1
Classe_vis_pseudo	NENHUMA=1;POUCAS=0;MUITAS=-1
Classe_nivel_compreensao	BAIXO = -2; MEDIO-BAIXO = -1; MEDIO_ALTO=1; ALTO = 2
Classe_avaliacao_aluno	BAIXO=-3; MEDIO_BAIXO=-2; MEDIO_ALTO=2; ALTO=3
Classe_retomadas	BAIXO = 2; MEDIO-BAIXO = 1; MEDIO_ALTO=-1; ALTO = -2

Fonte: Elaborado pelo autor

A pontuação adota é semelhante a tabela para reconhecimento do esforço do estudante, pois os atributos que possuem três valores possíveis apresentam pontuação variando de -2 a 2. No caso das variáveis categorizadas de acordo com a proporção das ocorrências dos valores, a pontuação varia de -1 a 1, com uma categoria neutra (poucas).

Embora possa causar certa estranheza as categorias do atributo que corresponde à avaliação do aluno, na qual ele indica que é difícil, médio ou fácil, a opção por manter as categorias dos demais atributos que vão de baixo a alto. Como citado anteriormente, categorias baixo e médio baixo indicam uma maior ocorrência de atividades consideradas difíceis por parte do estudante e categorias alto e médio alto estão associadas a exercícios considerados mais fáceis. Em virtude desta categorização, foram atribuídos pontos negativos (-2 e -3) aos valores baixos e positivos às categorias alto e médio alto (2 e 3). Esta variável foi considerada a mais decisiva para determinação do nível de confiança, por isso recebeu as maiores pontuações na tabela.

A soma da pontuação de cada uma das entradas é calculada e sobre o valor é aplicada uma regra que define como baixa confiança registros cuja soma é igual ou menor que 0 (zero), sendo os demais categorizados como altos. Com a classificação a priori, 41 dos 68 registros de estudantes foram classificados com alta confiança, ao contrário do que ocorreu no conjunto de treinamento usado para determinar o esforço dos estudantes.

Figura 62 – Resultados da aplicação do método BayesNet para determinar a confiança do estudante

```

Correctly Classified Instances      62          91.1765 %
Incorrectly Classified Instances    6           8.8235 %
Kappa statistic                    0.8157
Mean absolute error                0.1278
Root mean squared error            0.272
Relative absolute error            26.6242 %
Root relative squared error        55.5196 %
Total Number of Instances         68

=== Detailed Accuracy By Class ===

                TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                0,889   0,073   0,889     0,889   0,889     0,816   0,964    0,958    BAIXO
                0,927   0,111   0,927     0,927   0,927     0,816   0,964    0,974    ALTO
Weighted Avg.   0,912   0,096   0,912     0,912   0,912     0,816   0,964    0,968

=== Confusion Matrix ===

 a  b  <-- classified as
24  3  | a = BAIXO
 3 38  | b = ALTO

```

Fonte: Elaborado pelo autor

A capacidade e eficiência do método bayesiano para classificar as entradas é ilustrada na figura 62, a qual demonstra uma acuracidade superior a 91% após a utilização do algoritmo BayesNet no Weka. Embora o percentual seja considerado satisfatório, é o menor dentre aqueles que foram mostrados nos componentes anteriores, o que enfatiza que existe uma dificuldade um pouco maior para determinar a confiança do estudante. Novamente a classe baixo é a que apresenta menor taxa de verdadeiros positivos (TP), além de menores valores para precision e recall.

Figura 64 – Tabela de probabilidades da rede para reconhecimento da confiança do estudante

Node	Chance	% Probability	Reset	Close
classe_retomadas	nivel_confia...	ALTO MEDIO A... MEDIO B... BAIXO	ALTO	15.556 17.778 28.889 37.778
		BAIXO		25.806 51.613 19.355 3.226
classe_tp_hip_cf	nivel_confianca	NENHUMA POUCAS MUITAS	ALTO	45.455 47.727 6.818
		BAIXO		43.333 43.333 13.333
classe_prop_ph_cf	nivel_confianca	NENHUMA POUCAS MUITAS	ALTO	6.818 63.636 29.545
		BAIXO		3.333 46.667 50
classe_nivel_compreensao	nivel_confia...	ALTO MEDIO A... MEDIO B... BAIXO	ALTO	13.333 48.889 28.889 8.889
		BAIXO		9.677 12.903 51.613 25.806
classe_vis_dica2	nivel_confianca	NENHUMA POUCAS MUITAS	ALTO	75 20.455 4.545
		BAIXO		6.667 83.333 10
classe_vis_pseudo	nivel_confianca	NENHUMA POUCAS MUITAS	ALTO	54.545 43.182 2.273
		BAIXO		6.667 70 23.333
classe_avaliacao_aluno	nivel_confia...	ALTO MEDIO A... MEDIO B... BAIXO	ALTO	11.111 71.111 11.111 6.667
		BAIXO		3.226 32.258 41.935 22.581
classe_tp_prob_cf	nivel_confianca	NENHUMA POUCAS MUITAS	ALTO	38.636 54.545 6.818
		BAIXO		30 56.667 13.333
classe_vis_dica1	nivel_confianca	NENHUMA POUCAS MUITAS	ALTO	68.182 27.273 4.545
		BAIXO		6.667 80 13.333

Fonte: Elaborado pelo autor

5.5 Módulo para acompanhamento da motivação do aluno

Nesta seção será apresentado o módulo para reconhecimento da motivação, que foi implementado baseado nas redes bayesianas. Serão mostradas as interfaces e gráficos exibidos pelo sistema, além da arquitetura do protótipo que foi desenvolvido e usado nos testes e validação do modelo afetivo. A incorporação dos resultados da mineração de dados em um módulo do sistema PROALG atende o objetivo de apoiar o reconhecimento do esforço e confiança qualificando o processo de ensino e aprendizagem de programação.

O módulo para o reconhecimento dos fatores da motivação consiste na junção do método da problematização utilizado nas atividades, nas quais foram coletados os dados e o modelo afetivo, composto pelas redes bayesianas. A disponibilização de uma interface para acompanhamento da motivação está entre os objetivos desta tese, além de ser um dos requisitos funcionais do PROALG, apresentados no capítulo 3. Além disso a interface para consulta à confiança e ao esforço foi fundamental para a análise dos dados e validação do modelo em conjunto com o professor da disciplina.

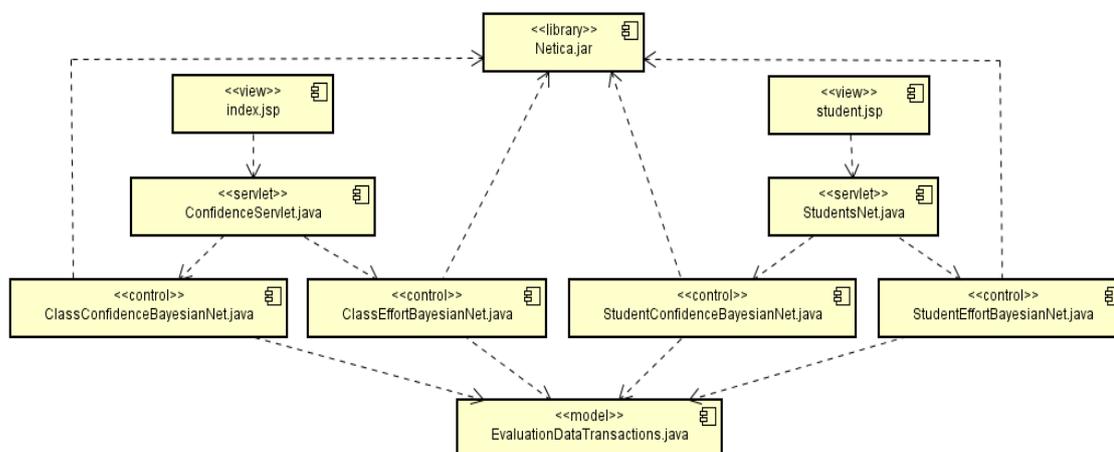
Inicialmente é necessário ressaltar que os resultados que serão apresentados a seguir descrevem o funcionamento de um protótipo do referido módulo, uma vez que o mesmo não foi completamente integrado ao sistema PROALG. As dificuldades para inserir a interface no sistema se deram em função do tempo que demandam os ajustes necessários para a integração

e devido à problemas relacionados à utilização da biblioteca da ferramenta Netica no servidor do software PROALG.

O software Netica possui APIs ou interfaces de programação para diversas linguagens de programação, entre elas C, Java, C#, entretanto, no momento em que o módulo foi desenvolvido não estavam disponíveis interfaces para linguagens para desenvolvimento web, tais como Java Script e PHP. Tal contexto elevou a dificuldade de inserir o módulo para acompanhamento da motivação diretamente no sistema PROALG, uma vez que o sistema utiliza essencialmente tecnologias voltadas ao desenvolvimento de aplicações para Internet.

Um aspecto importante do protótipo é que mesmo não tendo sido inserido diretamente no PROALG, há uma integração, uma vez que os dados utilizados na validação provém da base do sistema, o que possibilitou uma avaliação com dados reais, coletados durante as atividades. Mesmo com as dificuldades citadas, todos os recursos previstos, a exibição de gráficos para confiança e esforço, o uso das redes bayesianas e a transformação automática dos dados foram implementados na sua totalidade. A figura 65 exibe os principais componentes e classes que compõe o protótipo do módulo implementado.

Figura 65 - Diagrama de componentes que ilustra as principais classes do módulo.



Fonte: Elaborado pelo autor

A linguagem Java Script foi usada para criar as páginas que permitem a seleção de uma atividade ou turma e a exibição dos gráficos correspondentes. A interface index.jsp é responsável por mostrar os resultados por atividade, enquanto student.jsp exibe a confiança e esforço para cada estudante ao longo da disciplina.

A figura 66 mostra a escolha de uma atividade e a exibição dos gráficos por estudante, o que neste caso representa a motivação do aluno na tarefa selecionada. No exemplo foi

escolhida uma atividade cujo código é igual a 68 e o título “Público e renda”, que corresponde a primeira atividade publicada para a turma “2018B – T1”.

Foram utilizados gráficos do tipo Gauge (Figura 66), que se assemelham a velocímetros ou mostradores que indicam o quanto foi atingido de um valor máximo. O uso de cores que simbolizam alto (verde) e baixo (vermelho) permitiu refletir a situação do estudante e a proporção de cada categoria. Estes tipos de gráficos têm sido utilizados com frequência cada vez maior em sistemas de informação, especialmente em *dashboards* que contribuem para acompanhar indicadores relacionados a metas pré-definidas.

Figura 66 – Interface para visualização da motivação do estudante



Fonte: Elaborado pelo autor

Abaixo dos gráficos exibidos é exibido o número correspondente a probabilidade (em valores inteiros) do esforço ou confiança ser alto (cor verde), neste caso quanto menor este valor, maior a chance do estudante apresentar baixos níveis (cor vermelha). O menu à esquerda permite acessar a tela que mostra as mesmas informações por estudante (opção Aluno). Outro recurso interessante está disponível no botão denominado “Mais informações, que exhibe os valores de todos os atributos usados para gerar o gráfico, ou seja, as variáveis que compõe as redes bayesianas que retornam as probabilidades mostradas em cada gráfico.

As classes denominadas *ConfidenceServlet* e *StudentsNet* são acessadas pelas interfaces *jsp* e devem retornar as probabilidades para criação dos gráficos. Estas por sua vez recorrem aos componentes de controle (`<<control>>`) que efetivamente aplicam as redes bayesianas nos dados de cada atividade ou estudante para obter os percentuais relacionados à classe alto. O código mostrado na figura 67 ilustra o acesso às redes bayesianas realizado pela classe *ClassConfidenceBayesianNet*, que tem como objetivo obter as probabilidades relacionadas à confiança por atividade.

Figura 67 – Exemplo de código que retorna a probabilidade da confiança por atividade

```

Environ env = new Environ( null );

Net net = new Net( new Streamer( "C:\\Modelo_afetivo\\Resultados_confianca_atividade\\reco_confianca_ativ.dne" ) );
Node classe_tp_prob_cf      = net.getNode( "Classe_tp_prob_cf" );
Node classe_tp_hip_cf      = net.getNode( "Classe_tp_hip_cf" );
Node classe_prop_ph       = net.getNode( "Classe_prop_ph" );
Node visualizou_dica1     = net.getNode( "Visualizou_dica1" );
Node visualizou_dica2     = net.getNode( "Visualizou_dica2" );
Node visualizou_pseudo    = net.getNode( "Visualizou_pseudo" );
Node classe_nivel_compreensao = net.getNode( "Classe_nivel_compreensao" );
Node classe_retomadas     = net.getNode( "Classe_retomadas" );
Node avaliacao_aluno      = net.getNode( "Classe_avaliacao_aluno" );
Node nivel_confianca      = net.getNode( "Nivel_confianca" );

net.compile();

classe_tp_prob_cf.finding().enterState( student.getClasseTProbCF() );
classe_tp_hip_cf.finding().enterState( student.getClasseTPhipCF() );
classe_prop_ph.finding().enterState( student.getClassePropPH() );
visualizou_dica1.finding().enterState( student.getDica1() );
visualizou_dica2.finding().enterState( student.getDica2() );
visualizou_pseudo.finding().enterState( student.getPseudo() );
classe_retomadas.finding().enterState( student.getRetomadas() );
classe_nivel_compreensao.finding().enterState( student.getNivelCompreensao() );
avaliacao_aluno.finding().enterState( student.getAvaliacaoAluno() );

Object[] data = { nivel_confianca.getBeliefs(), student.getNome() + " " + student.getSobrenome() };

beliefs.add( data );

student.setNivelConfianca( nivel_confianca.getBelief( "ALTO" ) * 100 );

```

Fonte: Elaborado pelo autor

As classes denominadas *Environ*, *Net* e *Node* estão disponíveis na biblioteca *Netica.jar* que é fornecida pela desenvolvedora do software *Netica*, que foi utilizado na construção e treinamento das redes bayesianas que formam o modelo. Observa-se no código a abertura do arquivo “*rec_confianca_ativ.dne*” que armazena todas as informações da rede bayesiana após o treinamento. O mesmo é gerado na ferramenta *Netica*, a qual apresenta um recurso para gravar as configurações das redes em um documento (*.dne*), que pode ser empregado em instruções *java* ou em outra linguagem dentro de uma aplicação customizada.

As demais classes *ClassEffortBayesianNet*, *StudentConfidenceBayesianNet* e *StudentEffortBayesianNet* fazem uso de arquivos que guardam informações das redes que determinam respectivamente, o esforço por atividade, a confiança geral do estudante e o esforço geral do estudante. Percebe-se no exemplo que cada objeto da classe *Node* representa um atributo da rede e o valor da variável é atribuído pelo método *EnterState*. Estes valores são obtidos nas tabelas mostradas na figura 68, que armazenam os valores calculados para os atributos usados nas redes bayesianas. A tabela *dados_modafet_exercicio* guarda os dados por atividade e *dados_modafet_estudante* as informações resumidas do estudante.

Figura 68 - Tabelas que armazenam os valores dos atributos usados nas redes bayesianas

The image shows two screenshots of database tables. The left screenshot shows the table 'dados_modafet_exercicio' with the following attributes: id INT, exercicio_aluno_id INT(10), classe_tp_prob_ef VARCHAR(45), classe_tp_hip_ef VARCHAR(45), classe_tp_cod_ef VARCHAR(45), classe_tp_ph_ef VARCHAR(45), classe_tp_prob_cf VARCHAR(45), classe_tp_hip_cf VARCHAR(45), classe_prop_ph VARCHAR(45), classe_retom_adas VARCHAR(45), visualizou_dica1 CHAR(1), visualizou_dica2 CHAR(1), visualizou_pseudo CHAR(1), classe_nivel_detalle_prob VARCHAR(45), classe_nivel_detalle_hip VARCHAR(20), classe_nivel_com_preensao VARCHAR(45), classe_nivel_codigo VARCHAR(20), execucao_codigo CHAR(1), classe_avaliacao_aluno VARCHAR(45), probabilidade_baixo_ef DECIMAL(7,2), probabilidade_alto_ef DECIMAL(7,2), probabilidade_baixo_cf DECIMAL(7,2), and probabilidade_alto_cf DECIMAL(7,2). The right screenshot shows the table 'dados_modafet_estudante' with the following attributes: id INT, classe_tp_prob_ef VARCHAR(45), classe_tp_hip_ef VARCHAR(45), classe_tp_cod_ef VARCHAR(45), classe_tp_ph_ef VARCHAR(45), classe_vis_pseudo VARCHAR(45), classe_vis_dica1 VARCHAR(45), classe_vis_dica2 VARCHAR(45), classe_nivel_detalle_prob VARCHAR(45), classe_nivel_detalle_hip VARCHAR(45), classe_nivel_com_preensao VARCHAR(45), classe_ativ_nao_realizadas VARCHAR(45), grau_semelhanca VARCHAR(45), classe_prop_ph_cf VARCHAR(45), classe_tp_prob_cf VARCHAR(45), classe_tp_hip_cf VARCHAR(45), classe_retom_adas VARCHAR(45), classe_exec_cod VARCHAR(45), classe_avaliacao_aluno VARCHAR(45), probabilidade_baixo_ef DECIMAL(7,2), probabilidade_alto_ef DECIMAL(7,2), probabilidade_baixo_cf DECIMAL(7,2), probabilidade_alto_cf DECIMAL(7,2), turma_id INT(10), and usuario_id INT(10).

Fonte: Elaborado pelo autor

A classe *EvaluationDataTransaction* possui métodos para atualizar e consultar os dados armazenados nestas tabelas. A atualização dos dados obtém os dados das tabelas que registram dados da correção de cada exercício, acesso as dicas e pseudocódigo e sobre estes executa as ações de pré-processamento e transformações descritas nas seções anteriores. A rotina de atualização ou inclusão dos registros nas tabelas ocorre imediatamente após o usuário selecionar uma atividade ou turma e clicar no botão de “Ok”, antes da geração dos gráficos.

Após atualizar os dados de uma tarefa ou estudante, é executada uma consulta que obtém os registros que serão usados na elaboração dos gráficos do tipo Gauge. Tomando como exemplo a confiança e esforço por atividade, cada registro lido na tabela `dados_modafet_exercicio` gera dois gráficos, como mostrado na figura 66. Os valores dos campos são usados para compor um objeto da classe *Student*, cujas propriedades são utilizadas na sequência para alimentar a rede (Figura 67). O método *GetBelief* retorna as probabilidades associadas às categorias alto e baixo do nodo denominado `nivel_confianca` e posteriormente o percentual para a categoria alto é setado no objeto *student*.

A figura 69 exhibe o código SQL que faz a leitura dos dados da tabela e um exemplo dos resultados obtidos para um determinado exercício. Para cada registro retornado é instanciado

um objeto da classe *ConfidenceClassStudent*, que possui atributos para indicar tanto a confiança, quanto o esforço do estudante.

Figura 69 – Consulta aos dados da tabela dados_modafet_exercicio

```

ResultSet result = st.executeQuery( " select dm.id, " +
    " ea.exercicio_id, ea.titulo, u.nome, u.sobrenome, dm.classe_tp_prob_ef, dm.classe_tp_hip_ef, dm.classe_tp_cod_ef, " +
    " dm.classe_prop_ph, dm.visualizou_pseudo, dm.execucao_codigo, dm.classe_nivel_detalhe_prob, " +
    " dm.classe_nivel_compreensao, classe_nivel_detalhe_hip, dm.classe_tp_prob_cf, dm.classe_tp_hip_cf, " +
    " dm.classe_prop_ph, dm.visualizou_pseudo, dm.visualizou_dical, dm.visualizou_dica2, " +
    " dm.classe_nivel_compreensao, dm.classe_retomadas, dm.classe_avaliacao_aluno " +
    " from exercicio_aluno ea, dados_modafet_exercicio dm, usuario u " +
    " where ea.id=dm.exercicio_aluno_id and ea.usuario_id_aluno=u.id and ea.exercicio_id = " + classId +
    " order by u.nome ASC ");

while( result.next() )
{
    ConfidenceClassStudent student = new ConfidenceClassStudent();
    /*GENERAL DATA*/
    student.setDataId( result.getInt( "id" ) );
    student.setNome( result.getString( "nome" ) );
    student.setSobrenome( result.getString( "sobrenome" ) );
    student.setNivelCompreensao( result.getString( "classe_nivel_compreensao" ) );
    student.setClassePropPH( result.getString( "classe_prop_ph" ) );
    student.setDical( result.getString( "visualizou_dical" ) );
    student.setDica2( result.getString( "visualizou_dica2" ) );
    student.setPseudo( result.getString( "visualizou_pseudo" ) );

    /*CONFIANÇA*/
    student.setClasseTProbCF( result.getString( "classe_tp_prob_cf" ) );
    student.setClasseTPhipCF( result.getString( "classe_tp_hip_cf" ) );
    student.setRetomadas( result.getString( "classe_retomadas" ) );
    student.setAvaliacaoAluno( result.getString( "classe_avaliacao_aluno" ) );

    /*ESFORÇO*/
    student.setClasseTProbEF( result.getString( "classe_tp_prob_ef" ) );
    student.setClasseTPhipEF( result.getString( "classe_tp_hip_ef" ) );
    student.setClasseTPcodEF( result.getString( "classe_tp_cod_ef" ) );
    student.setNivelDetalheHip( result.getString( "classe_nivel_detalhe_hip" ) );
    student.setNivelDetalheProb( result.getString( "classe_nivel_detalhe_prob" ) );
    student.setNumeroExecucoes( result.getString( "execucao_codigo" ) );
}

```

exerc	titulo	exerc	classe_tp_prob_ef	classe_tp_hip_ef	classe_tp_cod_ef	classe_prop_ph	visualizou_pseudo	execucao_cod	classe_nivel_detalhe_prob	classe_nivel_compreensao	classe_nivel_detalhe_hip	classe_tp_prob	classe_tp_hip
68	Publico e renda	68	ALTO	BAIXO	MEDIO	PROPORCIONAL	N	N	BAIXO	MEDIO	MEDIO	NORMAL	NORMAL
68	Publico e renda	68	MEDIO	BAIXO	MEDIO	PROPORCIONAL	N	S	BAIXO	BAIXO	MEDIO	NORMAL	NORMAL
68	Publico e renda	68	ALTO	MEDIO	BAIXO	PROPORCIONAL	N	S	BAIXO	MEDIO	MEDIO	NORMAL	NORMAL
68	Publico e renda	68	BAIXO	ALTO	MEDIO	PROPORCIONAL	N	N	BAIXO	ALTO	ALTO	NORMAL	NORMAL
68	Publico e renda	68	MEDIO	BAIXO	ALTO	PROPORCIONAL	N	S	BAIXO	ALTO	ALTO	NORMAL	NORMAL
68	Publico e renda	68	BAIXO	MEDIO	MEDIO	PROPORCIONAL	N	S	MEDIO	ALTO	ALTO	NORMAL	NORMAL

Fonte: Elaborado pelo autor

Embora o exemplo descrito tenha focado na confiança por atividade, as rotinas que utilizam as demais redes bayesianas e geram gráficos para esforço por atividade ou para o esforço e confiança por estudante são semelhantes, apresentam a mesma lógica de processamento. As probabilidades são retornadas para as interfaces (index.jsp e student.jsp) que utilizam os valores para gerar os gráficos.

6 RESULTADOS E DISCUSSÃO

Este capítulo descreve os resultados obtidos na validação do modelo para identificação da motivação dos estudantes. As redes bayesianas foram aplicadas aos dados coletados em uma turma de estudantes na disciplina de Algoritmos e Programação que ocorreu no segundo semestre de 2018, da qual participaram 26 estudantes. A validação incluiu um acompanhamento em conjunto com o professor, visando apoiar ações baseadas nos índices e níveis de motivação mostrados no sistema.

Na seção 6.2 serão descritos e analisados os resultados das respostas dos estudantes participantes do estudo piloto e estudo caso para os questionários que foram aplicados com o objetivo de obter um autorrelato por parte dos discentes. Serão apresentados gráficos e estatísticas segmentadas de acordo com o nível de esforço e confiança atribuídos aos estudantes nos dados usados no treinamento das redes bayesianas.

A análise destes resultados buscou verificar a relação entre o autorrelato e a classificação adotada na construção do modelo. A análise dos resultados buscou responder questões como:

- Estudantes definidos com baixo nível de esforço reconheceram que se esforçaram menos ou que poderiam ter se dedicado mais?
- Estudantes com baixa confiança reconhecem as dificuldades e problemas no desenvolvimento das atividades?
- Quais estados afetivos foram mais intensos entre estudantes com baixos níveis de esforço e confiança?

6.1 Validação do modelo para reconhecimento da motivação

Os testes e a validação do modelo afetivo e do módulo para reconhecimento da motivação, apresentados no capítulo 5 foram realizados no segundo semestre de 2018. As atividades foram desenvolvidas em uma turma da disciplina de Algoritmos e Programação, composta por 25 estudantes, oriundos dos cursos de Engenharia de Software, Engenharia de Computação e Sistemas de Informação. Inicialmente estavam matriculados 27 alunos, porém 2 deles desistiram ainda nas duas primeiras semanas e desta forma foram excluídos da pesquisa.

Ao longo do semestre foram publicadas 18 exercícios de programação, sendo 6 destes nas primeiras 6 semanas da disciplina, aproximadamente 1 exercício por semana. Estas atividades foram corrigidas e os dados destas foram utilizadas na realização dos testes, para validação do módulo de acompanhamento da motivação e das redes bayesianas que compõe o modelo afetivo.

Com a aplicação do protótipo para o processamento dos dados e determinação das probabilidades relacionadas à confiança e esforço foram obtidos os percentuais mostrados na tabela 25, na qual os nomes dos alunos foram substituídos por um código sequencial. Os valores correspondem aos percentuais da categoria alto, tanto no caso da confiança, quanto do esforço, sendo assim, o valor de 78,59 como o mostrado na primeira coluna para o aluno 1 indica que a probabilidade de confiança ser baixa é de 21,41, o que indica que a rede bayesiana classificou o nível de confiança como predominantemente alto, com aproximadamente 78% de certeza.

Percebe-se que em grande parte dos casos, os valores se aproximam de 100 ou ficam próximos a 0 (zero), o que mostra que o modelo apresenta alto grau de certeza na determinação de um nível alto ou baixo. Entretanto, há outras situações onde existe uma tendência maior para uma das categorias, mas não é possível determinar claramente a classe dominante, como no caso da confiança associada à atividade 6 realizada pelo estudante 2, cujo percentual foi 49,98. Neste exemplo há aproximadamente 50% de chance de ser alto e praticamente o mesmo percentual de ser baixo, o que caracteriza um baixo grau de certeza na classificação do nível de confiança para a atividade.

Este é um dos aspectos importantes de uma abordagem probabilística, a determinação de uma classe com uma probabilidade associada, tal como observa-se no trabalho de Longhi (2011). No exemplo citado, se o sistema apenas indicasse que a confiança do estudante foi alta, o professor poderia inferir que não haveria necessidade de um acompanhamento mais direto e que o estudante apresentou confiança plenamente satisfatória, porém um baixo grau de certeza indica que o cenário não foi tão favorável e que talvez seja necessário avaliar melhor o comportamento do discente ao longo das atividades seguintes.

Os dados mostrados na tabela 25 demonstram claramente a evolução e o perfil de cada aluno ao longo de um conjunto de atividades, o que confirma a afirmação de Silva et al. (2016) que argumenta que a mineração de dados por atividade contribui para um acompanhamento constante e para o desenvolvimento de ações entre uma atividade e outra.

As células que não apresentam valores correspondem a exercícios não resolvidos pelos estudantes, neste caso, não houve correção e os dados não foram usados para aplicação do modelo, portanto, não é possível determinar o nível de confiança e esforço, embora estes registros sejam importante, posteriormente para a indicação do esforço geral do estudante.

Tabela 25 – Probabilidades de confiança e esforço por atividade e estudante

Aluno	Atividade 1		Atividade 2		Atividade 3		Atividade 4		Atividade 5		Atividade 6	
	Conf.	Esf.										
Aluno1	78,59	13,18	56,48	24,54	16,90	2,70	88,29	78,80	88,29	98,03	2,96	95,40
Aluno2	88,41	99,97	92,71	29,97	82,66	58,92	61,27	31,69	0,04	92,13	49,98	100,00
Aluno3	99,37	99,77	92,03	99,99	99,07	100,00	96,84	100,00	99,76	100,00	97,55	99,20
Aluno4	0,00	97,09			0,05	5,54			10,88	1,19		
Aluno5							11,99	61,27	96,18	8,85	1,72	81,69
Aluno6	99,37	99,56	99,37	93,72	99,01	47,80	97,55	94,78				
Aluno7	97,55	99,58	99,63	100,00	99,37	93,72	71,13	100,00	97,55	55,97	87,26	83,63
Aluno8	98,13	42,33	90,04	0,06	97,08	4,02	69,43	0,10	58,97	0,00	76,39	4,01
Aluno9	76,39	25,07	87,26	86,78	92,95	73,60	1,46	93,72	56,59	97,48		
Aluno10	99,07	99,94	76,39	99,69	95,26	100,00	98,53	100,00	98,53	100,00	99,07	100,00
Aluno11	97,14	99,80	91,99	99,84	0,35	42,71	8,46	75,50	0,01	42,71	4,32	0,06
Aluno12	99,37	84,49	66,94	98,23	76,39	97,20	9,46	0,76			87,26	96,35
Aluno13			97,55	99,73	0,01	0,03			18,33	60,29		
Aluno14	0,49	15,18	26,90	18,23	0,05	0,04	1,41	0,04	19,25	0,03	0,22	22,27
Aluno15	87,90	99,31	87,90	93,39	91,99	99,44	0,45	10,00	99,37	95,58		
Aluno16	0,157	14,60	74,10	47,96	27,36	2,59					0,19	5,85
Aluno17	97,55	95,58							45,09	86,68		
Aluno18	99,76	100,00	99,07	100,00	92,71	100,00	97,55	99,91	99,07	100,00	97,55	99,73
Aluno19	99,76	99,98	84,37	100,00	76,39	91,05	94,52	77,18	99,01	99,77	35,78	96,57
Aluno20	98,53	99,93	97,55	87,01	58,97	0,06	16,50	96,63	13,73	94,40	35,53	85,23
Aluno21	16,50	81,59	98,13	8,17	99,01	99,29	76,19	96,22				
Aluno22	92,95	8,17	76,39	96,63	81,25	83,07	99,01	85,57	99,01	62,07	85,13	2,23
Aluno23	99,01	23,62	92,95	0,05	15,60	0,13	69,43	0,00				
Aluno24	99,01	94,09	99,07	100,00			99,01	99,77	99,37	94,40	85,54	99,97
Aluno25	96,18	97,73	89,61	99,99	16,90	2,29	51,76	2,29	89,30	2,29	45,09	85,22

Fonte: Elaborado pelo autor

Alguns estudantes chamaram a atenção por apresentam sistematicamente baixos níveis de motivação, como nos casos dos alunos 14 e 16, cujos percentuais tanto para esforço quanto para confiança ficaram bem abaixo do desejado. Já os estudantes identificados com os números 23 e 8 demonstraram na maioria dos casos boa confiança associada a baixos níveis de esforço, o que pode indicar que eles confiam na sua capacidade de resolução, embora exista a necessidade de acompanhamento do desempenho por parte do professor com o intuito de verificar se o esforço foi suficiente.

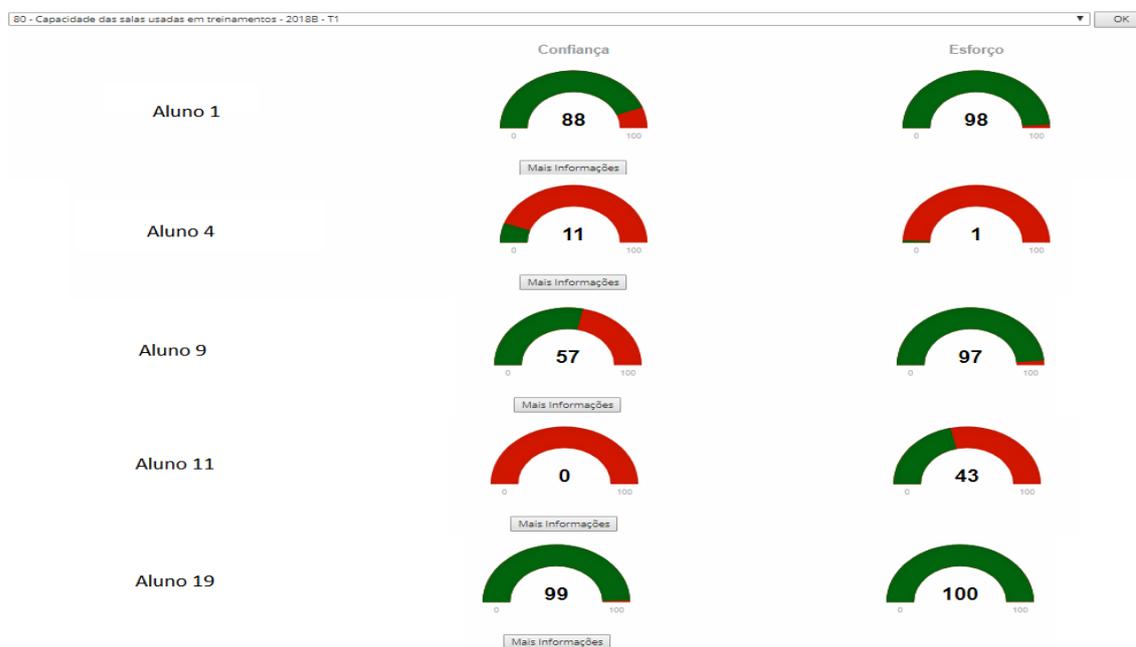
Com relação às atividades observa-se uma maior predominância de valores mais reduzidos nas duas últimas, que apresentaram um grau de dificuldade maior para a sua resolução. A atividade 1 exigiu apenas o uso de variáveis, entradas, saídas e expressões de cálculo, enquanto as duas seguintes (2 e 3) envolviam também instruções condicionais. Já para

resolver os exercícios seguintes (3 a 6) era necessário usar instruções de repetição e eventualmente vetores para o armazenamento de dados em memória.

É possível perceber que o aumento da complexidade pode influir na motivação, pois um aluno pode demonstrar um esforço abaixo do necessário ou uma menor confiança na sua capacidade de resolver o problema. Problemas mais complexos exigem persistência e dedicação (Dundunmay, Banawan e Rodrigo, 2017), e neste caso demandam por parte do estudante mais tempo ou um maior detalhamento do problema, de forma análoga ao que ao número de tentativas definido como indicativo de esforço por Del Soldato e Du Boulay (1995, 2016).

Para demonstrar os resultados do modelo no protótipo implementado foram gerados os gráficos das atividades. Em função do grande número de participantes e da dificuldade de exibir todos os gráficos, a figura 70 ilustra parte dos resultados, gráficos correspondentes à atividade 5, para alguns dos estudantes listados na tabela 25. Os gráficos contribuem para ilustrar a intensidade da confiança e do esforço. A variação na intensidade em diferentes tarefas pode demonstrar a relevância da situação-problema para o indivíduo, bem como a crença do estudante em sua capacidade de enfrentamento para responder ao problema (Scherer, 2005).

Figura 70 – Resultados do módulo de acompanhamento para a atividade 5



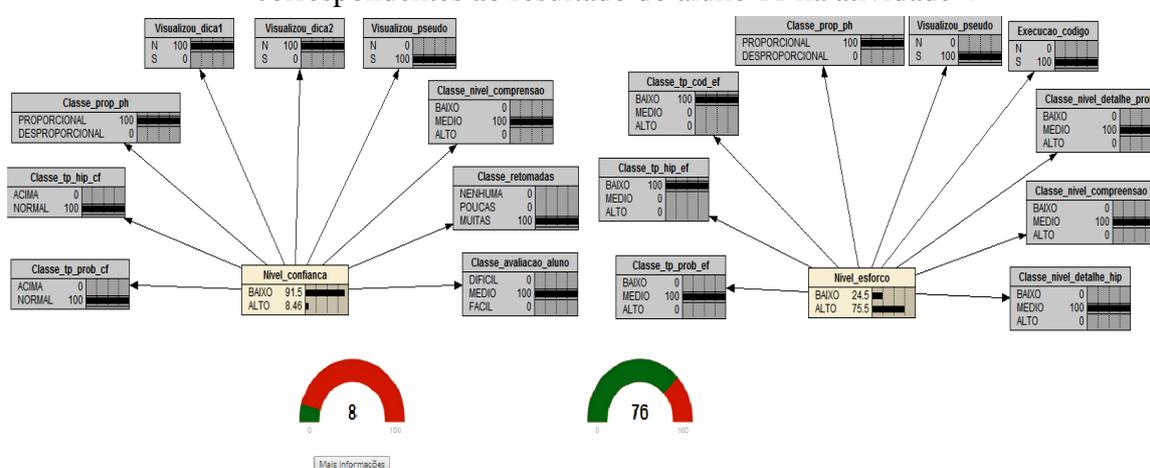
Fonte: Elaborado pelo autor

Visualmente é possível ter uma ideia sobre o perfil do comportamento do aluno na atividade, na qual os estudantes 4 e 11 apresentam baixos níveis tanto para confiança, e no caso do quarto estudante também para o esforço. Os que foram identificados como 3 e 19 apresentam alta motivação com uma combinação de percentuais próximos do máximo em ambos os fatores.

Uma ressalva importante é que no protótipo são apresentados os nomes reais dos participantes, que foram substituídos na imagem para evitar a divulgação de informações pessoais.

Uma ilustração dos valores dos atributos e da aplicação da rede bayesiana para gerar os gráficos é ilustrada na figura 71, a qual apresenta os dados de uma atividade para um determinado aluno. Ao atribuir os valores aos atributos, a RB produz as probabilidades associadas as classes alto e baixo e estas se constituem nas crenças (*beliefs*) obtidas pelo módulo que as utiliza para exibir o gráfico mostrado no final da imagem.

Figura 71 - Configuração das redes bayesianas para confiança e esforço por atividade correspondentes ao resultado do aluno 11 na atividade 4



Fonte: Elaborado pelo autor

No exemplo da figura 71 é possível perceber a influência de alguns atributos nos níveis de confiança e esforço. O fato de ter muitas retomadas, aliado ao acesso ao pseudocódigo é decisivo para que o nível de confiança neste exercício seja baixo, mesmo que nas demais variáveis, como nos tempos, os valores sejam os desejáveis. Os dados observados corroboram com a teoria de Du Boulay e Del Soldato (2016), a qual indica que o acesso a recursos de ajuda reflete uma menor confiança. No caso do esforço, alguns atributos apresentam valores negativos, como os tempos do código e da hipótese, porém a ocorrência de valores médios nos níveis de detalhe e compreensão contribui para aumentar o nível de esforço, embora este tenha sido relativo (75%).

O exemplo das redes exibidas na imagem ilustra como os percentuais são obtidos no módulo desenvolvido, da mesma forma que a ferramenta Netica permite definir os valores de forma interativa, é possível executar esta ação no código fonte, como mostrado na figura 66. O método *EnterState* da classe *node* redundante em uma definição de 100% de chance para um valor de um atributo, e, conseqüentemente a alteração dos percentuais das demais variáveis, incluindo o nível de confiança ou esforço.

O módulo de acompanhamento apresenta uma interface para consultar os dados para cada estudante, que utiliza as redes bayesianas para determinar os níveis de confiança e de esforço por aluno. O professor pode selecionar uma turma para a qual são processados os dados, gerando os valores de cada atributo utilizado nas RB. A tabela 26 mostra os percentuais de esforço e confiança para os estudantes da turma de 2018B, na qual foi realizada a validação do modelo.

Tabela 26 - Probabilidades de confiança e esforço por estudante

Aluno	Confiança	Classe_cf	Esforço	Classe_ef
Aluno1	81,00	ALTO	52,87	ALTO
Aluno2	12,96	BAIXO	91,38	ALTO
Aluno3	100,00	ALTO	99,79	ALTO
Aluno4	0,40	BAIXO	3,43	BAIXO
Aluno5	86,14	ALTO	0,02	BAIXO
Aluno6	100,00	ALTO	73,64	ALTO
Aluno7	100,00	ALTO	99,39	ALTO
Aluno8	99,89	ALTO	3,31	BAIXO
Aluno9	22,29	BAIXO	62,00	ALTO
Aluno10	99,96	ALTO	99,32	ALTO
Aluno11	0,43	BAIXO	1,91	BAIXO
Aluno12	95,95	ALTO	72,12	ALTO
Aluno13	2,00	BAIXO	0,06	BAIXO
Aluno14	1,13	BAIXO	0,79	BAIXO
Aluno15	25,12	BAIXO	52,76	ALTO
Aluno16	27,84	BAIXO	15,09	BAIXO
Aluno17	99,84	ALTO	99,79	ALTO
Aluno18	99,98	ALTO	99,94	ALTO
Aluno19	99,99	ALTO	99,09	ALTO
Aluno20	60,94	ALTO	92,87	ALTO
Aluno21	98,58	ALTO	99,57	ALTO
Aluno22	99,97	ALTO	13,78	BAIXO
Aluno23	99,89	ALTO	0,81	BAIXO
Aluno24	100,00	ALTO	89,35	ALTO
Aluno25	99,86	ALTO	3,18	BAIXO

Fonte: Elaborado pelo autor

Analisando os percentuais mostrados na tabela e os gráficos apresentados na figura 72 é possível verificar que em muitos casos há um alto grau de certeza quanto à classe dominante, pois vários estudantes apresentam valores próximos ao máximo (100) ou mínimo (0). Os gráficos apresentam os mesmos percentuais mostrados na tabela 26, com valores inteiros, contribuindo para uma análise visual, na qual é possível perceber claramente os casos mais positivos ou negativos.

Os resultados por estudante apresentam uma visão complementar aos resultados por atividade, seguindo uma abordagem holística, na qual é possível visualizar o todo, de forma

análoga ao que ocorre nos trabalhos de Longhi (2011) e Santos (2016). Um acompanhamento apenas por atividade pode ser insuficiente, uma vez que algumas tarefas não são concluídas pelos estudantes. Para determinar a motivação do estudante são levados em conta também o número de atividades não realizadas e o grau de semelhança entre as respostas de todas as tarefas.

Figura 72 - Gráficos exibidos pelo módulo de acompanhamento para cada estudante



Fonte: Elaborado pelo autor

Na imagem mostrada na figura 72 os gráficos foram colocados lado a lado, em função do grande número de alunos, para que fosse possível mostrados a lista completa, porém, no software os mesmos são exibidos sequencialmente, um abaixo do outro.

Os resultados descritos nesta seção foram analisados em conjunto com o professor que estava ministrando o componente curricular, e que também foi o professor responsável pelos componentes que serviram de base para o estudo piloto e estudo de caso. Em uma reunião entre pesquisador e professor os dados e gráficos foram apresentados ao docente com objetivo de registrar as observações e impressões do mesmo. Por ter participado de forma ativa da pesquisa nos semestres anteriores, não foi necessário explicar ao professor o método da problematização e os princípios que nortearam o desenvolvimento do modelo para reconhecimento da motivação.

Inicialmente foram apresentados os gráficos por atividade e em seguida por estudante, com o objetivo de verificar a reação e a impressão do docente sobre os resultados. Foi possível perceber um alto grau de satisfação e um certo entusiasmo relacionado a interface, e à facilidade de perceber claramente os casos de maior e menor confiança e esforço. A abordagem baseada em gráficos do tipo Gauge foi considerada excelente, simples e intuitiva, sem a necessidade de muitas informações adicionais, na visão do professor. Para analisar os resultados das atividades, foi apresentada também a tabela 26 que resume os percentuais para cada aluno, por atividade. Da mesma forma que já havia sido observado pelo pesquisador, foi constatado pelo docente uma dificuldade maior nas últimas atividades, que apresentaram maior complexidade. Nas palavras do professor: *“Quanto mais complexo o problema, maior é a dificuldade de especificar, de entender o exercício. Muitos deles conseguem produzir o código, a solução, mas a descrição do problema é simples ou as vezes confusa...Acho que nós também precisamos revisar e melhorar alguns enunciados e textos sobre o problema.”*

Na avaliação dos estudantes e da situação de cada um deles foi apresentada a tabela 27, para complementar os gráficos exibidos pelo sistema (Figura 72). A primeira constatação foi de que um número reduzido de atividades finalizadas é um problema para a confiabilidade dos níveis de esforço e confiança. Tomando como base o caso do aluno 17, que realizou somente duas atividades e apresentou altos níveis nos fatores da motivação, percebe-se que o resultado final pode ser enganoso, não sendo possível afirmar que ele realmente está confiante ou que se esforçou adequadamente. Um alto número de atividades não realizadas foi insuficiente para determinar um nível menor de esforço.

Ao avaliar as informações sobre os estudantes, o professor considerou que a maioria dos casos era coerente com as suas observações e até mesmo com o desempenho apresentados pelos estudantes até aquele momento. Os discentes identificados com os códigos 4, 11, 13, 14, 16 apresentavam níveis baixos tanto para confiança, quanto para o esforço, fato que na opinião do professor estava de acordo com as observações feitas durante a disciplina. Observando as avaliações e notas atribuídas a estes alunos até o momento da aplicação do modelo, foi possível perceber que estas estavam próximas ou abaixo da média mínima para aprovação. Inclusive o aluno 16 veio a desistir da disciplina em um momento posterior, o que de certa forma contribuiu para confirmar baixos índices de esforço e especialmente de confiança.

Dentre os estudantes que apresentaram uma combinação de altos níveis de confiança e esforço, a maioria segundo o docente estava obtendo um bom aproveitamento e de acordo com a percepção deste realmente se esforçavam para realizar as atividades. Tomando como exemplo os estudantes com os códigos 24, 18, 7, 10, 3 e 2, estes apresentaram índices positivos para os dois fatores e não despertavam grandes preocupações por parte do docente, sendo considerados dedicados e ativos durante as atividades.

Alguns casos, porém despertaram dúvidas na análise, especialmente os estudantes 12 e 17 que apresentavam um comportamento pouco ativo e dedicado na visão do professor e que foram classificados com alta motivação pelo modelo. Uma dúvida levantada pelo docente diz respeito a uma distinção clara entre a confiança e o esforço, que segundo ele era algo complexo de ser percebido pelas observações durante as aulas. Vários estudantes apresentaram alta confiança e baixo, esforço, porém na análise do professor, alguns apresentavam desempenho satisfatório até o momento, enquanto outros, geravam uma certa preocupação.

Mesmo que algumas dúvidas e questões tenham sido suscitadas pelo professor, a avaliação final dele sobre o sistema e o apoio da ferramenta foram extremamente positivos. Os resultados foram considerados relevantes e nas palavras do docente “seriam valiosos” ao longo dos exercícios para que este pudesse conversar e dar uma atenção maior aos alunos com maiores dificuldades e índices de motivação reduzidos. Durante as análises foi levantada a possibilidade de gravar ou registrar uma posição, a situação dos alunos em determinado momento para que posteriormente fosse possível estabelecer uma comparação, após a finalização de novas atividades. Embora não tenha havido tempo hábil para implementar este recurso, a sugestão foi considerada importante, sendo considerada uma demanda para trabalhos futuros.

A análise em conjunto com o professor permitiu constatar que, na maioria dos casos, a observação do docente está mais associada com o desempenho dos estudantes, uma vez que a

avaliação da questão afetiva é mais complexa. Na maioria das situações o desempenho e aspectos cognitivos percebidos durante a disciplina embasam as ações do docente, entretanto, o módulo de acompanhamento da motivação contribui com informações sobre a motivação do estudante, o que segundo Picard et al. (2004) é importante para reduzir o desequilíbrio entre o cognitivo e o afetivo.

6.2 Resultados da autoavaliação do estudante

A partir desta seção serão apresentados e analisados os resultados dos questionários para autorrelato dos estudantes, aplicados no estudo piloto e nas duas fases do estudo de caso. A lista completa de questões e alternativas pode ser encontrada no apêndice B. Em diversas questões as respostas foram segmentadas de acordo com a classificação do estudante quanto ao esforço e confiança, categorização realizada nos dados utilizados no aprendizado das redes bayesianas, descrito no capítulo 5.

Uma análise conjunta dos resultados do modelo e do relato dos estudantes é compatível com as ideias de diferentes autores (Picard, 1997; Bercht, 2001; Longhi, 2011; Santos, 2016), que argumentam que a identificação de estados afetivos devem considerar dados de diferentes fontes. O autorrelato dos estudantes, porém pode não ser suficiente para identificar um estado afetivo e pode mascarar determinadas situações, caso o aluno não responda de maneira adequada.

As questões relacionadas à confiança e ao esforço foram aplicadas tanto no estudo piloto, quanto no estudo de caso e foram respondidas por 57 estudantes, de um total de 68, um percentual de 83% de respondentes. A questão relacionada ao sistema PROALG foi aplicada somente no estudo de caso, obtendo 46 respostas de 54 possíveis (85%).

O principal objetivo foi verificar se a autoavaliação do estudante corrobora com a classificação utilizada no modelo, se existe padrões diferentes de respostas para estudantes com baixa confiança e esforço. A análise das respostas contribuiu para indicar que em algumas questões as respostas confirmam a categorização adotada no modelo e em outras situações não é possível afirmar que existe uma relação direta entre os níveis de confiança e esforço determinados pelo modelo e os resultados da autoavaliação.

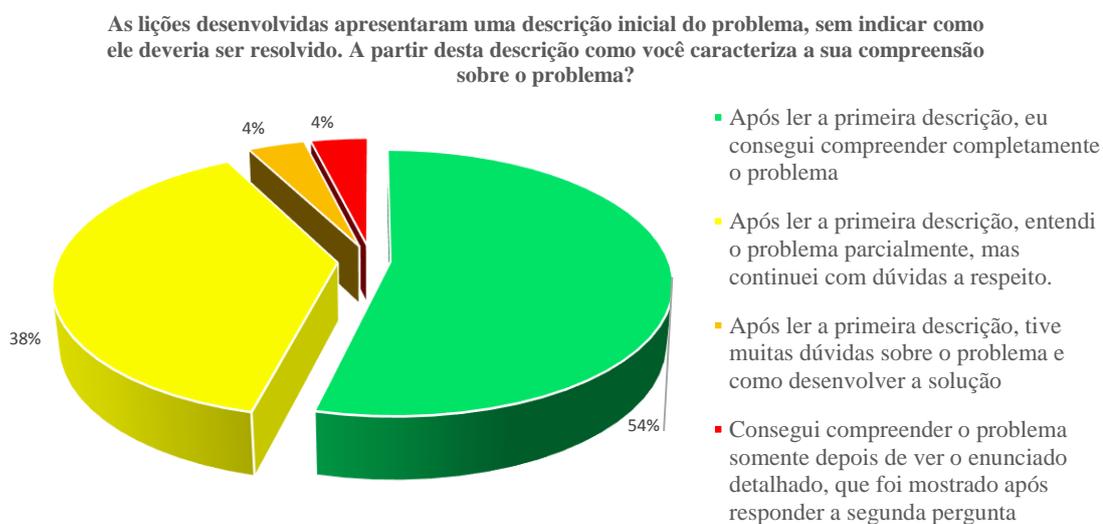
6.2.1 Respostas para questões relacionadas à confiança

As questões inseridas no questionário tiveram como objetivo avaliar a percepção do próprio aluno sobre a confiança ou esforço demonstrado por ele durante as atividades, sendo

assim, cada questão está relacionada a um dos fatores. Apenas a questão 15 que perguntava qual o grau de satisfação do estudante foi analisada sob o ponto de vista dos dois fatores.

Inicialmente serão apresentadas as questões baseadas no fator de confiança, como é o caso da questão 1, cujo objetivo foi verificar a percepção do aluno sobre a compreensão que ele teve do problema. Esta questão permite verificar também se os estudantes acreditam que a definição para o problema foi, de certa forma, suficiente para a resolução da atividade.

Figura 73 - Respostas dos estudantes classificados com alta confiança para a questão 1 (compreensão do problema)

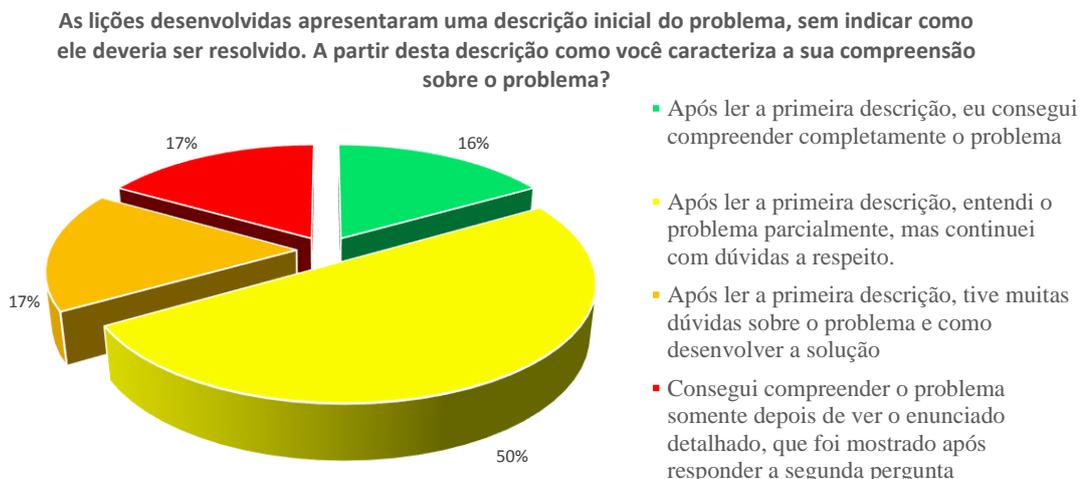


Fonte: Elaborado pelo autor

O gráfico mostrado na figura 73 reflete as respostas dos estudantes cuja confiança foi categorizada como alta, enquanto a figura 74 mostra as respostas para os que exibiram baixa confiança. Observa-se no um crescimento dos estudantes que indicam que tiveram dúvidas ou compreenderam o problema somente depois de ver o enunciado, no grupo que têm baixa confiança.

Entre os alunos com alta confiança, 92% relatam que compreenderam total ou parcialmente o problema, sendo que no segundo grupo este percentual é de 66%. Os resultados para esta questão podem indicar que há uma discrepância entre os dois grupos e que a classificação usada no treinamento é compatível com o autorrelato do estudante. Os percentuais indicam claramente que estudantes que foram categorizados com baixa confiança no modelo afetivo demonstram uma crença menor na compreensão do problema.

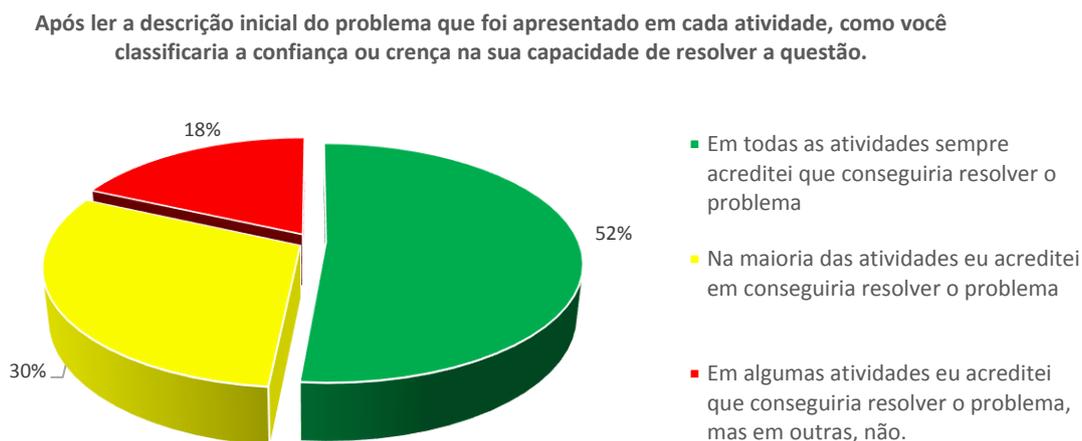
Figura 74 - Respostas dos estudantes classificados com baixa confiança para a questão 1 (compreensão do problema)



Fonte: Elaborado pelo autor

A questão 2 foi ainda mais direta na tentativa de verificar a confiança demonstrada pelo estudante, uma vez que o termo confiança foi explicitamente mencionado no enunciado da questão. Não foi apresentada uma alternativa negativa, ou seja, uma alternativa na qual o estudante indicasse que não acreditava que era capaz de resolver o problema, mas foram apresentadas opções que refletem níveis, desde uma confiança total, passando por crenças parciais na capacidade de resolução.

Figura 75 - Resposta dos que tem alta confiança para a questão 2 (capacidade de resolução do problema)



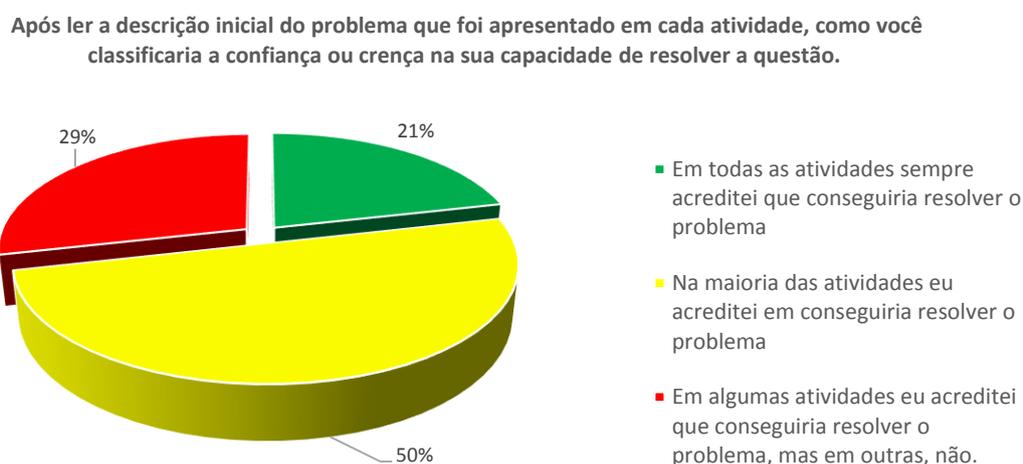
Fonte: Elaborado pelo autor

As diferenças entre os dois grupos ficam ainda mais evidentes nesta questão, uma vez que o grupo categorizado com alta confiança apresenta um percentual superior a 50% na alternativa que indica maior nível de crença (“Em todas as atividades eu acreditei que

conseguiria resolver o problema”), enquanto o segundo grupo exibe apenas 21% de respostas nesta opção.

No grupo classificado com baixa confiança há um crescimento dos que indicam que entendiam que conseguiriam resolver o problema em algumas situações e em outras não. Os resultados exibidos nas figuras 75 e 76 reafirmam a relação entre a categorização adotada no modelo afetivo para a confiança e o autorrelato dos estudantes. Scherer (2005), Du Boylay e Del Soldato relacionam a crença na capacidade de enfrentar uma situação ao estado afetivo, desta forma a associação entre uma baixa confiança e a crença relatada pelo estudante é importante para demonstrar a validade do modelo para reconhecimento da motivação.

Figura 76 -Resposta dos que tem baixa confiança para a questão 2 (capacidade de resolução do problema)



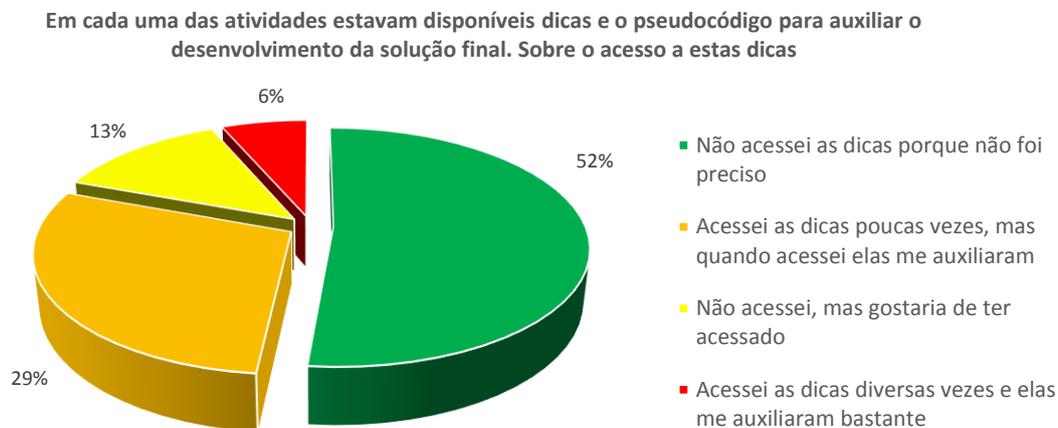
Fonte: Elaborado pelo autor

A ordem das questões propostas não seguiu uma lógica de apresentar inicialmente todas as que refletem confiança e em seguida as que indicam esforço, as mesmas foram intercaladas também com questões sobre o método e o sistema utilizado. Tal estratégia foi empregada para diminuir a possibilidade dos estudantes perceberem um padrão e a partir deste responderem de maneira muito semelhante as questões.

A questão 10 foi elaborada para avaliar a percepção do aluno quanto às dicas, os motivos que o fizeram acessar ou não as dicas que foram disponibilizadas em cada exercício. É importante ressaltar que os atributos que indicam se o aluno acessou ou não as dicas e o conteúdo do pseudocódigo foram relevantes no modelo para o reconhecimento da motivação. Del Soldato e Du Boulay (1995) e Bercht (2001) destacam que a quantidade de requisições ou acesso a recursos que auxiliam na resolução das tarefas é essencial para determinar a confiança,

por isso foram propostas questões para verificar se os estudantes reconhecem que precisaram de auxílio durante a resolução dos problemas.

Figura 77 - Resposta dos alunos com alta confiança para a questão 10 (acesso as dicas)

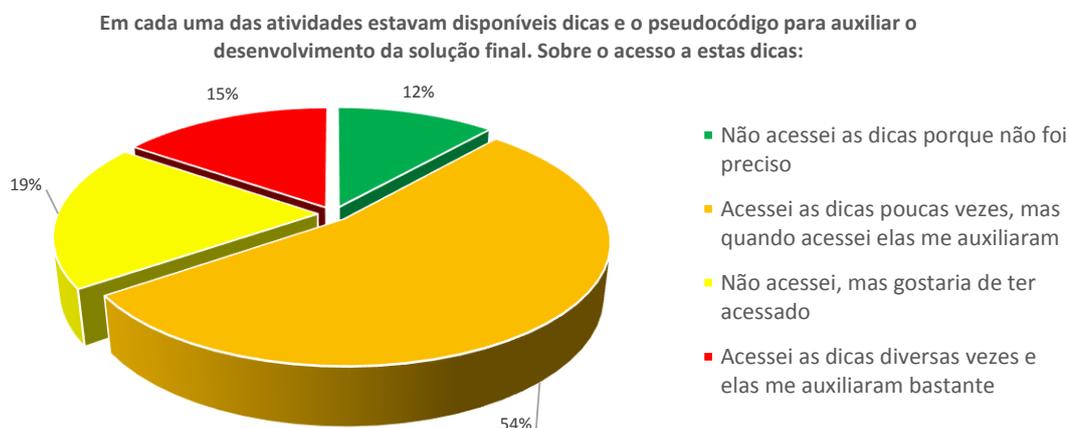


Fonte: Elaborado pelo autor

Constata-se no gráfico mostrados na figura 77 que estudantes que foram determinados com baixa confiança reconhecem que acessaram as dicas, poucas ou diversas vezes que estas auxiliaram na resolução dos problemas. No grupo de maior confiança o percentual de respostas indicando que não houve acesso é bem superior (52%), se comparado ao segundo grupo.

A relação entre um menor nível de confiança e o acesso mais frequente às dicas fica evidente neste caso, o que contribui para demonstrar que variáveis relacionadas ao uso dos recursos de ajuda são importantes para o reconhecimento da confiança. A questão 11, mostrada no apêndice B foi aplicada como forma de complementar a questão 10, sendo que os resultados obtidos com as respostas para ele mostram as mesmas tendências e por este motivo não serão apresentados com detalhes.

Figura 78 - Resposta dos alunos com baixa confiança para a questão 10 (acesso as dicas)



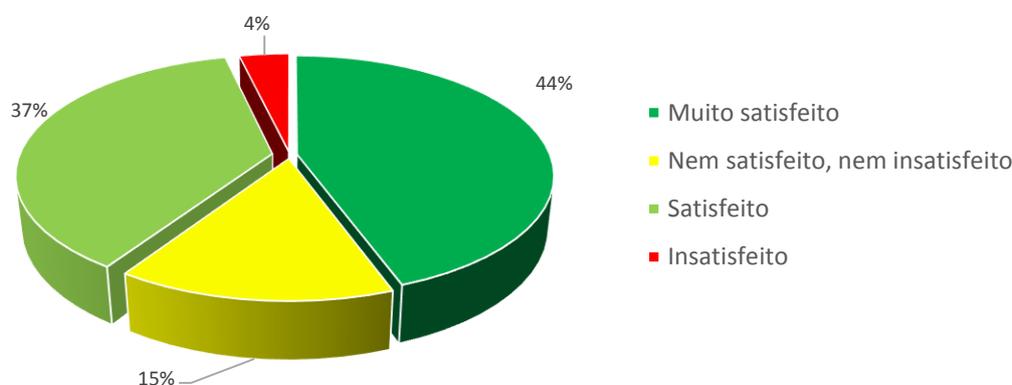
Fonte: Elaborado pelo autor

A questão 14 permitiu uma análise da satisfação do estudante com as atividades desenvolvidas. A satisfação pode estar relacionada tanto à confiança, quanto ao esforço, e indica ainda se o demonstrou satisfação quanto ao método aplicado. O objetivo com a apresentação dos resultados segmentados de acordo com a confiança e também esforço é verificar se níveis menores de confiança ou esforço impactam mais diretamente a satisfação do estudante e por consequência a motivação deste.

Autores relacionam a satisfação do estudante com sentimentos como o prazer, felicidade, interesse, esperança ou a valorização (SACHARIN, SCHLEGEL e SCHERER et al., 2012). Uma maior valorização ou auto estima por sua vez associada a uma maior confiança (BAUMEISTER et al.), sendo assim a análise do grau de satisfação contribui para avaliar a confiança demonstrada pelo estudante. Além da confiança, alunos mais satisfeitos na maioria dos casos tendem a se dedicar e perseverar mais nas tarefas, o que caracteriza um maior nível de esforço.

Figura 79 - Respostas de alunos com alta confiança para a questão 14 (satisfação)

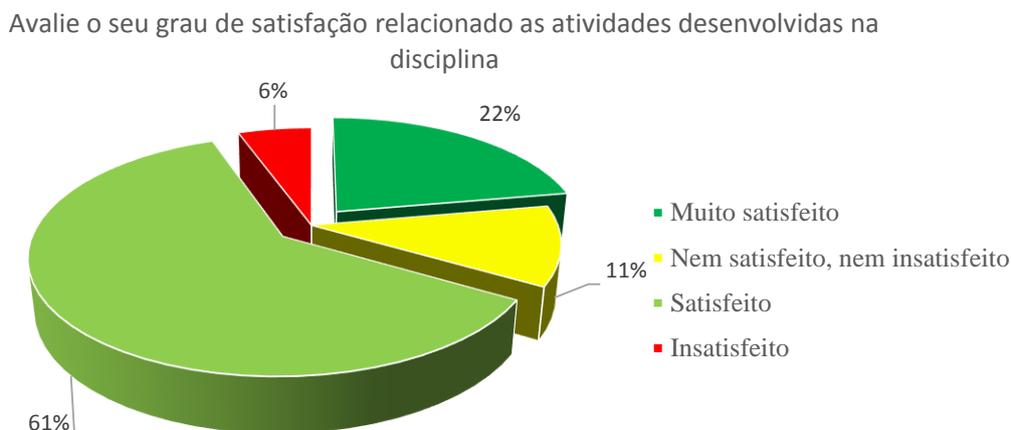
Avalie o seu grau de satisfação relacionado as atividades desenvolvidas na disciplina



Fonte: Elaborado pelo autor

Analisando os gráficos das figuras 79 e 80 é possível constatar que um percentual maior de estudantes com alta confiança relatam estar muito satisfeitos (44%), enquanto este mesmo percentual é de 22% no conjunto de alunos com baixa confiança. No primeiro grupo, um número um pouco maior se mostra indiferente (nem satisfeito, nem insatisfeito), embora a soma dos satisfeitos e muito satisfeitos seja semelhante nos dois grupos (81% e 83%). Considerando os percentuais exibidos, não é possível perceber claramente uma distinção entre os dois grupos, embora o item que indica maior satisfação seja mais forte entre estudantes com alto grau de confiança.

Figura 80 - Respostas de alunos com baixa confiança para a questão 14 (satisfação)

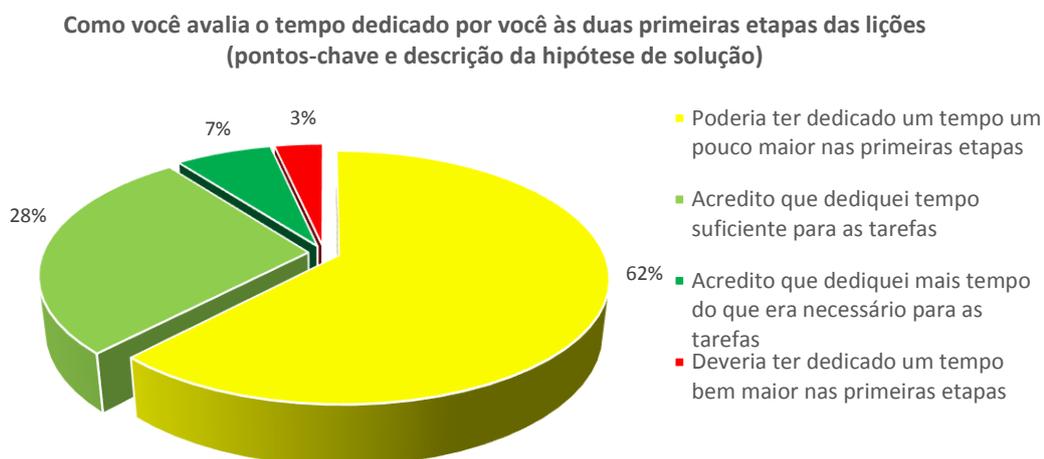


Fonte: Elaborado pelo autor

6.2.2 Respostas para questões relacionadas ao esforço

Nesta seção serão apresentados e analisados os resultados para as questões que foram inseridas com o objetivo de verificar como o estudante reconhece o seu esforço durante a resolução das atividades. As questões focam em três aspectos essenciais, os tempos, o uso de recursos e a qualidade ou na visão do aluno a crença que ele se esforçou o suficiente para detalhar o problema.

Figura 81 - Respostas de alunos com alto esforço para a questão 5 (tempo dedicado)



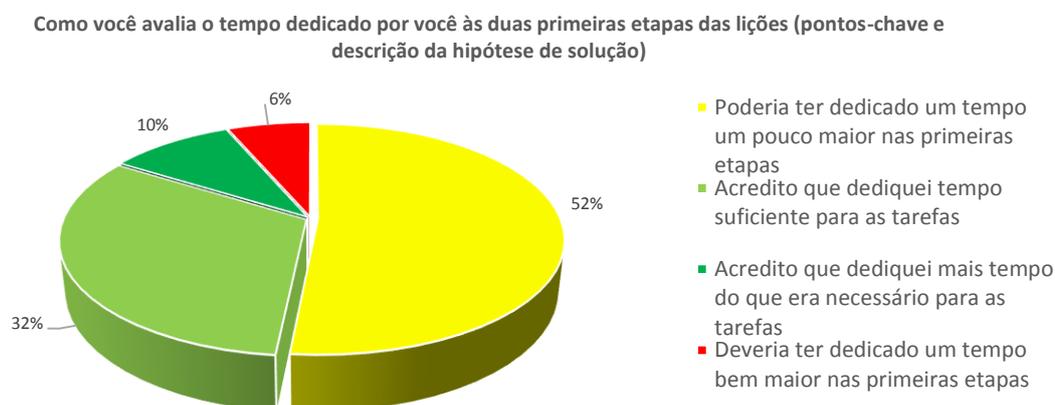
Fonte: Elaborado pelo autor

A questão 5 teve como objetivo verificar como o estudante avalia o tempo que ele dedicou à tarefa, se o tempo foi suficiente ou se poderia ter se dedicado mais. Os resultados (Figuras 81 e 82), demonstram um paradoxo, estudantes classificados com alto esforço indicam que poderiam ter se dedicado mais (62%) enquanto no grupo de baixo esforço o percentual para esta mesma resposta é de 52%. Observa-se ainda que um percentual maior de estudantes com

baixo esforço entendem que se dedicaram o suficiente (32%) contra 28% no grupo que foi classificado com alto esforço.

Embora os números não difiram tanto de um grupo para outro, os resultados confrontam a categorização adotada no modelo. Outra interpretação possível é que os discentes nem sempre conseguem avaliar claramente se os tempos são suficientes para realização da tarefa, tendência que pode ser percebida com mais força em estudantes com menor esforço. Tal constatação pode confirmar a percepção de professores, a qual mostra que estudantes que têm maior facilidade ou melhor desempenho são aqueles que participam mais e se dedicam cada vez mais para as tarefas.

Figura 82 - Resposta de alunos com baixo esforço para a questão 5 (tempo dedicado)



Fonte: Elaborado pelo autor

Sob a ótica da categorização dos dados usados no treinamento das RBs, os resultados podem indicar que é preciso revisar os atributos e critérios adotados na classificação e que os tempos sozinhos não são suficientes para determinar o esforço. Por outro lado é possível estabelecer uma discussão sobre a qualidade do autorrelato para avaliar o esforço empreendido pelo estudante na resolução do problema, uma vez que nem sempre os alunos percebem ou relatam de forma adequada aspectos relacionados ao esforço empreendido.

A questão 6 serve como complemento para a anterior, permitindo detectar possíveis inconsistências no relato do estudante e os motivos que ele alega para não ter dedicado mais tempo. Um tempo menor pode estar relacionado com uma confiança maior ou à dificuldade de compreender a tarefa, o que requer uma avaliação por parte do professor e ações para reduzir as dificuldades do estudante.

Figura 83 - Respostas dos que tem esforço alto para a questão 6 (tempo dedicado)

Considerando ainda o tempo dedicado às duas primeiras etapas. Você poderia ter dedicado um tempo maior? Qual o motivo de não ter dedicado mais tempo?

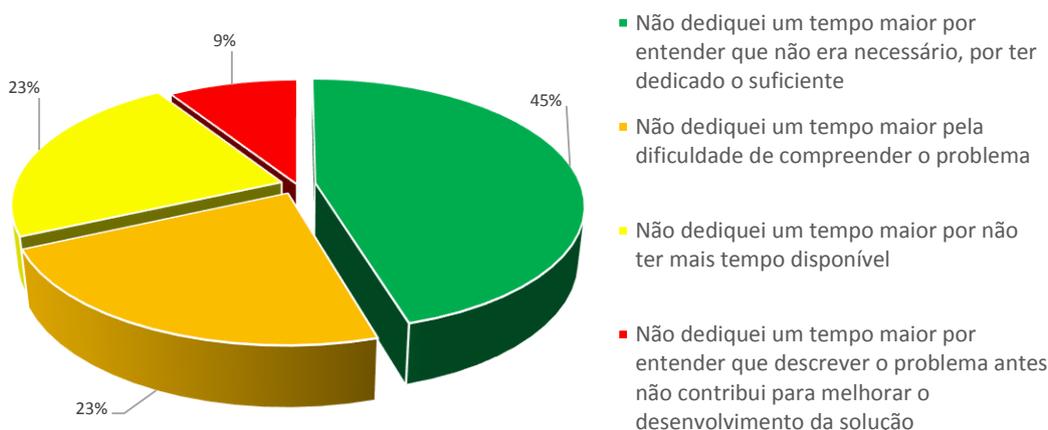


Fonte: Elaborado pelo autor

Os resultados exibidos nas figuras 83 e 84 apontam para uma questão importante e que confrontam os dados da questão anterior, na qual um alto percentual de estudantes com baixo esforço indicou que dedicou tempo suficiente. Consta-se que 55% do grupo de alunos com baixo esforço não dedicou mais tempo pelas dificuldades, por alegar que não teve tempo disponível ou por não concordar com o método adotado. Entre os participantes que foram categorizados com alto esforço, 70% está convicto de que não era necessário dedicar um tempo maior.

Figura 84 - Respostas dos que tem esforço baixo para a questão 6 (tempo dedicado)

Considerando ainda o tempo dedicado às duas primeiras etapas. Você poderia ter dedicado um tempo maior? Qual o motivo de não ter dedicado mais tempo?



Fonte: Elaborado pelo autor

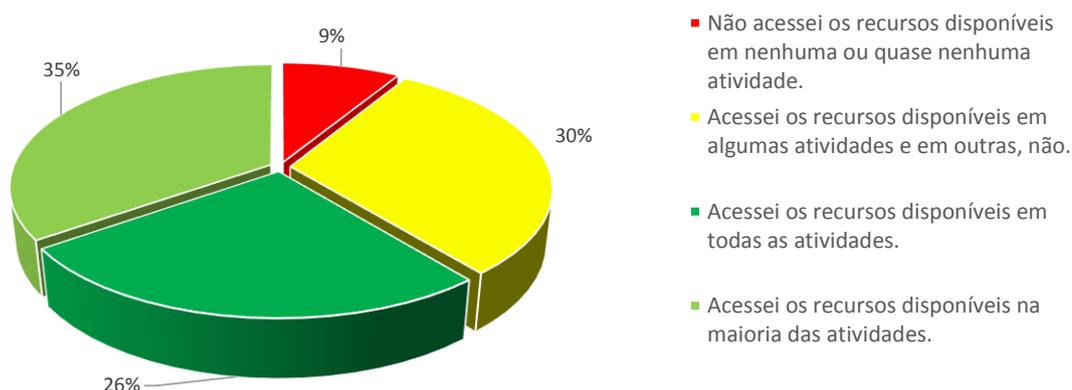
Uma análise conjunta das duas questões contribui para indicar que os tempos são importantes e que estudantes que apresentam tempos menores podem estar com dificuldades para compreender a tarefa ou desmotivados. A alegação de que não teve tempo suficiente pode

ser apenas uma maneira de não reconhecer a importância da dedicação ou de mascarar uma possível falta de interesse ou confiança. Os resultados para esta questão reforçam a ideia de que o autorrelato por si só pode não ser suficiente para determinar o esforço do estudante.

Um aspecto que deve ser considerado no esforço do estudante é o fato de o mesmo ter buscado materiais de apoio, ter pesquisado e utilizado recursos que poderiam auxiliá-lo a resolver o problema. A metodologia da problematização (BERBEL, 2012; BERBEL e GAMBOA, 2011) que serviu de base para o método adotado propõe uma etapa denominada teorização, na qual devem ser buscados subsídios para a elaboração da solução. Sendo assim, foi aplicada uma questão que avaliava a percepção do estudante quanto a esta busca, à capacidade de mobilizar recursos para realizar a atividade (Figuras 85 e 86).

Figura 85 - Respostas dos que tem esforço alto para a questão 7 (recursos)

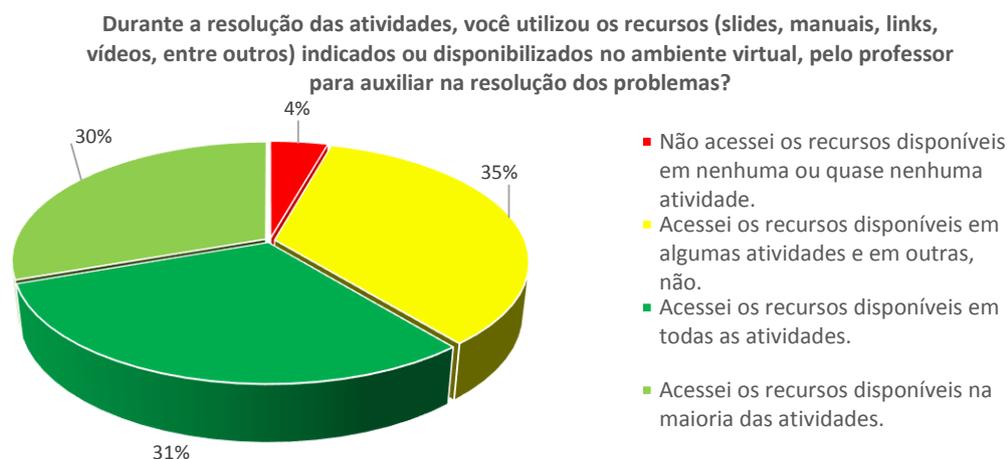
Durante a resolução das atividades, você utilizou os recursos (slides, manuais, links, vídeos, entre outros) indicados ou disponibilizados no ambiente virtual, pelo professor para auxiliar na resolução dos problemas?



Fonte: Elaborado pelo autor

Os resultados mostram que não existem diferenças significativas para os dois grupos de estudantes, uma vez que a soma dos percentuais dos que acessaram os recursos em todas ou na maioria das atividades é de 61% nos dois casos. É possível inferir que esta questão talvez não seja um indicativo claro do esforço ou que o acesso aos recursos e materiais não é tão relevante para determinar a motivação. Em virtude da situação verificada nesta questão, não serão detalhados os dados da questão 8, que era um complemento à pergunta 7 e que apresentou dados semelhantes.

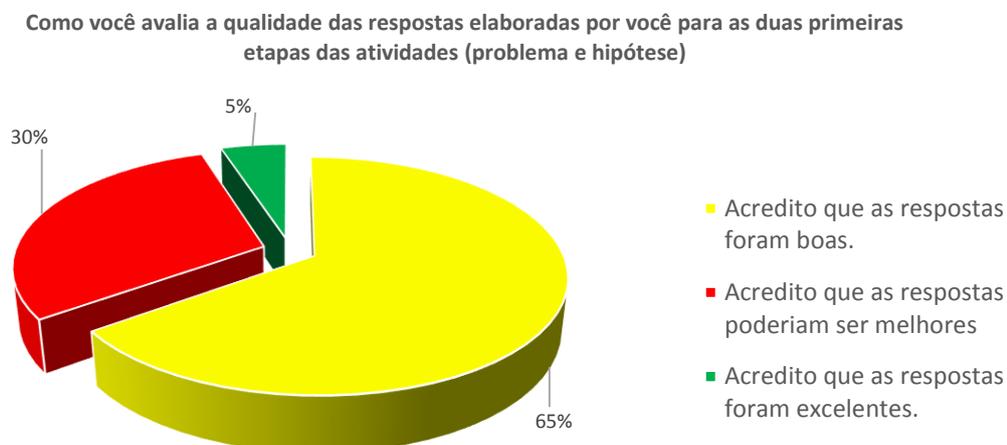
Figura 86 - Respostas dos que tem esforço baixo para a questão 7 (recursos)



Fonte: Elaborado pelo autor

Um relato sobre a percepção que o próprio estudante tem sobre a qualidade das respostas que ele elaborou, especialmente para o problema e para a hipótese foi obtido com as respostas para a questão 9. Em função do modelo afetivo ter sido concebido com base no método da problematização e apresentar uma ênfase na especificação do problema, a questão citava as duas primeiras etapas e as respostas para as mesmas.

Figura 87 - Respostas dos que tem esforço alto para a questão 9 (qualidade das respostas)



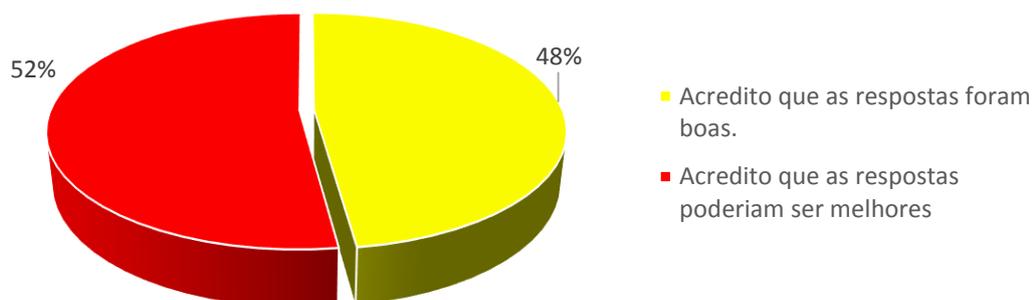
Fonte: Elaborado pelo autor

Entre os estudantes categorizados com baixo esforço, nenhum indicou que as respostas foram excelentes e mais da metade reconheceu que poderiam ser melhores (figura 88). Já no primeiro grupo, cujos resultados são mostrados na figura 87, um número superior de alunos entendeu que as respostas foram boas e 5% acreditou que foram excelentes. Nesta questão percebe-se que há uma tendência de estudantes com esforço mais reduzido reconhecerem que

talvez poderiam ter apresentado soluções melhores, o que pode indicar indiretamente que na visão destes o esforço poderia ter sido maior.

Figura 88 - Respostas dos que tem esforço baixo para a questão 9 (qualidade das respostas)

Como você avalia a qualidade das respostas elaboradas por você para as duas primeiras etapas das atividades (problema e hipótese)

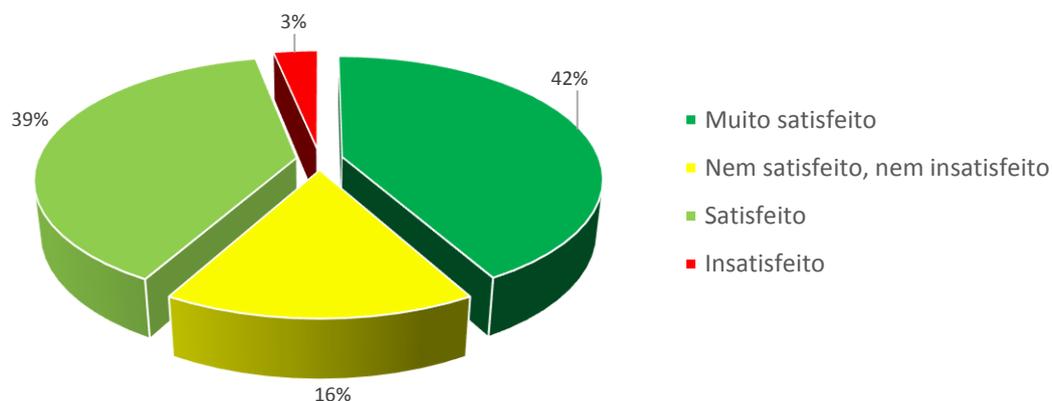


Fonte: Elaborado pelo autor

Os resultados das respostas para a satisfação dos usuários que foi segmentado conforme a confiança também foi estratificado levando em consideração o nível de esforço aferido no modelo. A análise destes resultados é fundamental pois uma maior satisfação normalmente implica em uma participação mais ativa, podendo acarretar em maior motivação dos estudantes. O envolvimento e postura ativa do estudante é o objetivo central da Aprendizagem Ativa (BORDENAVE, 1999; BERBEL, 2011).

Figura 89 - Resposta dos que tem esforço alto para a questão 15 (satisfação)

Avalie o seu grau de satisfação relacionado as atividades desenvolvidas nas últimas semanas



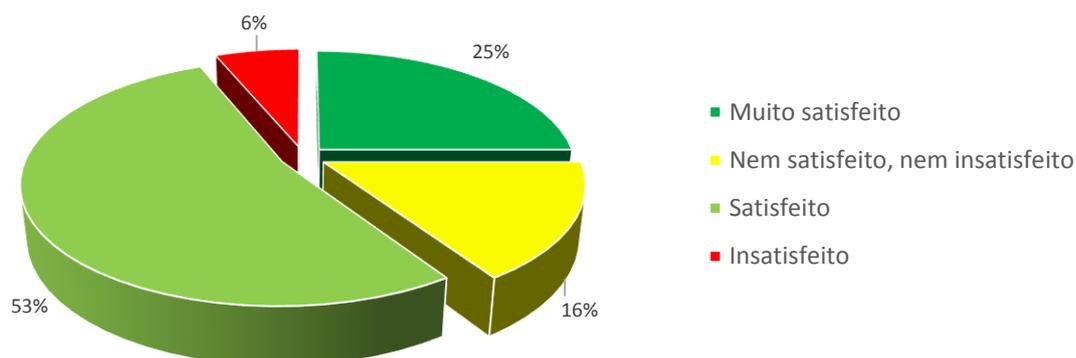
Fonte: Elaborado pelo autor

Os percentuais mostrados nas figuras 89 e 90 mostram um maior índice associado ao maior índice de satisfação (42%) entre os estudantes com alto esforço, se comparado ao grupo de nível baixo de esforço (25%). O cenário é bastante semelhante ao mostrado nos grupos de

alta e baixa confiança, demonstrando que alunos com maiores níveis de esforço e confiança relatam que estão mais satisfeitos com as atividades. Tal constatação permite inferir que ações que visam estimular a confiança e o esforço dos estudantes podem indiretamente incrementar a satisfação destes.

Figura 90 - Resposta dos que tem esforço baixo para a questão 15 (satisfação)

Avalie o seu grau de satisfação relacionado as atividades desenvolvidas nas últimas semanas



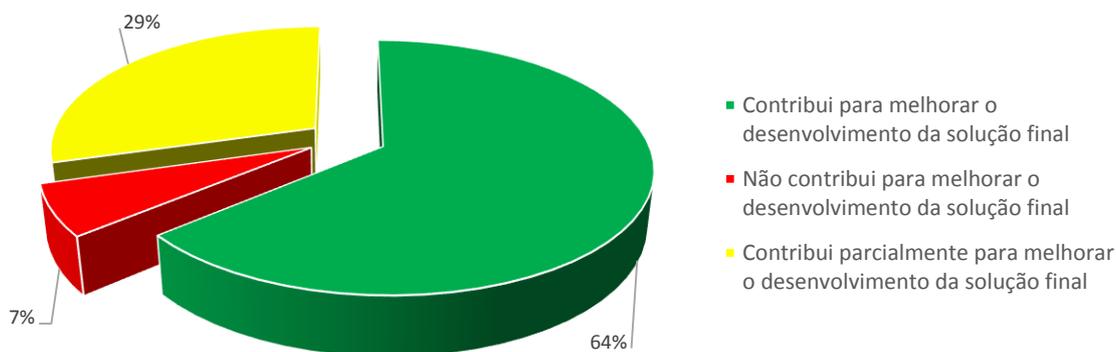
Fonte: Elaborado pelo autor

6.2.3 Questões para avaliar a satisfação quanto ao método e o sistema

Além das questões associadas ao esforço e à confiança foram inseridas também perguntas que tiveram como objetivo aferir a satisfação quanto ao método adotado e ao sistema PROALG utilizado no estudo de caso. Os resultados para estas questões não serão segmentados de acordo com o esforço, nem a confiança, portanto, os gráficos ilustram as respostas de todos os participantes.

Figura 91 – Respostas para a questão 3 (problematização)

Como você avalia a estratégia que foi usada, que propõe descrever o problema e a proposta de solução antes de escrever o código.

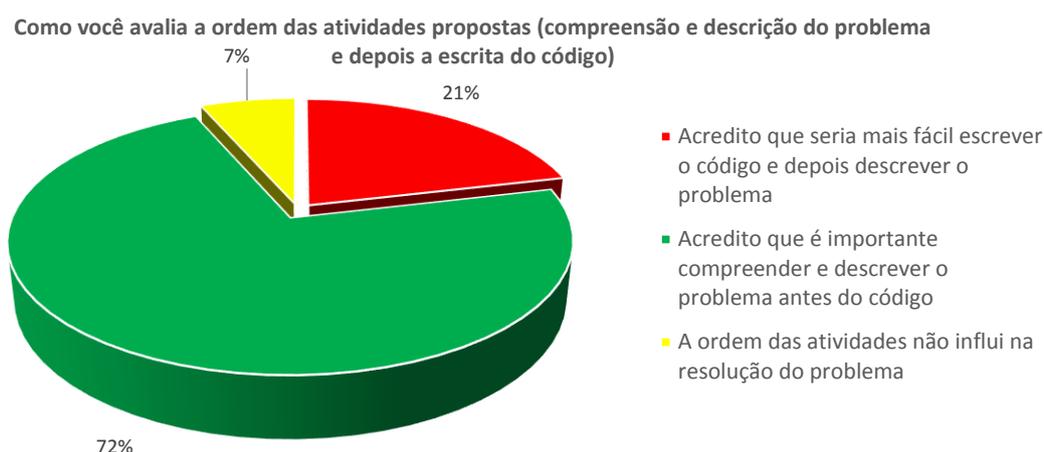


Fonte: Elaborado pelo autor

Foram propostas duas questões para avaliar a percepção dos estudantes quanto ao método baseado na problematização, a primeira focava na contribuição para a elaboração da solução e a segunda, na preferência do estudante quanto à ordem das etapas para solução. Analisando os dados dos gráficos (Figuras 91 e 92) que a ampla maioria acredita que é importante compreender e descrever o problema e que esta compreensão contribui para uma melhorar a solução final.

Como era de se esperar, ainda existe um conjunto de estudantes (21%) que acredita que seria melhor ou mais fácil escrever primeiro o código e depois descrever o problema, estratégia oposta a que foi adotada. Os percentuais demonstram que há um pequeno contingente de estudantes que acreditam que a compreensão do problema é importante, mas que se sentem mais confortáveis com um método clássico baseado somente na escrita do código.

Figura 92 - Respostas para a questão 3 (problematização)



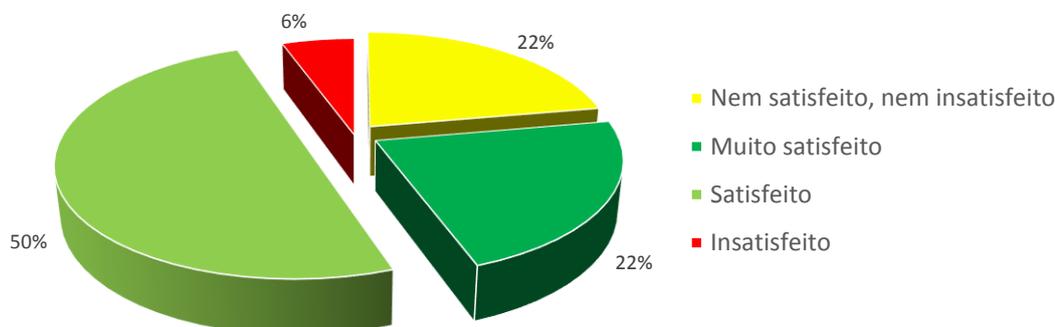
Fonte: Elaborado pelo autor

Uma das contribuições desta tese é o desenvolvimento e aplicação de uma ferramenta de apoio para a utilização do método baseado na problematização e para o acompanhamento da motivação dos estudantes. O sistema denominado PROALG foi apresentado no capítulo 3, tendo sido fundamental nas atividades realizadas no estudo de caso. O questionário aplicado nas duas fases do estudo de caso incluiu uma questão que teve como objetivo avaliar a satisfação associada ao uso do referido software.

O gráfico mostrado na figura 93 permite concluir que os estudantes ficaram bastante satisfeitos com o sistema, mesmo que ao longo das atividades tenham se manifestado com sugestões de ajustes e melhorias que poderiam ser implementadas. Dentre os participantes 72% se declararam satisfeitos ou muito satisfeitos e apenas 6% responderam que ficaram insatisfeitos com a ferramenta.

Figura 93 - Respostas para a questão 13 (sistema PROALG)

Além do ambiente virtual Moodle, durante as atividades foi utilizado um sistema chamado PROALG. Como você classificaria o seu grau de satisfação relacionado ao uso deste sistema?



Fonte: Elaborado pelo autor

6.2.4 Respostas relacionadas a Roda dos Estados Afetivos (REA)

A REA (Roda dos Estados Afetivos) proposta por Tran (2004), Scharin, Schlegel, Scherer (2012) e adaptada por Longhi (2011) e Santos (2016), foi utilizada como base para um conjunto de questões para avaliação dos estados afetivos demonstrados pelos estudantes ao longo das atividades de cada semestre.

Os estados afetivos dividem-se em quatro quadrantes que representam maior satisfação e maior controle (quadrante 2), menor satisfação e maior controle (quadrante 1), maior satisfação com menor controle (quadrante 3) e um menor controle aliado a maior satisfação (quadrante 4). Segundo Santos (2016) e Longhi (2011) valores mais intensos no quadrante 1 indicam maior insatisfação, ao contrário do quadrante 2 que indica uma satisfação mais intensa. O quadrante 3 está associado mais diretamente ao desânimo ou à falta de motivação, enquanto o 4 a um nível de motivação maior.

O questionário foi aplicado nas duas fases do estudo de caso, tendo sido respondido por 46 estudantes, de um total de 57 estudantes que participaram das atividades no estudo de caso, ou seja, aproximadamente, 85% dos alunos responderam as questões associadas a REA. No estudo piloto o questionário não foi utilizado, porém após avaliar os resultados do questionário aplicado no experimento inicial constatou-se a possibilidade de complementar as questões com uma análise mais direta dos estados afetivos relatados pelos discentes.

Sendo assim os resultados para as questões da REA serviram de complemento às respostas para as demais questões descritas na seção 7.3.2. Seguindo o mesmo padrão de avaliação, os estudantes foram segmentados em grupos de acordo com a categorização da confiança e esforço utilizadas na elaboração do modelo para reconhecimento da motivação.

Utilizando esta categorização foi possível avaliar se determinados quadrantes ou estados apresentavam maior intensidade nas respostas dos alunos com maior ou menor nível de confiança e esforço.

Como é possível perceber no apêndice B, as respostas seguiram uma escala de intensidade, sendo 1 correspondente a intensidade mais baixa e 5, mais alta. As médias e o coeficiente de variação para cada item e quadrante são apresentados na tabela 28.

Tabela 27 – Resultados da aplicação da REA

Sentimento	Confiança				Esforço			
	Alto		Baixo		Alto		Baixo	
	Média	Coef.var.	Média	Coef.var.	Média	Coef.var.	Média	Coef.var.
Desvalorizado	1,58	0,63	1,83	0,60	1,84	0,61	1,56	0,61
Aversão	2,00	0,46	2,08	0,47	2,30	0,37	1,93	0,47
Irritado	1,94	0,64	2,41	0,51	2,00	0,50	2,25	0,57
Ciúme	1,11	0,43	1,33	0,48	1,23	0,48	1,18	0,45
Indiferente	1,94	0,56	2,16	0,47	1,84	0,48	2,18	0,50
Quadrante 1	1,71	0,59	1,96	0,53	1,83	0,51	1,82	0,57
Orgulhoso	3,41	0,31	3,16	0,18	3,30	0,33	3,31	0,21
Entusiasmado	3,82	0,24	3,66	0,33	3,92	0,26	3,62	0,30
Interessado	4,11	0,16	4,16	0,26	4,07	0,18	4,18	0,23
Feliz	3,58	0,27	3,41	0,29	3,69	0,25	3,37	0,30
Quadrante 2	3,73	0,25	3,60	0,29	3,75	0,26	3,62	0,27
Triste	1,88	0,61	2,83	0,44	2,00	0,64	2,50	0,50
Envergonhado	1,64	0,71	1,66	0,59	1,69	0,73	1,62	0,58
Desapontado	1,70	0,57	2,00	0,47	1,84	0,57	1,81	0,50
Medo	1,47	0,54	2,41	0,67	1,84	0,69	1,87	0,68
Quadrante 3	1,67	0,61	2,22	0,57	1,84	0,64	1,95	0,58
Surpreso	3,17	0,27	3,58	0,14	3,15	0,31	3,50	0,14
Expectativa	3,11	0,40	3,41	0,26	3,23	0,28	3,31	0,32
Tranquilo	3,58	0,34	3,00	0,34	3,46	0,34	3,18	0,34
Satisfeito	3,82	0,26	3,66	0,24	3,84	0,27	3,56	0,25
Quadrante 4	3,42	0,32	3,41	0,25	3,42	0,30	3,39	0,27

Fonte: Elaborado pelo autor

Analisando os dados por quadrante, observa-se que as médias dos quadrantes que indicam sentimentos negativos são significativamente menores do que nos outros quadrantes, o que permite concluir que a grande maioria dos estudantes relatou valências mais positivas, além de um maior controle. O quadrante 2 que reflete maior satisfação e controle é o que apresenta médias mais altas em todos os grupos de estudantes.

Comparando os segmentos é possível observar maiores diferenças entre alunos com maior ou menor confiança, como por exemplo, no quadrante 3, no qual o grupo com baixa confiança apresentou média mais alta (2,22) em relação aos alunos com alta confiança (1,67). É importante ressaltar que em ambos os grupos o coeficiente de variação se aproxima de 60%

o que indica um alto grau de variação nas respostas, o que se verifica também para estudantes com alto e baixo esforço, embora os valores das médias para o esforço sejam mais próximos (1,84 e 1,95). Estes percentuais permitem concluir que estudantes com níveis mais baixos de confiança e esforço indicam podem apresentar uma valência com tendência mais negativa, além de um menor nível de controle.

Os quadrantes 2 e 4 que refletem aspectos mais positivos, com maior satisfação e motivação apresentam médias próximas em todos os grupos, especialmente nas categorias associadas ao esforço. A análise das médias permite inferir que não existem diferenças significativas no autorrelato dos estudantes com baixo ou alto esforço, sendo que os dois grupos apresentaram intensidade relativamente alta nas respostas.

O quadrante 1 exibe médias com diferenças mais significativas no fator de confiança, uma vez que estudantes com baixa confiança possuem média maior (1,96) se comparados aos demais (1,71). Já no caso do esforço, não há diferença nos valores das médias, nos dois grupos o valor é praticamente o mesmo (1,82 e 1,83). Neste item também constata-se uma maior variabilidade nas respostas, com um coeficiente de variação está entre 50% e 60%.

Entre os sentimentos positivos, merecem destaque o interesse, o entusiasmo e a satisfação, uma vez que estes apresentaram maior intensidade em todos os segmentos de estudantes, com valores próximos a 4 e coeficiente de variação em torno de 20%. Tais médias contribuem para demonstrar que a tecnologia educacional aplicada, com um método baseado na problematização, combinada com um software de apoio foi considerada positiva pelos alunos.

Os aspectos negativos que aparecem com mais intensidade normalmente estão no segmento de estudantes com baixa confiança e em alguns casos, nos que foram categorizados com menor esforço. Medo (2,41), tristeza (2,83) e irritação (2,41) aparecem na categoria de baixo esforço com valores mais altos se comparados ao grupo de alunos com alta confiança e esforço. A irritação (2,25) e a tristeza (2,50) também são superiores entre os estudantes com baixo esforço, quando se estabelece uma comparação com os demais.

As menores variações nos segmentos relacionados ao esforço segue uma tendência verificada também nas respostas para as demais questões descritas na seção 6.3.2, onde verificou-se que um nível mais alto ou baixo de esforço, de acordo com o modelo não é tão claramente percebido no autorrelato. No caso da confiança, percebe-se de forma mais direta uma correlação entre a categorização do modelo e as respostas dos alunos, tanto nas questões específicas, quanto na REA.

7 CONCLUSÕES

O trabalho desenvolvido e descrito nesta tese teve como principal objetivo investigar a possibilidade de utilizar uma estratégia de ensino e aprendizagem ativa, considerando a dimensão afetiva. O foco principal recaiu sobre as dificuldades apresentados por estudantes nas disciplinas de programação, especialmente aquelas que ocorrem no início do curso. A análise dos pressupostos teóricos e trabalhos relacionados mostrou que muitas das dificuldades no processo de ensino e aprendizagem de algoritmos e programação estão associadas à motivação dos estudantes.

Foi proposto e aplicado um método baseado nos fundamentos da problematização com o arco de Maguerez (BERBEL, 2012), que foi composto por um conjunto de etapas que incluíram a descrição dos pontos-chave do problema, uma hipótese de solução, antes da elaboração da solução final, o código-fonte. A realização de um estudo piloto foi essencial para constatar a viabilidade do uso do método e a possibilidade de coletar dados que fossem utilizados posteriormente no desenvolvimento de um modelo para o reconhecimento da motivação dos discentes.

O método foi aplicado com êxito e contribuiu para que os estudantes refletissem sobre o problema e especificassem o mesmo antes da elaboração do código fonte. Os resultados do autorrelato demonstraram que os alunos compreenderam a importância das etapas que solicitavam a descrição dos pontos chave e da hipótese. Desta forma, conclui-se que os princípios da Aprendizagem Ativa foram assimilados e contribuíram para desenvolver habilidades relacionadas à resolução de problemas.

Ainda com relação ao método baseado na problematização é possível inferir que o mesmo contribuiu para o desenvolvimento das habilidades relacionadas à abstração e ao pensamento computacional, o que demonstra o alinhamento da pesquisa com preocupações e iniciativas recentes (HAZZAN e KRAMER, 2016; BRACKMANN et al., 2019). Entretanto o trabalho desenvolvido não teve como objetivo comprovar esta influência e tampouco medir o impacto do método aplicado no aprendizado dos estudantes.

Com o intuito de suportar a utilização do método e a coleta de dados foi implementado um sistema denominado PROALG, que contribuiu também para manter uma base de atividades que podem ser reutilizadas em diferentes turmas e semestres. Os requisitos definidos para o sistema se originaram das necessidades e objetivos desta pesquisa, além das experiências e resultados do estudo piloto. A aplicação e avaliação do sistema foi importante para atender de forma plena diversos objetivos específicos propostos na pesquisa, entre eles o desenvolvimento

e validação de um sistema para a resolução de problemas para automatizar a coleta de dados e a aplicação do modelo associado à motivação.

Após o experimento inicial foi realizado um estudo de caso, composto por duas turmas de estudantes, no qual foi utilizado método da problematização, além do sistema de apoio desenvolvido. Os dados coletados durante esta etapa, combinados com os do estudo piloto permitiram a definição de um conjunto de atributos que foram utilizados na construção do modelo para o reconhecimento da motivação. A adoção de uma metodologia baseada em estudo de caso se mostrou adequada, uma vez que foi possível observar o comportamento dos participantes e obter registros das atividades realizadas por estes. Verificou-se que os tempos, níveis de detalhe e o acesso aos recursos de ajuda se constituiriam em informações importantes para o reconhecimento do esforço e da confiança dos estudantes.

No desenvolvimento do modelo para o reconhecimento da motivação foi utilizada uma metodologia baseada nas etapas da descoberta de conhecimento em bases de dados (KDD – Knowledge Discovery Database) proposta por Fayad (1996). Os dados foram extraídos dos logs e da base de dados do sistema PROALG passaram por um pré-processamento e transformação para gerar os valores utilizados no modelo, esta transformação foi essencial para a construção das Redes Bayesianas.

Técnicas para classificação supervisionada, com ênfase nas Redes Bayesianas foram aplicadas para determinar os níveis de esforço e confiança em cada atividade e para cada estudante. A aplicação das Redes Bayesianas foi importante para determinar a intensidade associada aos níveis de esforço e confiança, contribuindo para identificar os casos de estudantes que exibiam níveis mais baixos e, portanto, deveriam ser acompanhados de forma mais intensa pelo professor.

A implementação do protótipo que utilizou as redes para reconhecer os níveis de motivação permitiu a realização de testes em uma nova turma de estudantes, na qual foi verificada a validade do modelo e que o mesmo pode ser aplicado em qualquer momento durante o semestre. O uso de redes bayesianas conferiu uma alta flexibilidade ao modelo, uma vez que a estrutura e dependências entre as variáveis podem ser ajustadas sem a necessidade de maiores alterações na implementação do módulo para reconhecimento da motivação.

Para ajustar as probabilidades, basta uma atualização do conjunto de registros e um novo treinamento para que o sistema retorne diferentes valores de crença para as variáveis que correspondem à confiança e esforço. Após o treinamento o módulo para acompanhamento da

motivação passa a utilizar as novas configurações de forma automática, uma vez que este acesso o arquivo do software Netica que armazena as probabilidades da rede.

Outro aspecto importante do módulo para acompanhamento da motivação diz respeito ao uso de gráficos que representaram os níveis de confiança e esforço. Ao demonstrar os resultados ao professor da disciplina foi possível perceber a satisfação quanto à visualização dos dados e a rápida compreensão das informações. Prontamente foi constatado de forma visual que estudantes que na percepção do docente apresentavam dificuldades e exibiam um comportamento pouco ativo apareciam com baixos níveis de motivação.

A validação junto ao docente mostrou ainda que o uso do modelo para reconhecimento da motivação em muitos casos corrobora com a impressão do professor sobre alguns estudantes, reafirmando a necessidade de ações pedagógicas que estimulem a confiança e o esforço. As informações sobre a motivação apontaram também situações de estudantes que demonstraram baixos níveis de motivação e que na percepção do professor apresentavam bom desempenho. Em tais casos o modelo afetivo sinaliza que pode existir algum problema, cabendo ao professor realizar um acompanhamento nas atividades seguintes para verificar se esta situação se mantém.

A partir da interação com o professor foi possível concluir, portanto, que o objetivo de combinar problematização com um modelo para reconhecimento da motivação foi atingido, principalmente com o desenvolvimento e validação do módulo citado. Na visão do professor da disciplina, as informações apresentadas contribuíram fortemente para que este confirmasse as percepções sobre alguns estudantes e adotasse ações que poderiam melhorar a aprendizagem dos mesmos.

Por fim, foram apresentados os resultados do autorrelato dos estudantes, das respostas para os questionários e para a REA, ambos aplicados no final de cada etapa. Foi possível perceber um grau elevado de satisfação dos estudantes com as atividades, além de respostas que indicaram o reconhecimento da importância de refletir e descrever o problema antes de elaborar a solução final. É possível concluir que o método da problematização foi bem recebido pelos estudantes e contribuiu para que estes se sentissem interessados, satisfeitos.

Também foi solicitada a avaliação do sistema PROALG que foi um dos resultados apresentados nesta tese. A avaliação da ferramenta foi igualmente positiva, embora ao longo das atividades os estudantes tenham se manifestado com sugestões de ajustes que foram realizados e contribuíram para melhorar a utilização do software a partir da segunda fase do estudo de caso.

A análise dos indicadores segmentados de acordo com os níveis de confiança e esforço contribuiu para verificar se a categorização usadas nos dados que serviram de base para o treinamento do modelo era minimamente confiável. Respostas de estudantes com baixa confiança apresentaram uma relação mais direta com a classificação do modelo, com uma tendência por parte deste grupo que relatar menor crença na capacidade de resolver as atividades, além de menor satisfação. Observou-se também que estes reconheceram o acesso às dicas e que estas foram relevantes para a resolução dos exercícios.

Uma hipótese possível é que uma baixa confiança é reconhecida e admitida com mais facilidade pelo estudante, enquanto um baixo esforço nem sempre é reconhecido. O estudante percebe que o acesso aos recursos de ajuda combinado com interrupções na tarefa e uma avaliação de maior grau de dificuldade no exercício pode demonstrar que ele não está tão confiante.

Com relação ao esforço não foi possível perceber diferenças tão significativas entre a categoria (baixo, alto) e o perfil das respostas, tanto nas questões específicas, quanto no questionário baseado na REA. Em ambos, foi possível constatar um nível de satisfação um pouco maior em alunos com alto nível de esforço, o que pode indicar que um baixo esforço pode estar associado a menor satisfação.

O fator tempo, relacionado ao esforço, nem sempre é percebido por um estudante da mesma maneira que outro, enquanto alguns acreditam que um tempo baixo é suficiente, outros crêem que necessitam de uma dedicação maior. Nesta caso a análise conjunta do autorrelato com a categorização dos tempos que indicou os que estavam abaixo dos demais permitiu identificar os alunos que foram menos perseverantes na resolução das atividades.

De maneira geral os resultados do autorrelato permitem concluir que a categorização utilizada no modelo afetivo foi adequada, embora seja importante em um futuro próximo reavaliar atributos do esforço, especialmente os que representam os tempos das tarefas. Ajustes na categorização ou na influência destas variáveis na determinação do nível de esforço podem tornar o modelo mais adequado ao que relatam os próprios estudantes.

A partir das considerações expostas é possível concluir que os objetivos definidos nesta tese foram atingidos de forma plena ou parcial e que o problema da pesquisa foi resolvido de forma satisfatória. O desenvolvimento de um método baseado na problematização, combinado com um modelo computacional para reconhecimento da motivação, ambos suportados por um sistema automatizado permitiu a resolução do problema proposto.

7.1 Limitações da tese

Uma dos aspectos a serem ressaltados quanto ao método adotado e aos dados coletados é que a colaboração e interação entre os estudantes e destes com o professor não foi registrada de forma explícita e não foi contemplada integralmente no modelo afetivo. Embora esta interação tenha ocorrido e tenha sido estimulada pelo professor as informações relacionadas á colaboração não foram gravadas pelo software PROALG.

Esta restrição foi adotada após o estudo piloto, no qual foram disponibilizados fóruns e recursos para que os estudantes registrassem dúvidas e respondessem aos colegas. Constatou-se que os alunos preferiam utilizar recursos de comunicação alternativos ou uma interação presencial direta com o professor, o que dificultaria a coleta de dados relativa a esta interação. Além disso a implementação de recursos de interação no PROALG aumentaria significativamente a complexidade do software e o tempo de desenvolvimento.

Entre as limitações relacionadas ao modelo para reconhecimento da motivação destaca-se a utilização somente dos fatores de confiança e esforço, sem uma representação separada para a independência. Conforme foi ressaltado ao longo da tese, as variáveis utilizadas para reconhecimento da confiança também podem indicar a independência do estudante. A construção de um componente específico para reconhecer a independência baseado nos mesmos atributos da confiança ou na maioria destes não agregaria informações significativas para o reconhecimento da motivação e em contrapartida aumentaria a complexidade modelo.

Uma das principais limitações verificadas na construção do modelo computacional para reconhecimento da motivação foi o reduzido número de registros disponíveis para o treinamento das redes bayesianas, especialmente no reconhecimento da motivação dos estudantes. Cabe ressaltar, porém que este fato não afetou a flexibilidade das redes e do módulo para acompanhamento da motivação, pois a atualização das probabilidades pode ser realizada a qualquer momento, não exigindo alterações na implementação do referido módulo. Com a utilização do sistema em novas turmas será possível obter um conjunto maior de dados que permitirão um retreinamento das RB, o que irá acarretar na atualização das probabilidades associadas ao esforço e a confiança.

Por fim, é importante reforçar que não foi possível integrar o módulo de acompanhamento ao sistema PROALG, sendo assim, esta integração pode ser apontada como uma limitação, mas também um trabalho futuro na continuidade da pesquisa. Cabe ressaltar porém, que os resultados mostrados na validação do modelo são oriundos da base de dados do sistema, o que permite constatar que não houve prejuízo na análise e validação do modelo.

A validação do modelo afetivo se limitou a análise dos dados em conjunto com o professor, que por sua vez foi responsável por definir ações pedagógicas junto aos estudantes. A avaliação dos resultados destas ações e a contribuição das mesmas para a melhora do aprendizado não estavam entre os objetivos da pesquisa, desta forma, não foi realizado um acompanhamento destas ações.

7.2 Trabalhos futuros

A continuidade das pesquisas desenvolvidas nesta tese pode ser direcionada para uma inserção mais direta dos dados relacionados à colaboração no modelo afetivo, com a definição de novas variáveis e de um componente para determinação da independência do estudante. Para tanto serão necessários recursos no sistema PROALG para capturar os dados da interação entre alunos, professores, além de um estímulo para que estas funcionalidades sejam utilizadas.

A integração completa do protótipo do módulo para acompanhamento com o sistema PROALG já foi indicada ao longo do texto como um possível trabalho futuro, uma vez que não foi possível implementar o módulo diretamente no software. Esta integração facilitaria o uso do módulo por parte do docente, embora seja importante ressaltar que o protótipo permitiu validar os resultados e acessar os dados reais das tarefas desenvolvidas.

A atualização das Redes Bayesianas com a realização de treinamentos com um número maior de registros, especialmente no número de estudantes é outra ação importante. Além dos ajustes nas probabilidades é possível realizar mais simulações com inserção de novos atributos e novas dependências entre os atributos já existentes.

Pode-se citar também como trabalho futuro a sugestão do professor da disciplina de manter armazenada uma posição da motivação do estudante, gerada em determinado momento para comparação com os níveis em outro instante, após a realização de novas atividades. Comparando os resultados em diferentes momentos do componente curricular, ao longo de um semestre seria possível verificar o impacto das ações e das atividades subsequentes na motivação dos estudantes.

8 REFERÊNCIAS

- ARMONI, Michal; MEERBAUM-SALANT, Orni; BEN-ARI, Mordechai. **From scratch to “real” programming**. ACM Transactions on Computing Education (TOCE), v. 14, n. 4, p. 25, 2015.
- ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. **Fundamentos da programação de computadores: algoritmos, Pascal, C/C++ (padrão ANSI) e Java**. 3. ed. São Paulo: Pearson, 2012.
- ASSESPRO. **Evasão de alunos reflete na escassez de mão de obra**. Disponível em <http://assespro.org.br/na-midia/clipping/2013-01-15-evasao-de-alunos-reflete-na-escassez-de-mao-de-obra/>. Acessado em 30 de agosto de 2017.
- AU-YONG-OLIVEIRA, Manuel et al. **The social impact of technology on millennials and consequences for higher education and leadership**. Telematics and Informatics, v. 35, n. 4, p. 954-963, 2018.
- AURELIANO, Viviane Cristina Oliveira; TEDESCO, Patrícia Cabral de Azevedo Restelli. **Ensino-aprendizagem de Programação para Iniciantes: uma Revisão Sistemática da Literatura focada no SBIE e WIE**. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2012.
- BAKER, Ryan; ISOTANI, Seiji; CARVALHO, Adriana. **Mineração de dados educacionais: Oportunidades para o brasil**. Brazilian Journal of Computers in Education, v. 19, n. 02, p. 03, 2011.
- BAKER, Ryan Shaun; INVENTADO, Paul Salvador. **Educational data mining and learning analytics**. In: Learning analytics. Springer New York, 2014. p. 61-75.
- BAUMEISTER, Roy F. et al. **Does high self-esteem cause better performance, interpersonal success, happiness, or healthier lifestyles?** Psychological science in the public interest, v. 4, n. 1, p. 1-44, 2003.
- BEHAR, Patricia Alejandra. **Competências em educação a distância**. Penso Editora, 2013.
- BENNEDSEN, Jens; CASPERSEN, Michael E. **Failure rates in introductory programming**. ACM SIGCSE Bulletin, v. 39, n. 2, p. 32-36, 2007.
- BENNEDSEN, Jens; CASPERSEN, Michael E. **Failure rates in introductory programming: 12 years later**. ACM Inroads, v. 10, n. 2, p. 30-36, 2019.
- BENNEDSEN, Jens; CASPERSEN, Michael E. **Abstraction ability as an indicator of success for learning computing science?** In: Proceedings of the Fourth international Workshop on Computing Education Research. ACM, 2008. p. 15-26.
- BERBEL, Neusi Aparecida Navas; GAMBOA, Sílvio Ancízar Sánchez. **A metodologia da problematização com o Arco de Maguerez: uma perspectiva teórica e epistemológica**. Filosofia e Educação, v. 3, n. 2, p. 264-287, 2011.
- BERBEL, Neusi Aparecida Navas. **As metodologias ativas e a promoção da autonomia de estudantes**. Semina: Ciências Sociais e Humanas, v.32, n.1, p.25-40, 2012.

- BERBEL, Neusi Aparecida Navas; GIANNASI, Maria Júlia. **A metodologia da problematização aplicada em curso de educação continuada ea distância**. Editora UEL, 1999.
- BERGIN, Susan; REILLY, Ronan. **Programming: factors that influence success**. In: ACM SIGCSE Bulletin. ACM, 2005. p. 411-415.
- BERGIN, Susan; REILLY, Ronan. **The influence of motivation and comfort-level on learning to program**. 2005b.
- BEZ, J. L.; TONIN, N. A.; RODEGHERI, P. R. **URI Online Judge Academic: A tool for algorithms and programming classes**. In: Computer Science & Education (ICCSE), 2014 9th International Conference on. IEEE. p. 149-152. 2014.
- BLATT, Lucas; BECKER, Valdecir; FERREIRA, Alexandre. **Mapeamento Sistemático sobre Metodologias e Ferramentas de apoio para o Ensino de Programação**. In: Anais do Workshop de Informática na Escola. 2017. p. 815.
- BOLTON, Ruth N. et al. **Understanding Generation Y and their use of social media: a review and research agenda**. Journal of Service Management, v. 24, n. 3, p. 245-267, 2013.
- BONWELL, Charles C.; EISON, James A. **Active Learning: Creating Excitement in the Classroom**. 1991 ASHE-ERIC Higher Education Reports. ERIC Clearinghouse on Higher Education, The George Washington University, One Dupont Circle, Suite 630, Washington, DC 20036-1183, 1991.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. Elsevier Brasil, 2006.
- BORDENAVE, J.D.; PEREIRA, A. M. **Estratégias de ensino aprendizagem**. 4. ed. Petrópolis: Vozes, 1982.
- BORDENAVE, Juan E. Dias et al. **Alguns fatores pedagógicos. Capacitação em Desenvolvimento de Recursos Humanos**. CADRHU, p. 261-268, 1999.
- BOSCH, Nigel et al. **Automatic detection of learning-centered affective states in the wild**. In: Proceedings of the 20th international conference on intelligent user interfaces. ACM, 2015.B p. 379-388.
- BOSCH, Nigel; D'MELLO, Sidney. **The Affective Experience of Novice Computer Programmers**. International Journal of Artificial Intelligence in Education, v. 27, n. 1, p. 181-206, 2017.
- BOSSE, Yorah; GEROSA, Marco Aurélio. **Why is programming so difficult to learn?: Patterns of Difficulties Related to Programming Learning Mid-Stage**. ACM SIGSOFT Software Engineering Notes, v. 41, n. 6, p. 1-6, 2017.
- BRACKMANN, Christian Puhlmann et al. **Development of Computational Thinking in Brazilian Schools with Social and Economic Vulnerability: How to Teach Computer Science Without Machines**. International Journal of Innovation Education and Research, v. 7, n. 4, p. 79-96, 2019.

BRASSCOM. **O Mercado de Profissionais de Tecnologia da Informação e Comunicação no Brasil: uma análise do período de 2006 a 2013.** Disponível em www.brasscom.org.br/brasscom/Portugues/download.php?cod=1775. Acessado em 30 de julho de 2017.

CACEFFO, Ricardo; GAMA, Guilherme; AZEVEDO, Rodolfo. **Exploring active learning approaches to computer science classes.** In: Proceedings of the 49th ACM Technical Symposium on Computer Science Education. ACM, 2018. p. 922-927.

CANTO, Alberto B.do C.Filho. **Motrac–Modelo de trajetórias de aprendizagem conceitual.** Tese de doutorado. Universidade Federal do Rio Grande do Sul, 2015.

CHAKRABARTY, S. **Mining the Web: Analysis of Hypertext and Semi Structured Data.** Morgan Kauffman, 2002.

CHANG, Chia-Jung et al. **An analysis of student collaborative problem solving activities mediated by collaborative simulations.** Computers & Education, v. 114, p. 222-235, 2017.

CHATTI, Mohamed Amine et al. **A reference model for learning analytics.** International Journal of Technology Enhanced Learning, v. 4, n. 5-6, p. 318-331, 2012.

CRESWELL, John W. **Projeto de pesquisa: métodos qualitativo, quantitativo e misto.** 3. ed. Porto Alegre: Artmed, 2010.

DA CUNHA, Larissa Sara; TONETTI, Pedro; SANAVRIA, Claudio Zarate. **O Ensino de Informática no Brasil: Uma Análise da Produção Científica em Eventos da SBC (2010–2014).** Anais do Computer on the Beach, p. 031-040, 2017.

DA SILVA RIBEIRO et al. **Uma visão do cenário Nacional do Ensino de Algoritmos e Programação: uma proposta baseada no Paradigma de Programação Visual.** In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2012.

DA SILVA, Leandro Augusto; PERES, Sarajane Marques; BOSCARIOLI, Clodis. **Introdução à mineração de dados: com aplicações em R.** Elsevier Brasil, 2017.

DA SILVA, Luciana Guedes et al. **Dinâmicas de evasão na educação superior brasileira.** Examen: Política, Gestão E Avaliação Da Educação, v. 2, n. 2, p. 100-127, 2018.

DAMÁSIO, António R. **O erro de Descartes: Emoção, razão e o cérebro humano.** Editora Companhia das Letras, 2009.

DAMÁSIO, António R. **Em busca de Espinosa: Prazer e dor na ciência dos sentimentos.** Adaptação para o português do Brasil: Laura Teixeira Motta. São Paulo. Companhia das Letras. 2004.

DE BARROS COSTA, Evandro; ROCHA, Hemilis Joyse Barbosa. **Programação numa Abordagem de Aprendizagem baseada em Resolução de Problemas e Jogos: Um Mapeamento Sistemático.** SBC – Proceedings of SBGames. 2018.

DE OLIVEIRA BALDUINO, Jefferson; FERREIRA, Fani Santos Simão. **Proposta de uma nova abordagem para desenvolvimento de algoritmos de programação.** Linkscienceplace-Interdisciplinary Scientific Journal, v. 2, n. 1, 2015.

DE VICENTE, A.; PAIN, H. **Motivation Diagnosis in Intelligent Tutoring Systems.** p.86–95, Springer Berlin Heidelberg. 1998.

DEL SOLATO, Teresa; DU BOULAY, Benedict. **Implementation of motivational tactics in tutoring systems.** Journal of Interactive Learning Research, v. 6, n. 4, p. 337, 1995.

DEWEY, John. **Como pensamos: como se relaciona o pensamento reflexivo com o processo educativo: uma reexposição.** 3. ed. São Paulo: Nacional, 1959.

DRINI, Merlinda. **Using new methodologies in teaching computer programming.** In: Integrated STEM Education Conference (ISEC), 2018 IEEE. IEEE, 2018. p. 120-124.

DU BOULAY, Benedict; DEL SOLDATO, Teresa. **Implementation of Motivational Tactics in Tutoring Systems: 20 years on.** International Journal of Artificial Intelligence in Education, v. 26, n. 1, p. 170-182, 2016.

DUMDUMAYA, Cristina E.; BANAWAN, Michelle P.; RODRIGO, Ma Mercedes T. **Identifying Students' Persistence Profiles in Problem Solving Task.** In: Adjunct Publication of the 26th Conference on User Modeling, Adaptation and Personalization. ACM, 2018. p. 281-286.

FARREL, Joyce. **Lógica e design de programação: introdução.** Cengage Learning, 2010.

FAYYAD, Usama; PIATETSKY-SHAPIRO, Gregory; SMYTH, Padhraic. **From data mining to knowledge discovery in databases.** AI magazine, v. 17, n. 3, p. 37, 1996.

FIGUEIREDO, José; GARCÍA-PEÑALVO, Francisco José. **Building Skills in Introductory Programming.** In: Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality. ACM, 2018. p. 46-50.

FLICK, Uwe. **Introdução à pesquisa qualitativa.** 3. ed. Porto Alegre: Artmed, 2009.

FLORES, C. D.; BARONE, Dante et al. **Fundamentos dos Sistemas Especialistas. In:Sociedades artificiais: a nova fronteira da inteligência nas máquinas.** Porto Alegre: Bookman, p. 127-154, 2003.

FREEMAN, Scott et al. **Active learning increases student performance in science, engineering, and mathematics.** Proceedings of the National Academy of Sciences, v. 111, n. 23, p. 8410-8415, 2014.

FUHR, Bruno Edgar. **Desenvolvimento de uma ferramenta de coleta e armazenamento de dados para Big Data.** 2015. Trabalho de Conclusão de Curso.

GIL, A. C. **Como elaborar projetos de pesquisa.** 2010. 5ª. edição. São Paulo, SP. Atlas.

GIRAFFA, Maria Martins; DA COSTA MORA, Michael. **Evasão na disciplina de algoritmo e programação: um estudo a partir dos fatores intervenientes na perspectiva do aluno.** In: Congressos CLABES. 2013.

- GOMES, Anabela; MENDES, António José. **Learning to program-difficulties and solutions.** In: International Conference on Engineering Education–ICEE. 2007.
- GOMES, Anabela; HENRIQUES, Joana; MENDES, António. **Uma proposta para ajudar alunos com dificuldades na aprendizagem inicial de programação de computadores.** Educação, Formação & Tecnologias-ISSN 1646-933X, v. 1, n. 1, p. [93-103], 2008.
- GOMINHO, Fausto José Feitosa Barbosa. **Um assistente de recomendação sensível ao contexto para ambientes virtuais de aprendizagem baseados na metodologia da problematização.** 2014.
- JAQUES, Patrícia A.; NUNES, Maria Augusta SN. **Ambientes Inteligentes de Aprendizagem que inferem, expressam e possuem emoções e personalidade.** Jornada de Atualização em Informática na Educação, v. 1, n. 1, p. 30-81, 2013.
- HAN, Jiawei; PEI, Jian; KAMBER, Micheline. **Data mining: concepts and techniques.** Elsevier, 2011.
- HAZZAN, Orit; KRAMER, Jeff. **Assessing abstraction skills.** Communications of the ACM, v. 59, n. 12, p. 43-45, 2016.
- HEUSER, Carlos Alberto. **Projeto de banco de dados.** Volume 4 da Série Livros didáticos informática UFRGS. Bookman Editora, 2009.
- HOED, Raphael Magalhães. **Análise da evasão em cursos superiores: o caso da evasão em cursos superiores da área de Computação.** Brasília, DF: Universidade de Brasília, 2016.
- HOSSEINI, Roya; VIHAVAINEN, Arto; BRUSILOVSKY, Peter. **Exploring problem solving paths in a Java programming course.** 2014.
- IEPSEN, Edécio Fernando; BERCHT, Magda; REATEGUI, Eliseo Berni. **Avaliando a Dimensão Afetiva para Apoio ao Processo de Aprendizagem na Disciplina de Algoritmos: um Estudo de Caso.** RELATEC: Revista Latinoamericana de Tecnología Educativa, v. 12, n. 2, p. 55-66, 2013.
- IEPSEN, Edécio Fernando. **Ensino de algoritmos: detecção do estado afetivo de frustração para apoio ao processo de aprendizagem.** Tese de doutorado. PPGIE/UFRGS. 2013.
- JARRELL, Amanda; HARLEY, Jason M.; LAJOIE, Susanne P. **The link between achievement emotions, appraisals, and task performance: pedagogical considerations for emotions in CBLEs.** Journal of Computers in Education, v. 3, n. 3, p. 289-307, 2016.
- JENKINS, Tony. **On the difficulty of learning to program.** In: Proceedings of the 3rd Annual Conference of the LTSN Centre for Information and Computer Sciences. 2002. p. 53-58.
- JÚNIOR, J. C. R. P.; RAPKIEWICZ, Clevi Elena. **O processo de ensino-aprendizagem de fundamentos de Programação: uma visão crítica da pesquisa no Brasil.** In: Anais do XII Workshop sobre Educação em Computação (SBC). 2004.
- JUNIOR, Ademar Crotti; WILGES, Beatriz; NASSAR, Silvia Modesto. **Modelagem bayesiana da aprendizagem de estudantes em um AVA.** In: Brazilian Symposium on Computers in Education (SBIE). 2014. p. 772.

KAMPFF, Adriana Justin Cerveira; REATEGUI, Eliseo Berni; DE LIMA, José Valdeni. **Mineração de dados educacionais para a construção de alertas em ambientes virtuais de aprendizagem como apoio à prática docente.** Novas Tecnologias na Educação, v. 6, n. 2, 2008.

KAMPFF, Adriana Justin Cerveira. **Mineração de dados educacionais para geração de alertas em ambientes virtuais de aprendizagem como apoio à prática docente.** Tese de doutorado. 2009.

KELLER, John M. **Development and use of the ARCS model of instructional design.** *Journal of instructional development.* v. 10, n. 3, p. 2-10, 1987.

KNUTH, Donald Ervin. **The art of computer programming: sorting and searching.** Pearson Education, 1998.

KRAMER, Jeff. **Is abstraction the key to computing?** *Communications of the ACM,* v. 50, n. 4, p. 36-42, 2007.

KRAUSE, Kerri-Lee; COATES, Hamish. **Students' engagement in first-year university.** *Assessment & Evaluation in Higher Education,* v. 33, n. 5, p. 493-505, 2008.

KUNKLE, Wanda M.; ALLEN, Robert B. **The impact of different teaching approaches and languages on student learning of introductory programming concepts.** *ACM Transactions on Computing Education (TOCE),* v. 16, n. 1, p. 3, 2016.

LAHTINEN, Essi; ALA-MUTKA, Kirsti; JÄRVINEN, Hannu-Matti. **A study of the difficulties of novice programmers.** In: *Acm Sigse Bulletin.* ACM, 2005. p. 14-18.

LARSON, Ron; FARBER, Betsy; PATARRA, Cyro. **Estatística aplicada.** 2. ed. São Paulo: Prentice Hall, 2004.

LONGHI, Magalí T.; BEHAR, Patricia A.; BERCHT, Magda. **Máquina de inferência baseada na teoria bayesiana para identificar os estados de ânimo do aluno em um ambiente virtual de aprendizagem.** *RENOTE,* v. 8, n. 3, 2010.

LONGHI, Magalí Teresinha. **Mapeamento de aspectos afetivos em um ambiente virtual de aprendizagem.** Tese de doutorado. PPGIE/UFRGS. 2011.

LOWENTHAL, Patrick; WILSON, Brent G. Labels do matter! **A critique of AECT's redefinition of the field.** *TechTrends,* v. 54, n. 1, p. 38-46, 2010.

MACHADO, Vinícius; PINHEIRO, Nilcéia Aparecida Maciel. **Contribuições para a formação acadêmica do engenheiro: trabalhando por meio de Problemas Geradores de Discussões.** *Anais do I Simpósio Nacional de Ensino de Ciência e Tecnologia,* UTFPR, Ponta Grossa, p. 1488-1498, 2009.

MARCELINO, Maria José et al. **Learning Computational Thinking and scratch at distance.** *Computers in Human Behavior,* v. 80, p. 470-477, 2018.

MARCELINO, Jean Douglas; KEMCZINSKI, Avanilde. **Metodologias e Estratégias Didático-Pedagógicas para o Ensino de Lógica de Programação e Algoritmos: Um Mapeamento Sistemático da Literatura.** *Anais do Computer on the Beach,* p. 287-296, 2018.

MAYER, Richard E. **The psychology of how novices learn computer programming.** ACM Computing Surveys (CSUR), v. 13, n. 1, p. 121-141, 1981.

MEC: MINISTÉRIO DA EDUCAÇÃO. **Diretrizes Curriculares Nacionais para os cursos de graduação na área da Computação.** Brasília. 2016.

MEIRELES, Maria Costa; BONIFÁCIO, Bruno. **Uso de métodos ágeis e aprendizagem baseada em problema no ensino de engenharia de software: Um relato de experiência.** In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2015. p. 180.

MICAEL SOUZA, Draylson; DA SILVA BATISTA, Marisa Helena; BARBOSA, Ellen Francine. **Problemas e Dificuldades no Ensino e na Aprendizagem de Programação: Um Mapeamento Sistemático.** Revista Brasileira de Informática na Educação, v. 24, n. 1, 2016.

MOISSA, Barbara; GASPARINI, Isabela; KEMCZINSKI, Avaniilde. **Educational Data Mining versus Learning Analytics: estamos reinventando a roda? Um mapeamento sistemático.** In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2015. p. 1167.

MORAIS, Felipe et al. **Computação Afetiva aplicada à Educação: uma revisão sistemática das pesquisas publicadas no Brasil.** In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2017. p. 163.

MORAN, José. **Metodologias ativas para uma aprendizagem mais profunda.** Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática. Porto Alegre: Penso, p. 02-25, 2018.

NAGANO, Leandro. **Ensino de Lógica de Programação baseado na indução-dedução através de exemplos.** In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2016. p. 1352.

NIKULA, Uolevi; GOTEL, Orlena; KASURINEN, Jussi. **A motivation guided holistic rehabilitation of the first programming course.** ACM Transactions on Computing Education (TOCE), v. 11, n. 4, p. 24, 2011.

NOWACZYK, Ronald H. **The relationship of problem-solving ability and course performance among novice programmers.** International Journal of Man-Machine Studies, v. 21, n. 2, p. 149-160, 1984.

PACICO, Juliana Cerentini; BASTIANELLO, Micheline Roat. **As origens da psicologia positiva e os primeiros estudos brasileiros.** Hutz, CS Avaliação em Psicologia Positiva. Porto Alegre: Artes Médicas, 2014.

PALMEIRA, Luísa Behrens; SANTOS, Matheus Parreiras. **Evasão no Bacharelado em Ciência da Computação da Universidade de Brasília: análise e mineração de dados.** Monografia do curso de Ciência da Computação da UnB–Universidade de Brasília. Brasília: UnB, 2014.

PAPADAKIS, Stamatios et al. **The appropriateness of scratch and app inventor as educational environments for teaching introductory programming in primary and**

secondary education. In: *Early Childhood Development: Concepts, Methodologies, Tools, and Applications*. IGI Global, 2019. p. 797-819.

PARELLADA, I. L., RUFINI, S. É. **O uso do computador como estratégia educacional: relações com a motivação e aprendizado de alunos do ensino fundamental.** *Psicologia: Reflexão e Crítica*, v. 26, n. 4, p. 743–751, 2013.

PEARS, Arnold et al. **A survey of literature on the teaching of introductory programming.** *ACM SIGCSE Bulletin*, v. 39, n. 4, p. 204-223, 2007.

PEREIRA, Clarisse Ferrão et al. **Aprendizagem Baseada em Problemas (ABP)–Uma proposta inovadora para os cursos de engenharia.** *Simpósio de Engenharia de Produção–XIV SIMPEP 2007*, 2007.

PERRENOUD, Philippe. **Construir as competências desde a escola.** Porto Alegre: Artmed, 1999.

PICARD, Rosalind W.; PICARD, Rosalind. **Affective computing.** Cambridge: MIT press, 1997.

PICARD, Rosalind W. et al. **Affective learning—a manifesto.** *BT technology journal*, v. 22, n. 4, p. 253-269, 2004.

PINTRICH, Paul R. et al. **A manual for the use of the Motivated Strategies for Learning Questionnaire (MSLQ).** 1991.

PRADO DA SILVA JR, Carlos Alberto; FONTENELE, Heliana Barbosa; RODRIGUES DA SILVA, Antônio Néelson. **Transportation engineering education for undergraduate students: Competencies, skills, teaching-learning, and evaluation.** *Journal of Professional Issues in Engineering Education and Practice*, v. 141, n. 3, p. 05014006, 2014.

PRINCE, Michael. **Does active learning work? A review of the research.** *Journal of engineering education*. v. 93, n. 3, p. 223-231, 2004.

PRINCE, Michael J.; FELDER, Richard M. **Inductive teaching and learning methods: Definitions, comparisons, and research bases.** *Journal of engineering education*, v. 95, n. 2, p. 123-138, 2006.

QIU, Ming; CHEN, Lin. **Notice of Retraction A Problem-Based Learning Approach to Teaching an Advanced Software Engineering Course.** In: *Education Technology and Computer Science (ETCS), 2010 Second International Workshop on*. IEEE, 2010. p. 252-255.

QIAN, Yizhou; LEHMAN, James. **Students’ Misconceptions and Other Difficulties in Introductory Programming: A Literature Review.** *ACM Transactions on Computing Education (TOCE)*, v. 18, n. 1, p. 1, 2017.

QUARTO, Cícero Costa. **Em direção à formação otimizada de grupos para a aprendizagem colaborativa.** Tese de doutorado. UFRGS. 2016.

RAMOS, Vinicius et al. **A Comparação da Realidade Mundial do Ensino de Programação para Iniciantes com a Realidade Nacional: Revisão sistemática da literatura em eventos**

brasileiros. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2015. p. 318.

REEVE, Johnmarshall. **Understanding motivation and emotion.** John Wiley & Sons, 2014.

RICHEY, Rita C.; SILBER, K. H.; ELY, D. P. Reflections on the 2008 **AECT Definitions of the Field.** TechTrends, v. 52, n. 1, p. 24-25, 2008.

ROBB, C. **The impact of motivational messages on student performance in community college online courses.** PhD Thesis. University of Illinois, 2010.

ROBINS, Anthony; ROUNTREE, Janet; ROUNTREE, Nathan. **Learning and teaching programming: A review and discussion.** Computer science education, v. 13, n. 2, p. 137-172, 2003.

RODRIGUES, Francisco Scheffel. **Estudo sobre a evasão no curso de Ciência da computação da UFRGS.** 2013.

ROMERO, Cristóbal; VENTURA, Sebastián. **Educational data mining: a review of the state of the art.** IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), v. 40, n. 6, p. 601-618, 2010.

ROMERO, Cristobal; VENTURA, Sebastian. **Data mining in education.** Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, v. 3, n. 1, p. 12-27, 2013.

RUSSELL, Stuart; NORVIG, Peter. **Inteligência artificial.** Rio de Janeiro: Elsevier, 2013.

RUSSELL, Ingrid; MARKOV, Zdravko. **An introduction to the Weka data mining system.** In: Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. ACM, 2017. p. 742-742.

RYAN, Richard M.; DECI, Edward L. **Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being.** American psychologist, v. 55, n. 1, p. 68, 2000.

SACHARIN, V.; SCHLEGEL, K.; SCHERER, K. R. **Geneva Emotion Wheel rating study (Report).** Geneva, Switzerland: University of Geneva. Swiss Center for Affective Sciences, 2012.

SANTOS, Fabricia Damando. **Descoberta do desânimo de alunos em ambientes virtuais de ensino e aprendizagem: um modelo a partir da mineração de dados educacionais.** Tese de doutorado. PPGIE/UFRGS. 2016.

SASSAKI, Alex Hayato et al. **Por que o Brasil vai Mal no PISA? Uma Análise dos Determinantes do Desempenho no Exame.** 2018. Disponível em <https://www.insper.edu.br/wp-content/uploads/2018/08/Por-que-Brasil-vai-mal-PISA-Analise-Determinantes-Desempenho.pdf>. Acessado em 13/03/19.

SBC. **Educação Superior em Computação – Estatísticas 2017.** Disponível em: <http://www.sbc.org.br/documentos-da-sbc/summary/133-estatisticas/1200-pdf-png-educacao-superior-em-computacao-estatisticas-2017>. Acessado em 20 de dezembro de 2018.

SCAICO, Pasqueline Dantas et al. **Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem scratch.** Revista Brasileira de Informática na Educação, v. 21, n. 02, p. 92, 2013.

SERRES, M. **Filosofia mestiça.** Rio de Janeiro. Nova Fronteira. 1993.

SHEARD, Judy et al. **Analysis of research into the teaching and learning of programming.** In: Proceedings of the fifth international workshop on Computing education research workshop. ACM, 2009. p. 93-104.

SCHERER, Klaus R. **Psychological models of emotion.** The neuropsychology of emotion, v. 137, n. 3, p. 137-162, 2000.

SCHERER, Klaus R. **What are emotions? And how can they be measured?** Social science information, v. 44, n. 4, p. 695-729, 2005.

SCHERER, Klaus R. et al. **The GRID meets the Wheel: Assessing emotional feeling via self-report.** Components of emotional meaning: A sourcebook, p. 281-298, 2013.

SIEMENS, George; D BAKER, Ryan SJ. **Learning analytics and educational data mining: towards communication and collaboration.** In: Proceedings of the 2nd international conference on learning analytics and knowledge. ACM, 2012. p. 252-254.

SILVA, João Carlos Sedraz et al. **Mineração de Dados Educacionais Orientada por Atividades de Aprendizagem.** RENOTE, v. 14, n. 1, 2016.

SILVA, M. L.; CRUZ, V. A.; SILVA, F. F. **A Dimensão Afetiva e sua Relevância no Processo de Ensino Aprendizagem: Uma abordagem Sociocognitiva.** REGET - Revista Eletrônica em Gestão, Educação e Tecnologia Ambiental. V. 18, N. 3, 2014.

SOLLOWAY, Elliot; EHRLICH, Kate. **Empirical studies of programming knowledge.** IEEE Transactions on software engineering, n. 5, p. 595-609, 1984.

SOUTO, Mychelline; TEDESCO, Patrícia. **Uma Revisão sistemática da Literatura sobre conhecimentos, habilidades, atitudes e competências desejáveis para auxiliar a aprendizagem de programação.** In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2017. p. 1162.

SPIEGEL, Murray Ralph; STEPHENS, Larry J. **Estatística.** 4. ed. Porto Alegre: Bookman, 2009.

STADELHOFER, Luiza Engler; GASPARINI, Isabela. **Ensino de Algoritmos e Lógica de Programação para os Diferentes Cursos: Um Mapeamento Sistemático da Literatura.** In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2018. p. 108.

TEUSNER, Ralf; HILLE, Thomas. **On the Impact of Programming Exercise Descriptions.** In: 2018 Learning With MOOCS (LWMOOCS). IEEE, 2018. p. 51-54.

TAN, Pang-Ning; STEINBACH, Michael; KUMAR, Vipin. **Introdução ao Datamining: mineração de dados.** Rio de Janeiro: Ciência Moderna, 2009.

TRAMONTINA, PE. **Gerador automático de exercícios para apoio ao ensino de programação.** Trabalho de conclusão de curso. UNIVATES. 2016.

TRAN, Véronique. **The influence of emotions on decision-making processes in management teams=(l'influence des émotions sur les processus de prise de décision dans les équipes de cadres).** 2004. Tese de Doutorado. University of Geneva.

TWENGE, Jean; CAMPBELL, Stacy. **Generational differences in psychological traits and their impact on the workplace.** IEEE Engineering Management Review, v. 39, n. 2, p. 72-84, 2011.

UNTERLEIDER, Carlos Eduardo Appollo et al. **Percepção de alunos de engenharia de produção acerca da utilização da metodologia da problematização no ensino de QFD.** Revista Liberato, v. 11, n. 16, 2010.

ULLOA, Miguel. **Teaching and learning computer programming: a survey of student problems, teaching methods, and automated instructional tools.** ACM SIGCSE Bulletin, v. 12, n. 2, p. 48-64, 1980.

UTTING, Ian et al. **A fresh look at novice programmers' performance and their teachers' expectations.** In: Proceedings of the ITiCSE working group reports conference on Innovation and technology in computer science education-working group reports. ACM, 2013. p. 15-32.

VEERASAMY, Ashok Kumar et al. **Relationship between perceived problem-solving skills and academic performance of novice learners in introductory programming courses.** Journal of Computer Assisted Learning, v. 35, n. 2, p. 246-255, 2019.

VIER, Juliano; GLUZ, João; JAQUES, Patricia A. **Empregando Redes Bayesianas para modelar automaticamente o conhecimento dos aprendizes em Lógica de Programação.** Revista Brasileira de Informática na Educação, v. 23, n. 2, 2015.

VIHAVAINEN, Arto; AIRAKSINEN, Jonne; WATSON, Christopher. **A systematic review of approaches for teaching introductory programming and their influence on success.** In: Proceedings of the tenth annual conference on International computing education research. ACM, 2014. p. 19-26.

WAIKATO. **Weka Knowledge Explorer.** The University of WAIKATO Disponível em: <http://www.cs.waikato.ac.nz/~ml/weka/gui_explorer.html>. Acesso em: 01 de Julho de 2017.

WATSON, Christopher; LI, Frederick WB. **Failure rates in introductory programming revisited.** In: Proceedings of the 2014 conference on Innovation & technology in computer science education. ACM, 2014. p. 39-44.

WITT, Diego Teixeira; KEMCZINSKI, Avanilde; DOS SANTOS, Luciane Mulazani. **Resolução de problemas: Abordagens aplicadas no ensino de computação.** Anais do Computer on the Beach, p. 731-740, 2018

WITTEN, Ian H., FRANK, Eibe. DM: **Practical Machine Learning Tools and Techniques, Second Edition.** San Francisco: Morgan Kaufmann Publishers, 2005.

XIANG, Hua et al. **A mind model for an affective computer.** International Journal of Computer Science and Network Security, v. 6, n. 6, p. 62-69, 2006.

YIN, Robert K. **Estudo de caso: planejamento e métodos**. 4. ed. Porto Alegre: Bookman, 2010.

ZANINI, Adriana Salvador; RAABE, André Luís Alice. **Análise dos enunciados utilizados nos problemas de programação introdutória em cursos de Ciência da Computação no Brasil**. In: Anais do XXXII Congresso da Sociedade Brasileira de Computação, XX WEI-Workshop sobre Educação em Computação. 2012.

ZORZO, A. F.; NUNES, D.; MATOS, E.; STEINMACHER, I.; LEITE, J.; ARAUJO, R. M.; CORREIA, R.; MARTINS, S. **Referenciais de Formação para os Cursos de Graduação em Computação**. Sociedade Brasileira de Computação (SBC). 153p, 2017. ISBN 978-85-7669-424-3.

8.1 Publicações relacionadas à tese

BERCHT, Magda; VOSS, Gleizer; FRANZEN, Evandro. **Explorando a motivação na utilização de mundos virtuais**. In: Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE). 2016. p. 1116.

FRANZEN, Evandro; BERCHT, Magda; DERTZBACHER, Juliano. **Problematização aplicada ao ensino e aprendizagem de algoritmos: Uma análise dos fatores associados a motivação dos estudantes**. RENOTE, v. 15, n. 1. 2017.

FRANZEN, Evandro; HEMMING, Cleverton; BERCHT, Magda. **Sistema de apoio ao uso da problematização no ensino e aprendizagem de programação**. Revista Destaques Acadêmicos, v. 10, n. 4, 2018.

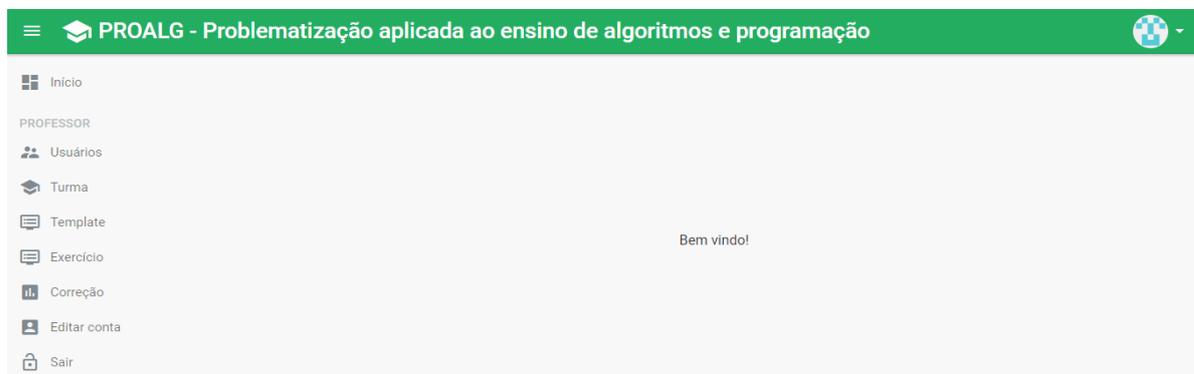
HEMING, Cléverton. **Ferramenta de apoio ao ensino e aprendizagem de algoritmos e programação**. 2018. Monografia (Graduação em Sistemas de Informação) – Universidade do Vale do Taquari - Univates, Lajeado, 05 jul. 2018. Disponível em: <<http://hdl.handle.net/10737/2228>>.

SAUSEN, Frederico Jacobi; FRANZEN, Evandro. **Projeto e desenvolvimento de um sistema para definição de aspectos e análise de sentimentos em textos**. Revista Destaques Acadêmicos, v. 7, n. 4, 2015.

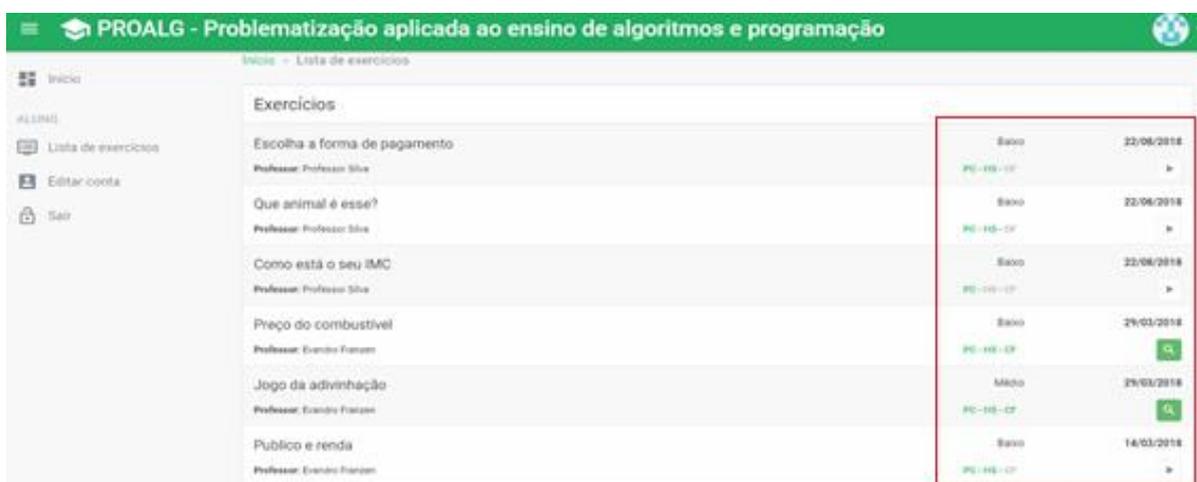
WILDNER, Maria Claudete S., FRANZEN, Evandro, GOMES, Eduardo Rodrigues. **HELPBLOCK: uma ferramenta web baseada na biblioteca Blockly para apoio ao ensino de algoritmos**. Revista Tecnologias na Educação. v.25. 2018.

APÊNDICE A – Interfaces (telas) do sistema PROALG

Tela inicial do sistema



Lista de atividades para o estudante



Listagem dos templates cadastrados



Manutenção de dicas

The screenshot shows the 'Lista de dicas' (List of tips) page in the PROALG system. The header is green with the text 'PROALG - Problematização aplicada ao ensino de algoritmos e programação'. The breadcrumb trail is 'Início > Lista de templates > Lista de dicas'. On the left, there is a sidebar menu with options: Início, PROFESSOR, Usuários, Turma, Template, Exercício, and Correção. The main content area has a table with two columns: 'Sequência' and 'Título'. The table contains two rows: one with '1' and 'Variáveis para armazenar o público', and another with '2' and 'Percentual de ocupação'. Below the table, there are buttons for 'Filtro (F2)', 'Voltar', and 'Nova dica'. Below the table, there is a form titled 'Adicionar/Alterar Dica' with fields for 'Sequência' (value: 1), 'Título' (value: Variáveis para armazenar o público), and 'Descrição' (text: O público das cadeiras e arquibancadas pode ser armazenado em duas variáveis inteiras. Os valores devem ser informados pelo usuário, como entrada para o programa.). At the bottom of the form are buttons for 'Atualizar', 'Excluir', and 'Cancelar'.

Publicação de novo exercício para uma determinada turma

The screenshot shows the 'Adicionar/Alterar Exercício' (Add/Edit Exercise) page in the PROALG system. The header is green with the text 'PROALG - Problematização aplicada ao ensino de algoritmos e programação'. The breadcrumb trail is 'Início > Lista de exercícios > Adicionar/Alterar Exercício'. On the left, there is a sidebar menu with options: Início, PROFESSOR, Usuários, Turma, Template, Exercício, Correção, Editar conta, and Sair. The main content area has a form titled 'Exercício' with the following fields: 'Template' (dropdown: Publico e renda), 'Turma' (dropdown: 2018B - T1 - Algoritmos e programação), 'Módulo' (dropdown: Operadores lógicos), 'Usa problematização' (dropdown: Sim), 'Pontos problematização' (input: 10), 'P. resolução' (input: 10), and 'Publicação' (calendar icon: 05/09/2018). At the bottom of the form are buttons for 'Adicionar' and 'Cancelar'.

Listagem das turmas e exercícios

The screenshot shows the 'Correção - Turmas' section of the PROALG system. It displays a grid of six class cards, each with a green graduation cap icon, the course name, semester, and a 'Corrigir' button. The data for each card is as follows:

Turma	Exercícios	Alunos
Algoritmos e programação 2018B - T1	18	29
Algoritmos e programação 2017B - T1	8	27
Algoritmos e programação 2018 - TESTE	24	5
Algoritmos e programação 2017B - T2	2	25
Algoritmos e programação 2018A - T1	13	29
Programação orientada a objetos 2017A	7	28

Lista de exercícios publicados para a turma 2018B – T1.

The screenshot shows the 'Algoritmos e programação 2018B - T1' class page. It displays a table of exercises with columns for Title, Module, Published, Finalized, and Corrected. A red box highlights the search icons in the 'Corrigido' column.

Título	Módulo	Publicado	Finalizado	Corrigido
Lista de chamada	Vetores e matrizes	22/10/2018	12 de 28	0 de 28
Controle de armazenamento de produtos	Vetores e matrizes	22/10/2018	10 de 28	0 de 28
Estatísticas de cursos superiores	Vetores e matrizes	22/10/2018	13 de 28	0 de 28
Avaliando os tempos das tarefas	Vetores e matrizes	22/10/2018	14 de 28	0 de 28
Capacidade das salas usadas em treinamentos	Vetores e matrizes	08/10/2018	20 de 28	0 de 28
Associados do clube da terceira idade	Vetores e matrizes	08/10/2018	18 de 28	0 de 28
Jogando dados: Qual é a soma?	Vetores e matrizes	08/10/2018	17 de 28	0 de 28

Eventos do editor de código

Evento	Data e hora	Valor
COPVAR	09/08/2018 20:30:47	<pre>public class publicoerenda { public static void main (String args[]){ int vlc; int vla; int ntl; int psc; int psa; int rnd; double pct; int cem; vlc = Integer.parseInt (args[0]); vla = Integer.parseInt (args[1]); ntl = Integer.parseInt (args[2]); psc = Integer.parseInt (args[3]); psa = Integer.parseInt (args[4]); cem = Integer.parseInt (args[5]); rnd = (vlc*psc)+(vla*psa); pct = ((psc+psa)/ntl)*cem; } }</pre>
COLAR	09/08/2018 20:50:14	<pre>public class publicoerenda { public static void main (String args[]){ int vlc; int vla; int ntl; int psc; int psa; int rnd; double pct; int cem; vlc = Entrada.leiaInt (); vla = Entrada.leiaInt (); ntl = Entrada.leiaInt (); psc = Entrada.leiaInt (); psa = Entrada.leiaInt (); rnd = (vlc*psc)+(vla*psa); pct = ((psc+psa)/ntl)*100.0; System.out.printf ("%d + %.2f", rnd, pct); } }</pre>
COPVAR	09/08/2018 20:51:06	<pre>Integer.parseInt(args[0])</pre>

Consulta das ocorrências de ajuda

Tipo da ajuda	Data e hora	Etapa	Informação da etapa
ETAPA	09/10/2018 00:56:23	CODIGO	Visualizado a Hipótese
DICA	09/10/2018 00:56:30	CODIGO	Visualizado a dica: Utilização de vetor
DICA	09/10/2018 00:56:41	CODIGO	Visualizado a dica: Leitura dos valores
PSEUDOCÓDIGO	09/10/2018 00:56:53	CODIGO	

Compilações e execuções do código

Data e hora	Código fonte	Informações
09/08/2018 20:52:38	<pre>public class publicoerenda { public static void main (String args[]){ int vlc; int vla; int ntl; int psc; int psa; int rnd; double pct; vlc = Integer.parseInt(args[0]); vla = Integer.parseInt(args[1]); ntl = Integer.parseInt(args[2]); psc = Integer.parseInt(args[3]); psa = Integer.parseInt(args[4]); rnd = (vlc*psc)+(vla*psa); pct = ((psc+psa)/ntl)*100.0; System.out.printf ("%d + %.2f", rnd, pct); } }</pre>	<p>Status COMPILAÇÃO</p> <p>00:00:01 Tempo COMPILAÇÃO</p> <p>NAO Execução ABORTADA</p> <p>Status EXECUÇÃO</p> <p>00:00:00 Tempo EXECUÇÃO</p> <p>Argumentos 80 40 2000 1000 1000</p>
09/08/2018 20:53:05	<pre>public class publicoerenda { public static void main (String args[]){ int vlc; int vla; int ntl; int psc; int psa; int rnd; double pct; vlc = Integer.parseInt(args[0]);</pre>	<p>Status COMPILAÇÃO</p> <p>00:00:02 Tempo COMPILAÇÃO</p> <p>NAO</p>

Opção para assistir uma animação do código pelo estudante

Resultado	Eventos no editor	Ajuda	Compilação	Code player
Code player				
▶ Play ▶ Play 5x ▶ Play 10x ▶ Play 20x				
<pre>1 public class Ingressos { 2 public static void main (String args []) { 3 double ingr1; 4 double ingr2; 5 double ocu_to; 6 double perc_total; 7 double ingre_vend_ingr1; 8 double ingre_vend_ingr2; 9 double renda; 10 ingr1 = 80.00; 11 ingr2 = 40.00; 12 ocu_to = 2000.00; 13 ingre_vend_ingr1 = Double.parseDouble (args[0]); 14 ingre_vend_ingr2 = Double.parseDouble (args[1]); 15 renda= (ingre_vend_ingr1 * ingr1) + (ingre_vend_ingr2 + ingr2); 16 perc_total= (ingre_vend_ingr1 +ingre_vend_ingr2)/2000.00 * 100.00; 17 System.out.printf ("A renda total de ingressos vendidos foi de %.2f e o percentual total" +renda); 18 } 19 } 20 } 21 }</pre>				

APÊNDICE B – Questionários aplicados no estudo de caso

Avaliação das atividades desenvolvidas na disciplina de Algoritmos - 2017

Esta pesquisa tem o objetivo de avaliar as atividades desenvolvidas durante este semestre. Os exercícios desenvolvidos incluíram duas etapas de problematização. As questões deste formulário estão diretamente relacionadas a estas etapas.

* Required

Nome do aluno *

Your answer

1. As lições desenvolvidas apresentaram uma descrição inicial do problema, sem indicar como ele deveria ser resolvido. A partir desta descrição como você caracteriza a sua compreensão sobre o problema? *

- Após ler a primeira descrição, eu consegui compreender completamente o problema
- Após ler a primeira descrição, entendi o problema parcialmente, mas continuei com dúvidas a respeito.
- Após ler a primeira descrição, tive muitas dúvidas sobre o problema e como desenvolver a solução
- Conseguiu compreender o problema somente depois de ver o enunciado detalhado, que foi mostrado após responder a segunda pergunta

2. Após ler a descrição inicial do problema que foi apresentado em cada atividade, como você classificaria a confiança ou crença na sua capacidade de resolver a questão.

- Em todas as atividades sempre acreditei que conseguiria resolver o problema
- Na maioria das atividades eu acreditei em conseguiria resolver o problema
- Em algumas atividades eu acreditei que conseguiria resolver o problema, mas em outras, não.
- Em quase nenhuma ou em nenhuma das atividades eu acreditei que conseguiria resolver o problema.

3. Como você avalia a estratégia que foi usada, que propõe descrever o problema e a proposta de solução antes de escrever o código. *

- Contribuí para melhorar o desenvolvimento da solução final
- Contribuí parcialmente para melhorar o desenvolvimento da solução final
- Não contribuí para melhorar o desenvolvimento da solução final

4. Como você avalia a ordem das atividades propostas (compreensão e descrição do problema e depois a escrita do código) *

- Acredito que seria mais fácil escrever o código e depois descrever o problema
- Acredito que é importante compreender e descrever o problema antes do código
- A ordem das atividades não influi na resolução do problema

5. Como você avalia o tempo dedicado por você às duas primeiras etapas das lições (pontos chave e descrição da hipótese de solução) *

- Acredito que dediquei mais tempo do que era necessário para as tarefas
- Acredito que dediquei tempo suficiente para as tarefas
- Poderia ter dedicado um tempo um pouco maior nas primeiras etapas
- Deveria ter dedicado um tempo bem maior nas primeiras etapas

6. Considerando ainda o tempo dedicado às duas primeiras etapas. Você poderia ter dedicado um tempo maior? Qual o motivo de não ter dedicado mais tempo? *

- Não dediquei um tempo maior por entender que não era necessário, por ter dedicado o suficiente
- Não dediquei um tempo maior pela dificuldade de compreender o problema
- Não dediquei um tempo maior por não ter mais tempo disponível
- Não dediquei um tempo maior por entender que descrever o problema antes não contribui para melhorar o desenvolvimento da solução

7. Durante a resolução das atividades, você utilizou os recursos (slides, manuais, links, vídeos, entre outros) indicados ou disponibilizados no ambiente virtual, pelo professor para auxiliar na resolução dos problemas?

- Sim, os recursos disponíveis em todas as atividades.
- Acessei os recursos disponíveis na maioria das atividades.
- Acessei os recursos disponíveis em algumas atividades e em outras, não.
- Não acessei os recursos disponíveis em nenhuma ou quase nenhuma atividade.

8. Durante a resolução das atividades, você utilizou outros recursos adicionais, além dos que foram indicados ou disponibilizados pelo professor, no ambiente virtual, para auxiliar na resolução dos problemas? Neste caso considere qualquer recurso (imagem, vídeo, apostilas, etc) buscados ou pesquisados durante a resolução do problema.

- Em todas as atividades busquei material adicional, ou realizei pesquisas em outros locais, além do ambiente virtual da disciplina
- Na maioria das atividades busquei material adicional, ou realizei pesquisas em outros locais, além do ambiente virtual da disciplina
- Apenas em algumas atividades busquei material adicional, ou realizei pesquisas em outros locais, além do ambiente virtual da disciplina
- Em nenhuma ou quase nenhuma atividade busquei material adicional, ou realizei pesquisas em outros locais, além do ambiente virtual da disciplina

9. Como você avalia a qualidade das respostas elaboradas por você para as duas primeiras etapas das lições (problema e hipótese) *

- Acredito que as respostas foram excelentes.
- Acredito que as respostas foram boas.
- Acredito que as respostas poderiam ser melhores
- Acredito que as respostas ficaram abaixo do esperado

10. Em cada uma das atividades estavam disponíveis dicas e o pseudocódigo para auxiliar o desenvolvimento da solução final. Sobre o acesso a estas dicas: *

- Não acessei as dicas porque não foi preciso
- Não acessei, mas gostaria de ter acessado
- Acessei as dicas poucas vezes, mas quando acessei elas me auxiliaram
- Acessei as dicas diversas vezes e elas me auxiliaram bastante

11. Qual o principal motivo que levou você a não acessar as dicas e o pseudocódigo durante a realização da atividade? *

- Não acessei porque entendia que conseguiria resolver sem as dicas
- Não acessei porque achei a minha avaliação seria prejudicada caso olhasse o conteúdo das dicas
- Após acessar em uma das questões percebi que o conteúdo não auxiliava muito, por isso não acessei mais.
- Acessei sempre que achei necessário pelas dificuldades que tive de desenvolver a solução sem conteúdo adicional

12. Como você costuma resolver os problemas na disciplina de algoritmos? *

- Faço uma leitura rápida do enunciado e inicio em seguida a escrita do código
- Dedico um pouco de tempo para compreender o problema e a seguir inicio escrita do código.
- Dedico um bom tempo para compreender o problema e a seguir inicio a escrita do código
- Elaboro uma sequencia de passos detalhada e somente após inicio a escrita do código

13. Além do ambiente virtual Moodle, durante as atividades foi utilizado um sistema chamado PROALG. Como você classificaria o seu grau de satisfação relacionado ao uso deste sistema? *

- Muito satisfeito
- Satisfeito
- Nem satisfeito, nem insatisfeito
- Insatisfeito

14. Avalie o seu grau de satisfação relacionado as atividades desenvolvidas na disciplina *

- Muito satisfeito
- Satisfeito
- Nem satisfeito, nem insatisfeito
- Insatisfeito

Durante a disciplina de Algoritmos e Programação, foi utilizada uma estratégia para a resolução de problemas que propôs a definição dos pontos chave e a descrição da hipótese de solução antes da elaboração do código fonte, como você classificaria o seu estado afetivo em relação a esta estratégia?
 Em cada questão assinale a alternativa que indica a intensidade de cada sentimento.
 Considere: 1 - Muito fraco, 2 - Fraco, 3 - Nem fraco, nem forte, 4 - Forte, 5 - Muito forte

Fiquei surpreso.

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Senti-me orgulhoso

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Por algum motivo me senti desamparado ou triste

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Fiquei entusiasmado

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Fiquei interessado

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Senti-me desvalorizado

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Por algum motivo fiquei envergonhado ou constrangido

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Fiquei feliz

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Criei expectativa

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Senti medo ou preocupação

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Fiquei tranquilo

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Senti uma certa aversão

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Senti-me satisfeito

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Fiquei irritado

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Por algum motivo senti ciúmes ou inveja

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Fiquei indiferente

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

Fiquei desapontado

	1	2	3	4	5	
Muito fraco	<input type="radio"/>	Muito forte				

APÊNDICE C – Termo de consentimento que é assinado pelo estudante

Termo de consentimento

Os professores Evandro Franzen e Juliano Dertzbacher, do Centro Universitário Univates, estão realizando pesquisas sobre o reconhecimento de estados afetivos associados ao ensino de Algoritmos e Programação. Convidamos você a participar desta pesquisa, cujo objetivo é investigar de que maneira é possível utilizar uma estratégia de ensino e aprendizagem ativa, para o aprendizado de programação, considerando a dimensão afetiva. Os participantes que concordarem em participar, poderão desistir em qualquer momento. Como benefícios aos participantes é possível citar a melhoria da aprendizagem e o uso de um método que estimule a resolução de problemas. Um possível risco seria a divulgação indevida de dados pessoais, entretanto, cabe ressaltar que não serão solicitados, nem divulgados dados pessoais. Semanalmente serão coletados dados relacionados aos exercícios desenvolvidos e no final da disciplina será aplicado um questionário (formulário online) para autorrelato. As pesquisas estão relacionadas com o trabalho do doutorando Evandro Franzen, que realiza o doutorado no PPGIE (Programa de Pós-Graduação em Informática na Educação/UFRGS), sob a orientação da Prof^a. Dr^a. Magda Bercht.

Durante o período em que ocorre a disciplina, estão sendo coletados dados relacionados as atividades realizadas pelos estudantes (tempos para resolução, conteúdos das respostas, código fonte), além de uma pesquisa para avaliar os fatores relacionados a afetividade. Os dados serão guardados pela equipe de pesquisa, por, no máximo 5 anos.

O sigilo dos nomes dos estudantes será preservado nos relatórios, com a substituição dos mesmos por números sequenciais, siglas ou nomes fictícios. Os dados não serão utilizados pelo professor da disciplina para efeito de avaliação dos alunos.

Contatos: Magda Bercht (bercht@ufrgs.br), Evandro Franzen (evandrofra@gmail.com),
 Juliano Dertzbacher (juliano.dertzbacher@univates.br), CEP-UFRGS
 (etica@propeq.ufrgs.br)

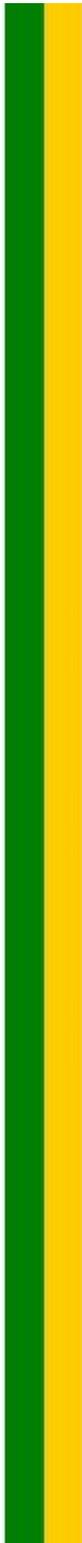
Agradecemos pela colaboração

Prof. Evandro Franzen
 Prof. Juliano Dertzabacher
 Prof^a. Dr^a. Magda Bercht.

Eu, _____,
 número de matrícula _____, autorizo a coleta de dados sobre as atividades vivenciadas na disciplina de Algoritmos e Programação, no segundo semestre de 2017.

 (assinatura)

APÊNDICE D – Certificado de registro do software PROALG






REPÚBLICA FEDERATIVA DO BRASIL
 MINISTÉRIO DA INDÚSTRIA, COMÉRCIO EXTERIOR E SERVIÇOS
INSTITUTO NACIONAL DA PROPRIEDADE INDUSTRIAL
 DIRETORIA DE PATENTES, PROGRAMAS DE COMPUTADOR E TOPOGRAFIAS DE CIRCUITOS INTEGRADOS

Certificado de Registro de Programa de Computador

Processo Nº: BR512018052265-3

O Instituto Nacional da Propriedade Industrial expede o presente certificado de registro de programa de computador, válido por 50 anos a partir de 1º de janeiro subsequente à data de 01/08/2018, em conformidade com o §2º, art. 2º da Lei 9.609, de 19 de Fevereiro de 1998.

Título: PROALG: Problematização aplicada ao ensino de programação

Data de publicação: 01/08/2018

Data de criação: 01/08/2018

Titular(es): EVANDRO FRANZEN

Autor(es): CLÉVERTON HEMING; WILLIAM VALER

Linguagem: JAVA SCRIPT; PHP

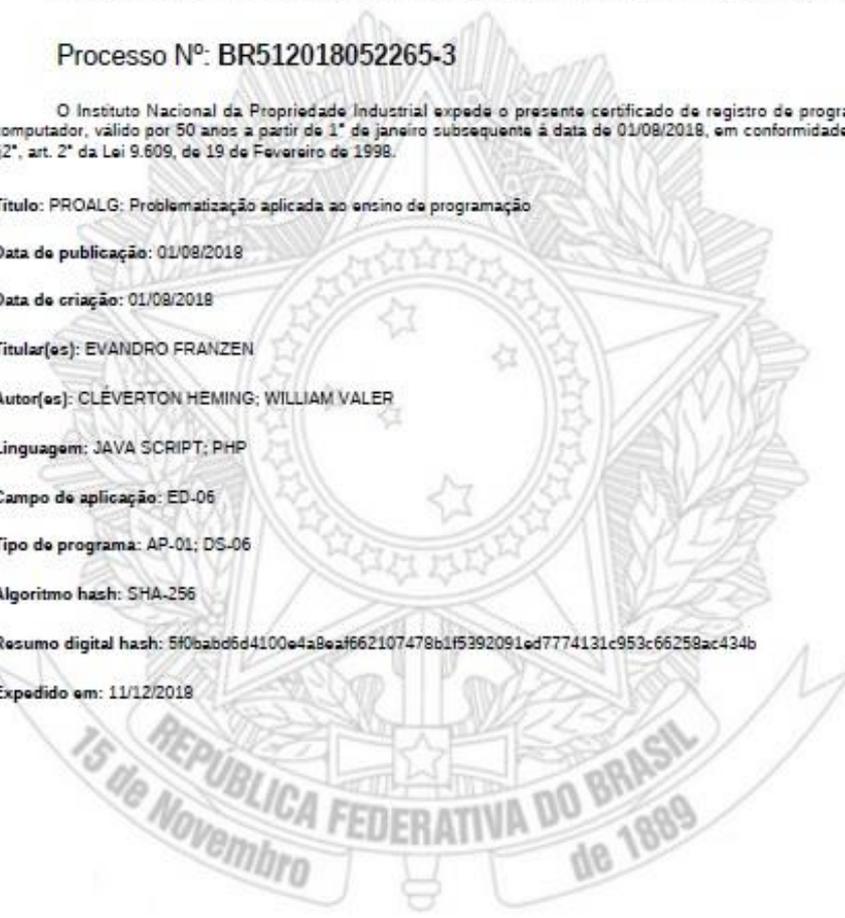
Campo de aplicação: ED-06

Tipo de programa: AP-01; DS-06

Algoritmo hash: SHA-256

Resumo digital hash: 5f0babb6d4100e4a8eaf662107478b1f5392091ed7774131c953c66258ac434b

Expedido em: 11/12/2018



15 de Novembro
REPÚBLICA FEDERATIVA DO BRASIL
de 1889

Aprovado por:
 Liane Elizabeth Caldeira Lage
 Diretora de Patentes, Programas de Computador e Topografias de Circuitos Integrados