



UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM
INSTITUTO DE COMPUTAÇÃO - ICOMP
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI

Uma Técnica Formal para Avaliação Automática e Geração de Feedback no Contexto da Educação Digital

por
Romário Lira Batista

31 de março de 2020



PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO AMAZONAS - UFAM
INSTITUTO DE COMPUTAÇÃO - ICOMP
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA - PPGI



Uma Técnica Formal para Avaliação Automática e Geração de Feedback no Contexto da Educação Digital

por
Romário Lira Batista

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Informática, da Universidade Federal do Amazonas, como parte dos requisitos necessários para obtenção do título de Mestre em Informática, na área de Informática na Educação e Engenharia de Software

Orientador: Raimundo da Silva Barreto, D.Sc.

31 de março de 2020

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

B333t Batista, Romário Lira
Uma técnica formal para avaliação automática e geração de feedback no contexto da educação digital / Romário Lira Batista . 2020
97 f.: il. color; 31 cm.

Orientador: Raimundo da Silva Barreto
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. Avaliação Automática. 2. Feedback Automático. 3. Educação Digital. 4. Métodos Formais. I. Barreto, Raimundo da Silva. II. Universidade Federal do Amazonas III. Título



PODER EXECUTIVO
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE COMPUTAÇÃO

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA



UFAM

FOLHA DE APROVAÇÃO

"Uma Técnica Formal para Avaliação Automática e Geração de Feedback no Contexto da Educação Digital"

ROMÁRIO LIRA BATISTA

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Raimundo da Silva Barreto - PRESIDENTE

Profa. Elaine Harada Teixeira de Oliveira - MEMBRO INTERNO

Dr. Javier Martinez Silva - MEMBRO EXTERNO

Manaus, 31 de Março de 2019

Esta dissertação é dedicada aos meus pais que me impulsionaram nesta jornada, concedendo a mim o apoio para persistir em busca das minhas metas, sonhos e realizações.

Agradecimentos

Agradeço primeiramente a Deus que tem me abençoado com capacidade intelectual neste processo de aprendizagem contínua, onde em cada etapa em busca pelo conhecimento, me permitiu deslumbrar novos ambientes, pelo privilégio de compartilhar tamanha experiência, contribuindo assim na consolidação do conhecimento em diferentes vertentes da minha vida.

A minha querida mãe Francisca Lira e minha irmã Franciely Lira, meus exemplos de mulher, fortes, honestas, e trabalhadoras que me incentivam a romper e superar desafios. Eu agradeço pelo apoio e paciência em cada momento, por estarem presente quando necessitei, mesmo que em alguns momentos longe pela distância, por ser um referencial de vida para mim. Muito obrigado por tudo mãe e irmã. Agradeço a minha madrinha, Rosália Alves, e minha tia Jandira Lira pelo auxílio em meus tempos difíceis e principalmente à minha prima Andreia Lira, que foi um anjo da guarda quando mais precisei.

Ao professor Raimundo Barreto, meu orientador, muito obrigado pelos ensinamentos, por todo o seu apoio e presteza no auxílio às atividades e discussões sobre o andamento deste trabalho e principalmente por acreditar no desenvolvimento do meu trabalho, tendo a paciência de um pai comigo nos últimos momentos dessa dissertação, proporcionando-me novas experiências que contribuíram significativamente no meu processo de aprendizagem e para a minha vida.

Agradeço ao Instituto de Computação, no geral, pela preocupação e compreensão que tiveram comigo no infortúnio que tive ao representar o IComp fora do Amazonas em um evento e pela troca de saberes nas discussões e esclarecimentos na execução deste trabalho.

Agradeço também aos amigos e colegas de laboratório que ajudaram com inúmeras discussões frutíferas sobre o meu trabalho, em especial aos meus amigos Márcio Alencar e Nilmara Salgado que me ajudaram sempre que precisei, me deram conselhos, orientações e muitas histórias de suas experiências que somaram à minha vida. Agradeço aos colegas de classe pela espontaneidade e alegria na troca de informações e materiais numa rara demonstração de amizade e solidariedade.

Aos professores que tanto me ajudaram e incentivaram nesta longa jornada que agora conclui mais uma etapa e aos demais idealizadores, coordenadores e funcionários da Universidade Federal do Amazonas (UFAM), em especial ao grupo do departamento do Programa de Pós-Graduação em Informática (PPGI). A FAPEAM pelo suporte financeiro no processo de execução desta pesquisa. Enfim, agradeço a todos que de alguma forma contribuíram para que eu conseguisse ultrapassar as dificuldades e alcançar meu objetivo. Meu muito obrigado!

RESUMO

As escolas modernas tentam se integrar as vantagens da tecnologia da informação em seus meios pedagógicos. No entanto, essas tentativas usualmente limitam-se a apenas acoplar junto ao ambiente físico, artefatos de *hardware* para que estudantes e professores possam utilizá-los. Quando busca-se incorporar a cultura tecnológica à educação, busca-se também incorporar novas formas de ensino e aprendizagem. Por conseguinte, não basta utilizar recursos alternativos e tecnologias de ponta, se não se desenvolver capacidades estratégicas de como e quando usar tais recursos no processo de ensino-aprendizagem. A tarefa de avaliar e classificar o desempenho de estudantes, além de fornecer informações com base nessas avaliações, é um dos pontos em que a integração com a tecnologia busca atuar. Essa automatização traz ganhos de produtividade e de gerenciamento de tempo tanto para professores quanto para alunos. A avaliação automática é uma técnica algorítmica usada para provar/refutar que a resposta de um aluno atende a critérios estabelecidos pelo professor. Quando uma resposta é classificada como “incorreta”, a avaliação dá sequência ao processo de *feedback*, que consiste no provimento de informação sobre o desempenho do estudante, objetivando reorientar ou estimular comportamentos futuros mais adequados por meio de recomendações. Nesse sentido, este trabalho apresenta uma metodologia que busca automatizar a tarefa de avaliar o desempenho de um estudante e fornecer *feedback* para alunos e professores no cenário de educação digital. Ao contrário de outros trabalhos com objetivos semelhantes, esta proposta busca utilizar um método formal, baseado em rede de Petri, como meio de garantir a consolidação e a sistematização das informações a serem avaliadas. A avaliação automática é apresentada em função da modelagem formal do comportamento do estudante ao responder uma atividade avaliativa. Já o *feedback* é baseado na verificação de propriedades e componentes desse modelo. Foi feita uma revisão sistemática da literatura onde foram identificados onze métodos utilizados a respeito do tema, sendo dez relacionados à avaliação e seis a *feedback*. Entretanto, há poucos trabalhos que usam métodos formais para a automatização da avaliação e geração de *feedback*. Foram conduzidos estudos de casos da aplicação do método proposto, onde são demonstradas situações de alunos, em duas perspectivas: a pedagógica, que relaciona as métricas com resultado final da avaliação do modelo formal do comportamento aluno; e a técnica, que mostra como os critérios que o professor seleciona podem ser atendidos a partir dos elementos e/ou propriedades do modelo para gerar *feedback* do desempenho do estudante na avaliação. Os resultados mostram que a metodologia proposta é promissora porque consegue efetivamente provar a correteza da avaliação e, adicionalmente, prover *feedback* para os alunos e professores quando estes apresentam alguma falha no desempenho.

ABSTRACT

Modern schools try to integrate the advantages of information technology in their pedagogical means. However, these attempts are usually limited to just attaching to the physical environment, hardware artifacts so that students and teachers can use them. When seeking to incorporate technological culture into education, they also seek to incorporate new forms of teaching and learning. Therefore, it is not enough to use alternative resources and state-of-the-art technologies unless strategic capacities are developed on how and when to use such resources in the teaching-learning process. The task of assessing and classifying student performance, in addition to providing information based on these assessments, is one of the points in which integration with technology seeks to act. This automation brings productivity and time management gains for both teachers and students. Automatic assessment is an algorithmic technique used to prove/reflect that a student's response meets criteria established by the teacher. When a response is classified as "incorrect", the assessment follows the feedback process, which consists of providing information about the student's performance, aiming to reorient or stimulate more appropriate future behavior through recommendations. In this sense, this work presents a methodology that seeks to automate the task of evaluating a student's performance and provide feedback to students and teachers in the digital education scenario. Unlike other works with similar objectives, this proposal seeks to use a formal method, based on Petri net, as a means of ensuring the consolidation and systematization of information to be evaluated. The automatic evaluation is presented as a function of the formal modeling of student behavior when responding to an evaluation activity. The feedback is based on the verification of properties and components of this model. A systematic review of the literature was performed where eleven methods were identified, ten of them related to the evaluation and six to the feedback. However, there are few works that use formal methods to automate the evaluation and generate feedback. Case studies of the application of the proposed method were conducted, where situations of students are demonstrated from two perspectives: the pedagogical, which relates the metrics with the final result of the evaluation of the formal model of student behavior; and the technique, which shows how the criteria that the teacher selects can be met from the elements and/or properties of the model to generate feedback on student performance in the evaluation. The results show that the proposed methodology is promising because it can effectively prove the correctness of the assessment and, additionally, provide feedback to students and teachers when they present some flaw in performance.

Sumário

Lista de Figuras	viii
Lista de Tabelas	xi
Lista de Algoritmos	xii
1 Introdução	1
1.1 Contexto	2
1.2 Definição do Problema	2
1.3 Justificativa	3
1.4 Objetivos	4
1.5 Metodologia Proposta	5
1.6 Organização do Trabalho	6
2 Conceitos e Definições	8
2.1 Tecnologia aplicada a educação: uma visão sobre a educação digital	8
2.2 A automatização do processo de avaliação	11
2.3 Da engenharia de <i>software</i> aos métodos formais	13
2.3.1 Métodos formais como suporte educacional	14
2.3.2 Métodos formais aplicado ao <i>software</i> e métodos formais aplicado a educação	15
2.4 As Redes de Petri	16
2.4.1 Visão geral	17
2.4.2 Definição formal	18
2.4.3 Propriedades matemáticas	18
2.4.4 Extensões das redes de Petri	19
2.4.5 Redes de Petri como técnica de avaliação	19
2.5 A automatização do <i>feedback</i>	20
2.6 Considerações finais	21
3 Revisão Sistemática sobre métodos, técnicas e metodologias usadas para avaliação e <i>feedback</i> automático	23
3.1 Execução da revisão sistemática	23
3.2 Análise e publicação dos resultados	26
3.2.1 Coeficiente Kappa	28
3.3 Discussão sobre os resultados	29

3.3.1	QP: Quais conceitos, métodos, metodologias, técnicas e ferramentas relacionados a avaliação e geração de <i>feedback</i> automático podem ser encontrados na literatura dentro do contexto de ambientes virtuais de aprendizagem?	29
3.3.2	Q1: Quais os métodos aplicados para produzir a avaliação automática de alunos em ambientes virtuais de aprendizagem?	30
3.3.3	Q2: Quais técnicas são capazes de gerar <i>feedbacks</i> automáticos baseados nos métodos usados como avaliação automática, satisfazendo critérios previamente definidos?	30
3.3.4	Q3: Em que cenários de ambientes de virtuais de aprendizagem esses conceitos são aplicados na literatura?	33
3.4	Tipos de estudos	34
3.5	Limitações desta revisão sistemática	35
3.6	Considerações finais	36
4	Trabalhos Correlatos	37
4.1	Métricas de Similaridade	37
4.2	Aprendizado de Máquina	38
4.3	Grafos	39
4.4	Outros Métodos	40
4.5	Resumo	40
5	Método Proposto	42
5.1	Visão geral sobre o método	42
5.2	Critérios de avaliação	43
5.2.1	Nota tradicional	43
5.2.2	Nota ponderada	44
5.2.3	Nível de desvio	44
5.2.4	Nível de dúvida	45
5.2.5	Nível de confusão	45
5.3	Etapa de pré-processamento	45
5.4	Etapa de avaliação automática	48
5.4.1	Sintetizador de rede de Petri	48
5.4.2	Executor da rede de Petri	49
5.5	Gerador de <i>feedback</i>	52
5.6	Formalização das métricas	53
5.6.1	Cenário	53
5.6.2	Nota Tradicional	61
5.6.3	Nota Ponderada	62
5.6.4	Nível de Desvio	62
5.6.5	Dúvida por Questão	63
5.6.6	Nível de Confusão	65
5.7	Resumo	66
6	Resultados	67
6.1	Publicações	67
6.2	Objetivos do estudo de caso	67
6.3	Resultados experimentais	68

6.3.1	Limitações do experimento	78
6.4	Feedback	78
6.5	Resumo	84
7	Conclusão	85
7.1	Trabalhos Futuros	86
A	Redes de Petri geradas	87
	Referências Bibliográficas	94

Lista de Figuras

1.1	Visão geral do trabalho.	5
2.1	Modelo de virtualização para adaptação tecnológica do ensino (Silva <i>et al.</i> , 2014). .	10
2.2	Modelo de virtualização que permite ao aluno construir seu próprio conhecimento (Silva <i>et al.</i> , 2014).	10
2.3	Exemplo da estrutura de uma rede de Petri.	17
2.4	Fluxo do processo de avaliação utilizando redes de Petri.	20
3.1	Processo de seleção das publicações.	27
3.2	Relação da quantidade de trabalhos retornados por período.	27
3.3	Quantificação dos critérios de aceitação em relação aos trabalhos aceitos para extração.	28
3.4	Quantificação dos critérios de rejeição em relação aos trabalhos excluídos da extração. .	28
3.5	Métodos encontrados durante a revisão em função da quantidade de trabalhos presentes.	30
3.6	Quantidade dos métodos identificados aplicados ao contexto de avaliações automáticas distribuídos por ano.	32
3.7	Quantidade dos métodos identificados aplicados ao contexto de avaliações automáticas distribuídos por ano.	32
3.8	Distribuição dos métodos por tipo de estudo.	35
4.1	Métricas utilizadas no trabalho de Thaler <i>et al.</i> (2016).	38
4.2	Metodologia proposta por Kyrilov e Noelle (2014).	39
5.1	Fluxo geral da estrutura do método.	43
5.2	Estrutura gerada pelo Interpretador.	46
5.3	Estrutura dos lugares de verificação.	47
5.4	Objetos e atributos preenchidos pelo “Interpretador”.	48
5.5	Representação das pré e pós condições para disparo de transições.	49
5.6	Exemplo de prioridades de disparo.	49
5.7	Exemplo de $m_i \in \overline{M_{SS}}$	52
5.8	Estado inicial da rede de Petri.	54
5.9	Primeiro estado da rede de Petri.	54
5.10	Estado m_2 da rede de Petri.	54
5.11	Estado m_3 da rede de Petri.	55
5.12	Estado m_4 da rede de Petri.	55
5.13	Estado m_5 da rede de Petri.	55
5.14	Estado m_6 da rede de Petri.	56
5.15	Estado m_7 da rede de Petri.	56

5.16	Estado m_8 da rede de Petri.	57
5.17	Estado m_9 da rede de Petri.	57
5.18	Estado m_{10} da rede de Petri.	57
5.19	Estado m_{11} da rede de Petri.	58
5.20	Estado m_{12} da rede de Petri.	58
5.21	Estado m_{13} da rede de Petri.	58
5.22	Estado m_{14} da rede de Petri.	59
5.23	Estado m_{15} da rede de Petri.	59
5.24	Estado m_{15} da rede de Petri com os lugares de questão que apresentam dúvidas. . .	64
5.25	Estado m_{15} da rede de Petri com os lugares de questão que apresentaram confusão. .	65
6.1	Relação entre as métricas de Nota Tradicional e Ponderada para a turma 01 do experimento no primeiro dia.	68
6.2	Relação entre as métricas de Nota Ponderada e Nível de Desvio para a turma 01 do experimento no primeiro dia.	69
6.3	Relação entre as métricas de Nota Tradicional e Ponderada para a turma 01 do experimento no segundo dia.	69
6.4	Relação entre as métricas de Nota Ponderada e Nível de Desvio para a turma 01 do experimento no segundo dia.	70
6.5	Relação entre as métricas de Nota Tradicional e Ponderada para a turma 02 do experimento no primeiro dia.	70
6.6	Relação entre as métricas de Nota Ponderada e Nível de Desvio para a turma 02 do experimento no primeiro dia.	71
6.7	Relação entre as métricas de Nota Tradicional e Ponderada para a turma 02 do experimento no segundo dia.	71
6.8	Relação entre as métricas de Nota Ponderada e Nível de Desvio para a turma 02 do experimento no segundo dia.	72
6.9	Relação entre as métricas de Nota Tradicional e Ponderada para a turma 03 do experimento no primeiro dia.	72
6.10	Relação entre as métricas de Nota Ponderada e Nível de Desvio para a turma 03 do experimento no primeiro dia.	73
6.11	Principais questões respondidas corretamente em que os estudantes tiveram dúvida no primeiro dia.	74
6.12	Nível de Dúvida dos estudantes avaliando individualmente as questões corretas no primeiro dia.	74
6.13	Principais questões respondidas incorretamente em que os estudantes tiveram dúvida no primeiro dia.	75
6.14	Nível de Dúvida dos estudantes avaliando individualmente as questões incorretas no primeiro dia.	75
6.15	Nível de Dúvida dos estudantes avaliando individualmente as questões corretas no segundo dia.	76
6.16	Nível de Dúvida dos estudantes avaliando individualmente as questões corretas no segundo dia.	76
6.17	Nível de Dúvida dos estudantes avaliando individualmente as questões incorretas no segundo dia.	77
6.18	Nível de Dúvida dos estudantes avaliando individualmente as questões incorretas no segundo dia.	77
6.19	Nível de Confusão da turmas.	78

6.20	Rede de Petri gerada para um estudante do segundo experimento.	80
6.21	Dúvida em relação a questão <i>Q3</i> da Figura 6.20.	81
6.22	Rede de Petri gerada por um estudante do segundo experimento.	82
6.23	Dúvida em relação a questão <i>Q3</i> da Figura 6.22.	83
6.24	Nível de dúvida pelo número de disparos de transições.	84
A.1	Exemplo 1 do modelo de rede de Petri gerada para um estudante.	88
A.2	Exemplo 2 do modelo de rede de Petri gerada para um estudante.	89
A.3	Exemplo 3 do modelo de rede de Petri gerada para um estudante.	90
A.4	Exemplo 4 do modelo de rede de Petri gerada para um estudante.	91
A.5	Exemplo 5 do modelo de rede de Petri gerada para um estudante.	92
A.6	Exemplo 6 do modelo de rede de Petri gerada para um estudante.	93

Lista de Tabelas

3.1	Objetivo da revisão segundo o paradigma GQM.	24
3.2	Estrutura usada para elaborar a expressão de busca.	25
3.3	Interpretação dos níveis de concordância pelo coeficiente Kappa.	29
3.4	Métodos aplicados para avaliação automática em função dos autores.	31
3.5	Métodos aplicados para geração automática de <i>feedback</i> em função dos autores. . .	33
3.6	Classificação dos trabalhos em função do cenário de aplicação dos métodos.	34
5.1	M das Figuras 5.9 à 5.23.	60
5.2	m_{15} da Figura 5.23, considerando somente os lugares de verificação.	61
6.1	Distribuição das questões por turmas e por dias.	68
6.2	Conjuntos de questões por disciplinas da primeira turma, no primeiro dia, do segundo experimento.	79

Lista de Algoritmos

1	Geração dos componentes da rede com base no <i>log</i>	46
2	Criação dos lugares de verificação e seus arcos.	47

Capítulo 1

Introdução

Em nossa sociedade, conforme explica Leitão (2017), a escola é a detentora oficial e formal dos mecanismos de validação do saber. Assim, para cumprir a sua função social, a escola precisa integrar os recursos tecnológicos nos processos de ensino, propondo novas metodologias e práticas pedagógicas. Em vista disso, diversas propostas vem sendo desenvolvidas para integrar cada vez mais os artefatos tecnológicos dentro de uma sala de aula, sem perder a essência de um centro de ensino.

A construção de novos cenários de aprendizagem, em consequência dessa demanda de integração de tecnologias em ambientes educacionais, bem como a utilização de ferramentas que colaborem com estes, vem ganhando destaque junto a sociedade. Isso acarreta em mudanças tanto no processo de ensino quanto no modo de avaliação dos estudantes. Cardoso e Burham (2007) explicam que novos fatores passam a fazer parte do processo de ensino-aprendizagem, como por exemplo, atividades ou comportamentos dos estudantes durante a aula, reconhecimento de contextos pelo ambiente de ensino, recomendações de objetos, dentre outros.

No entanto, de acordo com Leitão (2017), apenas criar ou utilizar recursos educacionais tecnológicos não garante a eficácia e nem a melhoria da qualidade da educação, já que o aproveitamento de uma aula está diretamente relacionado ao engajamento no processo de ensino-aprendizagem. Assim, a avaliação e *feedback*¹ automático surgem como ferramentas metodológicas que permitem classificar o desempenho dos alunos, identificando fatores relevantes, além de reduzir o trabalho excessivo dos professores, permitindo que estes possam se dedicar a outras tarefas dentro da sala de aula.

Neste trabalho, é apresentada uma proposta para automatizar o processo de avaliação e geração de *feedback* no contexto dessa adequação. O processo de automatização da tarefa de avaliar e classificar o desempenho de estudantes é demonstrado a partir da utilização de métodos formais, contidos na área de engenharia de software, que modelam o comportamento do aluno ao responder uma atividade avaliativa, de forma a garantir a consistência dessas avaliações. Este trabalho propõe que a automatização do *feedback* seja baseada em análises formais das propriedades desses modelos, verificando se os critérios de avaliação selecionados pelos professores são cumpridos nas soluções dos alunos.

¹Retorno de informação, retorno ou *feedback* é o procedimento que consiste no provimento de informação a uma pessoa sobre o desempenho, conduta, ou ação executada por esta, objetivando reorientar ou estimular comportamentos futuros mais adequados (Cardoso e Burham, 2007).

1.1 Contexto

A educação é um serviço e, como tal, sofre e se adequa às concepções paradigmáticas que vive a sociedade. Portanto, ela passa pelas mesmas transformações que outros segmentos da sociedade passam (Valente, 1999). Nesse contexto, Leitão (2017) afirma que a sociedade atual é profundamente marcada pela tecnologia, não como sendo a simples utilização de ferramentas, mas sim como parte integrante da vida das pessoas, moldando o modo como elas se relacionam consigo mesmas e com o mundo ao seu redor.

Relacionando esse pensamento à educação, Cardoso e Burham (2007) refletem que propostas vem sendo buscadas como meio de enriquecer as condições e as chances de aquisição e construção do conhecimento, adotando abordagens diferentes e complementares aos recursos “tradicionais” já existentes. Essa necessidade de integração de tecnologias juntamente com a transformação sofrida pela educação através do tempo resultou em esforços para desenvolver técnicas que auxiliem os profissionais em sala de aula ao mesmo tempo em que seja possível fornecer um retorno simples e objetivo aos alunos.

A avaliação e o *feedback* automático surgem aqui como uma solução, já que esse tipo de metodologia fornece orientações de ensino aos alunos, como por exemplo ajudando estes a compreender melhor sua nota e em que conteúdo precisam dar mais atenção, além de reduzir o trabalho excessivo de professores, permitindo que estes possam se dedicar a outras tarefas dentro da sala de aula.

1.2 Definição do Problema

De acordo com Valente (1999), “mudança” sempre foi a palavra de ordem na sociedade. A mudança citada por Valente implica em profundas alterações em praticamente todos os seus segmentos, afetando a maneira como atuamos e pensamos. O conhecimento e, portanto, os seus processos de aquisição assumem um papel de destaque nesse cenário.

O processo de automatizar a avaliação dos alunos, nesse sentido, consiste em permitir que um determinado artefato ou componente tecnológico seja capaz de avaliar uma solução como correta, baseando-se em critérios especificados pelo instrutor (Juniwal, 2014). Além disso, de acordo com Umbreja *et al.* (2014) e Pape *et al.* (2016), essa automatização traz ganhos de produtividade e de gerenciamento do tempo tanto dos professores quanto dos alunos.

Decorrente desse fato, a mudança pedagógica que todos almejam é a passagem de uma educação totalmente baseada na transmissão da informação e na instrução, para a criação de ambientes nos quais o aluno realiza atividades e constrói o seu conhecimento com o devido apoio tecnocientífico.

Em vista disso, a engenharia de *software*, como sendo a área que busca organização, produtividade e qualidade dentro de cenários que envolvem tecnologia (Pressman e Maxim, 2016), oferece diversos meios de suporte aplicáveis ao cenário de ensino apoiado por tecnologia, principalmente no que diz respeito às tarefas metodológicas contidas no processo de ensino-aprendizagem.

Um dos meios de suporte da engenharia de *software* que vem ganhando espaço em cenários de educação apoiada por tecnologia, por possibilitar que tarefas sejam verificadas por meio de provas construtivas e por fornecer garantias formais de correção, são os métodos formais. De acordo com Pressman e Maxim (2016), Woodcock *et al.* (2009) e Seshia (2015) esses métodos são técnicas

matemáticas, geralmente suportadas por ferramentas, usados para descrever propriedades de objetos por meios de especificações.

Dessa forma, o problema considerado neste trabalho é expresso na seguinte questão: *é possível gerar um método formal que gere avaliação automática consistente de atividades educativas baseada em critérios de aprendizagem específicos e, ao mesmo tempo, abra oportunidades para que futuros feedbacks sejam gerados e que ajude os alunos a identificar onde e porquê eventualmente erraram?*

1.3 Justificativa

As escolas buscam continuamente a integração da tecnologia da informação com os meios pedagógicos mas, como explicam Silva *et al.* (2014), essas tentativas incessantes limitam-se a apenas acoplar junto ao ambiente de ensino artefatos de *hardware* para que estudantes e professores possam utilizá-los.

Os recursos tecnológicos que são usados dessa maneira apenas servem como meio de instruir os estudantes sobre determinado conteúdo, não oferecendo qualquer oportunidade para que esse conhecimento seja trabalhado de alguma forma pelos professores. Rezende (2004) confirma essa ideia definindo o termo “aprender” como um processo de construção de significado e não um processo de transmissão de informação.

De acordo com Rezende (2004), ao incorporar a cultura tecnológica à educação incorporam-se também novas formas de ensino e aprendizagem. Então, não basta utilizar recursos alternativos ou tecnologias de ponta, se não se desenvolver capacidades estratégicas de como e quando usar tais recursos no processo de ensino-aprendizagem. Isso implica na necessidade de identificar meios que possam refinar/sofisticar outras tarefas incumbidas no processo de ensino, tais como:

- Avaliar o desempenho dos estudantes, englobando fatores como: permitir que o próprio professor selecione quais os critérios que ele deseja observar na avaliação do aluno e permitir que estes critérios possam ser alterados ou ajustados conforme seja necessário;
- Prover informações úteis (*feedback*) para que estes possam construir conhecimento por empenho próprio, além de servir como meio de geração para recomendação de aprendizagem para o próprio aluno.

Dentre as técnicas mais utilizadas na automatização dos processos de avaliação estão as métricas de similaridade (Thaler *et al.*, 2016), as técnicas de Aprendizagem de Máquina (Lan *et al.*, 2015) e o Processamento de Linguagem Natural (Shehab *et al.*, 2017), que utilizam modelos matemáticos e técnicas de identificação de padrões para inferir o desempenho dos alunos, seja em questionários de múltipla-escolha, seja em questões dissertativas.

Entretanto, o uso de métodos formais na automatização de processos avaliativos pode trazer ganhos em relação às outras técnicas utilizadas, pois, um modelo formal permite obter algum conhecimento sobre o desempenho atual de um aluno sem a necessidade de um grande conjunto de dados (aprendizagem de máquina) ou de uma correlação com outros alunos (métricas similaridade) e custo computacional relativamente baixo.

Seshia (2015) descreve como a utilização dos métodos formais afetam o ensino apoiado por tecnologia, relacionado ao seu uso com o propósito de avaliar o comportamento do aluno em sala de aula:

1. Durante a avaliação, um computador verifica se a solução fornecida por um aluno é “correta”, atendendo a determinados critérios selecionados pelo professor. Esses critérios são tratados pelos métodos formais como as especificações necessárias a serem verificadas;
2. Uma vez que o processo de avaliação julgue a solução como “incorreta”, o tipo de método usado pode ser analisado para fornecer *feedback* indicando onde o estudante errou.

Segundo Woodcock *et al.* (2009), o formalismo conhecido como rede de Petri tem apresentado contribuições relevantes nessa área, capturando modelos cognitivos de aprendizagem e usando-os para que os alunos, individualmente, construam seu próprio conhecimento.

Mas por que rede de Petri se destaca para aplicações desse tipo e não outros métodos, inclusive formais (autômatos, por exemplo)? Conforme explicam Cassandras e Lafortune (2009), como um autômato, uma rede de Petri é um dispositivo que manipula eventos de acordo com certas regras. Uma de suas características é que inclui condições explícitas sob as quais um evento pode ser ativado. Isto as permite ser convenientemente descrita graficamente, pelo menos para representações limitadas, resultando em esquemas gráficos intuitivos e que capturam muitas informações estruturais sobre o que está sendo modelado.

Por outro lado, nem todas as redes de Petri podem ser representadas como autômatos de estados finitos. Consequentemente, as redes de Petri podem representar uma classe maior de objetos. Outra motivação para considerar modelos de rede de Petri é o corpo de técnicas de análise que foram desenvolvidas para estudá-las. Essas técnicas incluem análise de alcançabilidade e técnicas algébricas lineares.

Conforme explica Murata (1989), as redes de Petri foram projetadas para modelar situações em que os recursos são concorrentes. Apesar disso, seus recursos podem ser utilizados em situações sequenciais, como por exemplo para representar o comportamento de um determinado objeto, levando em consideração todas as decisões tomadas durante esse comportamento. Com isso, é possível observar e extrair características relevantes do modelo utilizando a formalização e sistematização que a própria rede de Petri oferece em seu acervo.

1.4 Objetivos

O objetivo principal deste trabalho é *propor uma metodologia, utilizando redes de Petri, capaz de avaliar automaticamente estudantes em uma aula presencial de modo que seja possível provar/refutar que uma resposta/solução satisfaz critérios, ou métricas de correteude.*

Os objetivos específicos são:

1. Sintetizar automaticamente modelos formais, a partir do comportamento do estudante ao responder uma atividade avaliativa, e analisá-los, com base em critérios selecionados pelo professor, para gerar automaticamente as notas das avaliações dos estudantes;
2. Apresentar *feedback* de modo que, em caso de soluções incorretas, seja possível identificar os focos de maior dúvida dos estudantes e relacionar a ordem das questões respondidas com seus conteúdos;
3. Validar o método proposto aplicando-o em cenários reais de ensino baseado em educação digital.

1.5 Metodologia Proposta

A solução proposta neste trabalho busca oferecer benefícios para estudantes e professores envolvidos com o cenário de educação digital. Para os professores, a avaliação automática permite que parte do seu tempo seja dedicado a outras atividades acadêmicas com os estudantes, sem diminuir o rendimento de suas avaliações. Para os estudantes, esse tipo de sistema busca prover informações importantes sobre o processo de estudo que está sendo realizado e principalmente sobre seu desempenho, permitindo apontar falhas identificadas durante o processo.

A Figura 1.1 mostra uma visão geral deste trabalho, em que propõe-se uma metodologia composta por três etapas baseadas em pré-processo, processo de avaliação e processo de geração de *feedback*. Nesta seção, somente um resumo será descrito sobre cada componente do método proposto, mais detalhes serão apresentados no Capítulo 5.

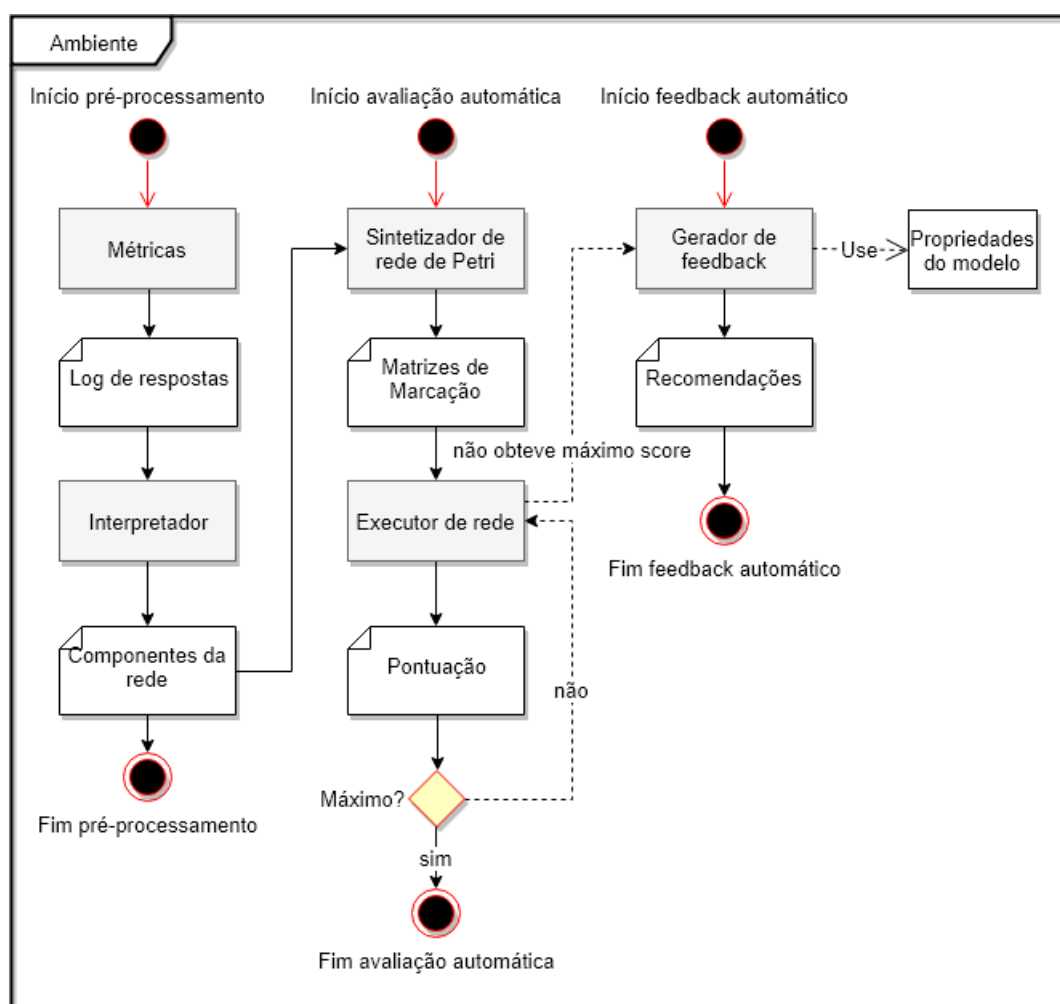


Figura 1.1: Visão geral do trabalho.

1. Pré-processamento – consiste na fase em que as informações iniciais de estudantes e professores são obtidas com a utilização de uma plataforma educacional e possui dois componentes, as “Métricas” e o “Interpretador”:

- **Métricas:** As métricas utilizadas neste trabalho são baseadas na abordagem proposta por Leitão (2017) para questionários de múltipla-escolha, tal como apresentado a seguir:

(i) Nota Tradicional: cálculo da média obtida pelo estudante como o produto do valor de cada questão pelo total de respostas corretas; (ii) Nota Ponderada: cálculo da média obtida pelo estudante levando em consideração os pesos das respostas finais; (iii) Nível de Dúvida: medida da incerteza de um estudante ao responder uma questão baseada na quantidade de vezes que uma resposta diferente da atual foi marcada; (iv) Nível de Desvio: valor entre 0 e 4 de uma resposta marcada pelo aluno, sendo 4 uma resposta totalmente correta, 0 uma resposta totalmente errada e 1, 2 e 3 valores intermediários de resposta; e (v) Nível de Confusão: relação entre o total de respostas corretas e a quantidade de marcações de respostas.

- **Interpretador:** O Interpretador analisa linha a linha os registros de ações dos alunos (ou *logs* gerados pelo ambiente educacional) e os transforma em componentes da rede de Petri. É este componente que identifica as questões e suas respostas, além de extrair outras informações relevantes para algumas métricas citadas acima.

2. Avaliação automática – esta etapa consiste no procedimento formal realizado sobre o comportamento do aluno modelado em rede de Petri e possui dois componentes: o “Sintetizador de rede de Petri” e o “Executor da rede”.

- **Sintetizador da rede de Petri** – é responsável por receber os componentes e sintetizar o modelo formal da rede de Petri, transformando-os em matrizes de marcações de entrada e saída para disparos da rede;
- **Executor da rede** – esse componente realiza os disparos das marcações do modelo formal em toda a estrutura da rede. É com base nessa execução que os cálculos dos critérios de correção são aplicados em cima das respostas do aluno, consequentemente gerando sua pontuação.

Nesse momento, a pontuação (de acordo com a métrica selecionada) é apresentada. O processo, como um todo, é finalizado.

3. *Feedback* Automático - esta etapa consiste em gerar o *feedback* formal para o aluno e para o professor. Primeiramente, ela faz uso do modelo e da execução da rede da etapa anterior. No entanto, ao invés de gerar a pontuação do aluno, o modelo tem algumas de suas propriedades estruturais e matemáticas analisadas. Seu único componente, o “Gerador de *feedback*”, busca extrair do modelo de rede de Petri as informações necessárias para gerar os *feedbacks* para as métricas selecionadas pelo professor, gerando assim recomendações acerca do desempenho do estudante.

Ao final de todo o processo envolvendo as três etapas, o *feedback* pode ser entregue ao aluno e professor. Com isso o aluno pode verificar em que pontos ele cometeu seus erros e o professor pode usar isso para identificar os conteúdos ao qual o aluno obteve menos êxito em suas respostas.

1.6 Organização do Trabalho

A introdução deste trabalho apresentou o contexto, definição do problema, justificativa, os objetivos e a solução proposta. Os capítulos restantes desta proposta estão organizados da seguinte forma:

No Capítulo 2 são discutidos os conceitos e definições abordados neste trabalho, tais como: educação digital, avaliações automáticas, *feedback* automático, métodos formais e redes de Petri.

No Capítulo 3 é apresentada e discutida a revisão sistemática realizada para fins de embasamento teórico deste trabalho. Em sua estrutura são apresentados itens do protocolo de revisão, bem como a execução e análise dos resultados. São apresentadas também as limitações desta revisão.

No Capítulo 4 são apresentados os trabalhos relacionados a esta proposta, produtos da revisão sistemática realizada. Os trabalhos são apresentados por métodos utilizados em seus contextos para gerar avaliação e *feedback* automático.

No Capítulo 5 é apresentado o método proposto utilizado para a obtenção dos resultados parciais. São apresentados em detalhes as informações de cada uma das etapas apresentadas na solução proposta.

No Capítulo 6 são apresentados os resultados obtidos pela execução do método discutido no capítulo anterior. São abordadas informações relevantes mostrando as limitações e até que ponto afetaram a confiabilidade das informações.

Por fim, o Capítulo 7 apresenta as considerações finais e os trabalhos futuros.

Capítulo 2

Conceitos e Definições

Este capítulo tem como objetivo apresentar os principais conceitos e definições abordados neste trabalho, tais como: educação digital, avaliações automáticas, *feedback* automático, métodos formais e redes de Petri.

2.1 Tecnologia aplicada a educação: uma visão sobre a educação digital

A tarefa de melhorar um sistema educacional, dinâmico e complexo, exige atuação em múltiplas dimensões e decisões fundamentadas, seguras e criativas. Valente (1999) já demonstrava que, de um lado há melhorias institucionais, que atingem instalações físicas e recursos materiais e humanos, tornando as escolas e organizações educacionais mais adequadas para o desempenho dos papéis que lhes cabem. De outro, há melhorias nas condições de atendimento às futuras gerações, traduzidas por adequação nos currículos e nos recursos para seu desenvolvimento, num nível tal que provoquem ganhos substanciais na aprendizagem dos estudantes.

Essas melhorias nas condições de atendimento às novas gerações podem ser descritas como a relação entre trabalho, ciência e tecnologia. Lima Filho e Queluz (2005) comentam que dentro desse aspecto, o processo de construção de ciência e tecnologia é compreendido como integrado ao processo de desenvolvimento de todo um complexo conjunto de práticas sociais e históricas, de saberes tácitos e de conhecimentos sistematizados que permitem a satisfação das necessidades humanas.

Essa compreensão de práticas sociais e históricas sobre a construção do saber apoiado por ciência e tecnologia é viabilizada pela utilização de artefatos tecnológicos que fazem a interface de comunicação com os profissionais da área de educação e os estudantes. Esses artefatos podem ser métodos, metodologias ou técnicas implícitas dentro de um processo, ou a utilização direta de meios físicos (computadores, *smartphones*, entre outros).

Uma abordagem muito comum sobre essas melhorias nas condições nas escolas é a utilização da tecnologia em atividades extra-classe, com o intuito de ter os artefatos computacionais na escola, porém não se busca modificar o esquema tradicional de ensino (Valente, 1999).

Em geral, esse tipo de abordagem tem sido adotado por escolas para que estas tenham a tecnologia implantada nas atividades educacionais, mas não estão interessados em resolver as dificuldades

que esses artefatos tecnológicos acarretam ao serem inseridos mais profundamente nos contextos disciplinares.

Uma pesquisa apresentada por Lima Filho e Queluz (2005) mostra a manifestação dos professores sobre essas relações entre tecnologia e trabalho ou atividades correlatas com tecnologia. As representações mais comuns foram:

- O uso da ciência, do saber, no trabalho, em busca de novos métodos para otimizar seu trabalho;
- Aplicação e implantação de conhecimentos científicos para o melhor desenvolvimento de trabalho;
- É o domínio de conhecimentos e técnicas específicas que permitem aos seus detentores modificar e aperfeiçoar um processo produtivo e do saber;
- Toda ciência do trabalho, pano de fundo para o conhecimento e domínio de sua disciplina;
- Toda e qualquer ação que otimiza o processo produtivo;
- Com a tecnologia é possível facilitar, agilizar e aperfeiçoar o trabalho, tornando-o competitivo.

Com essas informações, duas matrizes conceituais sobre tecnologia em contexto educacional foram propostas: (i) a matriz relacional, que compreende a tecnologia como construção, aplicação e apropriação das práticas, saberes e conhecimentos; (ii) a matriz instrumental, que compreende a tecnologia como técnica, isto é, como aplicação prática de saberes e conhecimentos.

Tomando como base a primeira matriz, podemos definir a tecnologia aplicada a educação (ou a própria educação digital) tratada aqui como sendo a metodologia que enfatiza o fato de o professor ter conhecimento sobre os potenciais educacionais dessa tecnologia e ser capaz de alternar adequadamente tarefas tradicionais de ensino-aprendizagem e tarefas que utilizam esses artefatos tecnológicos.

O aluno, por sua vez, utilizando esses artefatos tecnológicos continuará recebendo as mesmas informações que antes eram produzidas pelos métodos tradicionais de ensino-aprendizagem e terá um ambiente propício para construir seu próprio conhecimento. Valente (1999) afirma que essa construção do conhecimento advém do fato do aluno ter de buscar novos conteúdos e estratégias para incrementar o nível de conhecimento que já dispõe sobre o assunto que está sendo tratado.

Em vista do cumprimento desses dogmas da construção do conhecimento e da adaptação da tecnologia ao contexto educacional, vários meios de execução dessas tarefas foram propostos ao longo do tempo, por exemplo: ensino por correspondência, teleducação, ensino móvel, mundos virtuais, entre outros. Os mais conhecidos, atualmente, são os ambientes virtuais de aprendizagem.

Silva *et al.* (2014) contextualizam ambiente virtual de aprendizagem como sendo a virtualização da escola tradicional. Nesse sentido, a aplicação mais pura dessa definição não, necessariamente, estaria de fato proporcionando a construção do conhecimento, uma vez que permaneceria ligado ao conceito da escola tradicional, sendo apenas a adequação da tecnologia ao contexto educacional. A Figura 2.1 exemplifica esse tipo de ambiente.

Nessa abordagem o estudante apenas recebe e retém o conteúdo que o professor lhe repassa, sendo um caminho de mão única. Não existe qualquer estímulo para que a informação recebida seja trabalhada de alguma forma ou seja discutida com o professor.



Figura 2.1: Modelo de virtualização para adaptação tecnológica do ensino (Silva *et al.*, 2014).

Rezende (2004) evidencia isso afirmando que a aprendizagem não pode ser encarada por alunos e professores como uma atividade mecânica e simplista, dada a diversidade e a complexidade da realidade a que estamos inseridos. Os problemas exigem tomada de decisão, criatividade e a busca por inovação.

Então, não basta utilizar recursos alternativos, tecnologias de ponta, se não se desenvolver as capacidades estratégicas (por associação e repetição; por reestruturação, elaboração e organização; ou por aquisição e interpretação de informação; compreensão e organização conceitual; ou comunicação) de como e quando usar tais recursos no processo de ensino-aprendizagem.

Decorrente a esse fato, objetivando demonstrar como essa integração tecnológica com a educação ainda pode auxiliar no processo de construção do conhecimento, Papert (apud (Silva *et al.*, 2014); (Valente, 1999)) desenvolveu a abordagem construcionista em que o aprendizado ocorre por meio do fazer. Todavia, esse "fazer" apresenta um diferencial, que são os artefatos tecnológicos. Essa abordagem é exemplificada pela Figura 2.2.



Figura 2.2: Modelo de virtualização que permite ao aluno construir seu próprio conhecimento (Silva *et al.*, 2014).

Nessa abordagem, o aprendiz terá de refletir sobre o que fez e depurar suas ideias, objetivando chegar nas informações necessárias e incorporando-as à descrição da resolução de um problema, gerando assim um ciclo de descrição-execução-reflexão-depuração-descrição.

Valente (1999) classifica quatro características importantes sobre esse tipo de ambiente virtual de aprendizagem:

- Construção e não instrução: os estudantes são capazes de aprender mais efetivamente pela construção de seu próprio conhecimento, não sendo ensinados por meio da leitura ou por meio de uma sequência organizada de exercício e prática;
- Controle do estudante e não controle do sistema: o estudante tem um controle não exclusivo, porém mais significativo da interação na aprendizagem;
- Individualização é determinada pelo estudante e não pelo sistema: *feedback* e informação individualizada são chave na aprendizagem. Os estudantes recebem o *feedback* e as informações como função de sua interação com o sistema;
- *Feedback* rico: gerado pela interação do estudante com o sistema de aprendizagem e não pelo sistema. O *feedback* é gerado como função das escolhas e ações do estudante dentro do ambiente de aprendizagem.

É possível identificar por meio dessas características que o fator mais importante para que esse tipo de abordagem seja concretizada e aplicada a um cenário real é a presença de *feedback*. No entanto, o *feedback* mostra ser consequência do processo de avaliação feita sobre o desempenho do aluno em cima da sua interação com o sistema (mais especificamente, suas escolhas e ações sobre um determinado conteúdo).

Nesse contexto, é inserido o processo de automatização da avaliação e da geração de *feedback*. O processo de automatizar a tarefa de avaliação, segundo Juniwal (2014), consiste em permitir que um determinado artefato ou componente seja capaz de avaliar a solução de um aluno como correta, baseando-se em critérios especificados pelo instrutor. Adicionalmente, caso o avaliador automático interprete a solução como incorreta, o *feedback* deverá ser proposto como forma de mostrar onde e porque a solução é incorreta. Essas duas etapas serão descritas mais detalhadamente nas seções seguintes.

2.2 A automatização do processo de avaliação

Leitão (2017) comenta que a escola precisa integrar os recursos tecnológicos no processo de ensino, propondo novas técnicas e, principalmente, práticas pedagógicas. Em vista disso, diversas metodologias vêm sendo desenvolvidas para integrar cada vez mais esse conceito tecnológico dentro das salas de aula, sem perder a essência de um centro de ensino.

Como citado anteriormente, a virtualização da escola tem sido bastante difundida como meio de adequar essa necessidade ao cenário real da educação. Para Cardoso e Burham (2007), agregado a essa virtualização, existe a possibilidade de enriquecer as condições e as chances de aquisição e construção do conhecimento adotando diferentes abordagens complementares aos recursos “tradicionais” já existentes.

Segundo Bajaj e Seshia (2014), a automatização do processo de avaliação é indispensável nesses novos paradigmas de ensino. Essa avaliação é feita baseada em critérios escolhidos pelo professor

para medir a competência do aluno sobre determinado conteúdo. Umbreja *et al.* (2014) definem essa competência a ser avaliada como sendo o bom desempenho em contextos diversos e autênticos baseado na integração e ativação de conhecimento, regras, técnicas, procedimentos, padrões, habilidades, atitudes e valores.

Algumas motivações para integrar a automatização da tarefa de avaliação dentro do contexto educacional são descritas por Pape *et al.* (2016):

- Corrigir e classificar o desempenho do aluno sobre um determinado tema é uma tarefa demorada, tediosa e propensa a erros. Os recursos liberados pela redução desse fardo (permitindo que um artefato tecnológico realize essa tarefa) podem ser usados para melhorar o suporte de ensino para os alunos de outras maneiras, por exemplo, aumentando o tempo de discussão com os alunos ou criando exercícios adicionais;
- A avaliação automática supera o problema de que, para diferentes assistentes de ensino, o nível de detalhes das respostas e até mesmo a classificação em si poderia ser não uniforme;
- Outra vantagem das avaliações automáticas é que os alunos podem repetir os exercícios quantas vezes quiserem (por exemplo, para fins de treinamento geral ou para preparação de exames) e receber *feedback* a cada iteração.

Umbreja *et al.* (2014) assumem as mesmas considerações de Pape *et al.* (2016) afirmando que, como esse tipo de artefato busca ser automático e não requer intercepção de um instrutor para realizar a sua tarefa, é responsabilidade dele orientar o aluno no processo de aprendizagem e fornecer o *feedback* adequado de acordo com o erro cometido.

Observa-se que a utilização da avaliação automática em contexto educacional acarreta grandes contribuições tanto de produtividade quanto de gerenciamento de tempo para alunos e professores. Sua sobrecarga reduzida com a tarefa de avaliar o desempenho do aluno, podendo dedicar-se a outras atividades mais produtivas e o aluno tem a possibilidade de receber orientações sobre seu desempenho, bem como utilizar essas informações para aprimorar seu próprio nível de conhecimento.

Diversas técnicas ou metodologias são usadas como meio para atingir o objetivo desse tipo de avaliação, dentre elas podemos citar:

- Métricas de similaridade (Reischmann e Kuchen (2016) e Thaler *et al.* (2016)), nesse tipo de abordagens cálculos matemáticos são usados como meio de verificar a similaridade entre as soluções de estudantes e professores e costumam ser aplicados a cenários que consideram questões objetivas de múltipla-escolha;
- Aprendizado de máquina (Lan *et al.* (2015) e Waters *et al.* (2015)) e processamento de linguagem natural (Hsiao e Govindarajan (2016) e Shehab *et al.* (2017)), esses dois tipos de abordagens utilizam modelos matemáticos juntamente com técnicas de identificação de padrões para realizar uma avaliação sobre soluções de alunos e geralmente são aplicadas a tarefas do tipo dissertativa;
- Teoria dos grafos (Reischmann e Kuchen (2016) e Vujosevic-Janicic *et al.* (2013)), abordagens desse tipo geralmente são destinadas a cenários específicos de ensino, comumente é trabalhado em contexto de questões criativas que envolvem algum tipo de esquematização e utilizam algum tipo de métrica para avaliar se uma solução está correta ou não.

Entretanto, existe outro ramo de pesquisa dedicado a aplicação de bases mais formais e sistemáticas no contexto tecnológico da educação. Essa linha de pesquisa, segundo Woodcock *et al.* (2009), busca fornecer suporte automatizado necessário para verificar a integridade e verificabilidade de determinadas tarefas baseados em especificações sobre o que elas deveriam fazer. Essa tarefa só é possível utilizando métodos formais como apoio metodológico.

2.3 Da engenharia de *software* aos métodos formais

Primeiramente, é necessário estabelecer uma breve conexão entre o campo de estudos da engenharia de software e a diretriz original dos métodos formais. Pressman e Maxim (2016) comenta que o artefato *software* assume um duplo papel. É um produto e, ao mesmo tempo, o veículo para distribuir um produto.

Como produto fornece o potencial computacional representado pelo hardware. Como veículo de distribuição do produto, o software atua como a base para o controle do computador (sistemas operacionais), a comunicação de informações (redes) e a criação e o controle de outros programas (ferramentas de software e ambientes).

Para Monin e Hinchey (2003), uma série de fatores implicaram para a criação de uma área de estudo específica para o *software*, mas os principais foram:

- Uma série de noções chave foram reconhecidas, por exemplo, o conceito de ciclo de vida do software (comumente: requisitos, especificações, *design* geral e detalhado, codificação, teste unitário, teste de integração, instalação e manutenção). Nesse caso encarando o software como produto;
- A introdução de métodos rigorosos na produção do *software*, interpretando-o como veículo;
- O custo das diferentes etapas foi avaliado e comparado; Por exemplo, estima-se que a manutenção ocupa na lista dois terços do custo total de um projeto de *software*, e que corrigir um erro de especificação requer um esforço de vinte vezes mais se for detectado após o estágio de instalação, e as vezes muito mais do que isso.

Resumidamente, dominar a complexidade e o custo do *software*, que provaram ser um verdadeiro desafio tecnológico e econômico, deu origem a uma área bem estabelecida, nomeada de Engenharia de *Software*.

Adicionalmente, Pressman e Maxim (2016) apresentam uma visão menos crítica, mas que não deixa de estar de acordo com Monin e Hinchey (2003), da criação da Engenharia de *Software*. Em sua visão, a Engenharia de *Software* foi uma consequência do aumento da importância do *software*, onde a comunidade da área tentava desenvolver tecnologias que tornassem mais fácil, mais rápido e mais barato produzir e manter *software* de alta qualidade.

A Engenharia de *Software* foi inicialmente proposta para servir de base no processo de construção de artefatos tecnológicos, os *softwares*. Entretanto, como Pressman e Maxim (2016) afirma, o *software* é comumente incorporado a outras áreas: transportes, medicina, militar, industrial, entretenimento, entre outras. Acreditando na lei das consequências não intencionais, ainda há muitos campos em que outros efeitos podem ser obtidos por essas aplicações.

Em vista dessa realidade e considerando a observação de Monin e Hinchey (2003) sobre a criação da Engenharia de *Software* (de que um dos fatores que levaram a sua criação foi a necessidade de

métodos mais rigorosos no processo de produção), foram propostos métodos mais discretos que os convencionais, os métodos formais.

Segundo Woodcock *et al.* (2009), métodos formais são técnicas matemáticas, geralmente suportadas por ferramentas, para o desenvolvimento de *software* e sistemas de *hardware*. Pressman e Maxim (2016) contextualiza os métodos formais como técnicas com forte base matemática para descrever propriedades de sistemas. Esses métodos formais, proporcionam uma estrutura dentro da qual as pessoas podem especificar, desenvolver e verificar sistemas de maneira sistemática.

Monin e Hinchey (2003), de uma maneira geral, explicam que esse tipo de abordagem formal permite a escrita de especificações mais rigorosas, precisas e completas sobre um determinado cenário.

Compreende-se então que os métodos formais foram inicialmente propostos para servir de auxílio no processo de produção de artefatos tecnológicos (*softwares*). Eles oferecem como suporte, segundo Woodcock *et al.* (2009), embasamento matemático para a descrição de propriedades e características do cenário e, por meio dessas especificações, permitem a sua verificabilidade.

Entretanto, por mais que tomemos a premissa acima como verdadeira, Pressman e Maxim (2016) cita em seu texto que os métodos formais nunca terão um impacto significativo até que possam ser usados por pessoas que não os compreendem.

Referente a isso, Woodcock *et al.* (2009) comentam que a comunidade em si está focada em encontrar novas aplicações que transcendam os limites convencionais desses métodos, principalmente no que diz respeito à integração dos métodos formais a outras áreas.

2.3.1 Métodos formais como suporte educacional

Uma das aplicações diferenciais dos métodos formais fora do contexto de desenvolvimento de *software* que tem sido estudada é a utilização deles como ferramenta de suporte para ambientes educacionais. Segundo Abramson e Pike (2011), essa adaptação é possível por meio de especificações dentro do cenário de provas formais e modelagens matemáticas.

O conceito dessas provas formais usadas para essa aplicação pode ser facilmente compreendido pela justificativa de que, conforme Pressman e Maxim (2016), os métodos formais usam a teoria dos conjuntos e a notação lógica para criar uma definição clara dos fatos que podem ser analisados para melhorar (ou mesmo provar) a correção e a consistência em determinado contexto.

Seshia (2015) acrescenta que as especificações formuladas compõem as obrigações de prova, projetando componentes para cumprir essas obrigações e verificando, através de busca de verificação algorítmica, se esses componentes realmente atendem às suas especificações.

Em outras palavras, os métodos formais traduzem as especificações para uma representação mais discreta aplicando a notação e heurística de conjuntos para definir invariantes de dados, estados e operações para um determinado cenário.

De acordo com BenSassi e Laroussi (2013), a vantagem de se utilizar métodos formais em ambientes educacionais fica evidente por eles aumentarem significativamente a nossa compreensão do comportamento do aprendiz com o cenário existente, revelando inconsistências, ambiguidades e incompletudes que, de outra forma, não poderiam ser detectadas.

Seshia (2015) comenta que a razão de os métodos formais serem buscados para esse cenário foi o advento do processo de automatização das tarefas comumente realizadas pela virtualização das escolas. Outras áreas da computação também passaram a ser estudadas para essa finalidade, tais como: Inteligência Artificial, Aprendizado de Máquina, Engenharia de *Software*, Interação Humano-Computador, entre outras.

Conforme citado anteriormente por Umbreja *et al.* (2014) e Pape *et al.* (2016), essa virtualização apresenta uma série de problemas. Seshia (2015) apresenta em que ponto os métodos formais apresentam contribuições sobre eles:

- Classificação e comentários automáticos: na classificação automática, um programa de computador verifica se uma solução fornecida por um aluno é “correta”, ou seja, que atende a determinados critérios especificados pelo instrutor (a especificação). Além disso, e particularmente quando a solução está incorreta, o avaliador automático deve fornecer *feedback* ao aluno sobre onde ele/ela errou;
- Pedagogia: há uma vasta literatura no campo da educação sobre o uso de modelos cognitivos de aprendizagem para melhorar os processos de ensino e aprendizagem.

No contexto da classificação, várias técnicas de métodos formais são aplicáveis, incluindo a geração de testes sistemáticos, verificação formal, verificação de equivalência, técnicas de depuração para fornecer localização e *feedback* de erros, métodos para atribuir crédito parcial ou extra, etc.

No que se refere à pedagogia, os métodos formais podem contribuir fornecendo formalismos matemáticos rigorosos para capturar os modelos cognitivos juntamente com técnicas algorítmicas para usar esses modelos para melhor personalizá-los para alunos, individualmente, e configurações de classe.

2.3.2 Métodos formais aplicado ao *software* e métodos formais aplicado a educação

É essencial explicar de que forma a utilização dos métodos formais como suporte educacional se diferencia da função original ao qual eles foram propostos. Seshia (2015) estabelece algumas diretrizes:

- Escala menor de problema: o tamanho dos problemas é menor para educação do que para *softwares*, como por exemplo ter que lidar com todas as fases do desenvolvimento em um projeto;
- Maior diversidade de soluções: para uma especificação de problema fixo, é preciso lidar com uma maior diversidade de soluções no domínio da educação do que no domínio específico;
- Disponibilidade de referências para soluções: na educação, normalmente existe acesso a uma solução “modelo” fornecida pelo instrutor, enquanto que na prática de desenvolvimento de *software* é necessário seguir com as especificações parciais do comportamento desejado;
- Necessidade limitada de verificação exaustiva: na educação, talvez não seja necessário que a solução seja correta para todo o espaço de insumos, mas isso geralmente é necessário para aplicações de *software*;

- Necessidade de *feedback* rápido: na educação, é essencial que os alunos recebam *feedback* rápido e oportuno, por exemplo, na ordem de alguns minutos, de modo a ajudá-los a iterar rapidamente para aprender um conceito. Em contraste, para usos industriais de métodos formais, pode ser aceitável aguardar muito mais tempo para receber informações de depuração;
- *Feedback* ajustável: na educação, pode-se querer dar diferentes graus de *feedback* para melhorar a experiência de aprendizagem - por exemplo, apenas revelando uma sugestão de alto nível para o aluno, permitindo que ele descubra a falha na própria solução, em vez de revelar uma linha detalhada de informações a nível numérico sobre seus erros. Em contraste, na prática industrial, procura-se fornecer ao engenheiro tanta informação de depuração de modo que este possa isolar e corrigir rapidamente os erros.

Com essas informações podemos extrair as seguintes vantagens sobre a utilização dos métodos formais no cenário educacional: os métodos formais podem fornecer *feedback* mais detalhado; provar, por meio de fortes garantias teóricas, sobre sua performance de avaliação; podem ser efetivos onde há especificações de problemas precisos.

2.4 As Redes de Petri

Os métodos formais são um conjunto de técnicas formais que utilizam a teoria dos conjuntos e heurísticas para representar especificações de determinados componentes em cenários específicos. De acordo com Monin e Hinchey (2003), existem diversas técnicas que podem ser citadas aqui, como por exemplo:

- Notação Z, que é uma linguagem formal de especificação para descrever e modelar sistemas computacionais, sendo baseada na notação matemática padrão usada no axioma da teoria dos conjuntos, cálculos lambda, e lógica predicada de primeira ordem. Todas as expressões na notação Z são tipadas, conseqüentemente desviando alguns dos paradoxos da Teoria Ingênuos dos Conjuntos;
- VDM (*Vienna Development Method*), uma linguagem com base em descrições semânticas interpretadas por funções matemáticas. É comumente usada para especificar sistemas orientados a objetos e/ou concorrentes e costuma ser aplicado a engenharia de sistemas críticos, compiladores, sistemas concorrentes, entre outros;
- Método B, que pode ser entendido como uma extensão da notação Z. Diferencia-se de Z por prover cobertura as etapas de especificação, refinamento e implementação. Incrementalmente à Z, utiliza a notação de máquina abstrata, um modelo teórico de sistema computacional de hardware e software usado para detalhar o funcionamento do sistema.

No entanto, segundo Custódio (2009), existe uma técnica que chamou bastante atenção dos estudiosos na área por desenvolver muito da teoria, notação e representação dos que os métodos formais buscam. Essa técnica é conhecida como redes de Petri. Esta, por sua vez, foi apresentada na tese de doutoramento de Carl Adam Petri intitulada de “*Kommunikation mit Automaten*”, em 1962 na Faculdade de Matemática e Física da Universidade de *Darmstadt*, na então Alemanha Ocidental.

2.4.1 Visão geral

Inicialmente, é essencial entender que as redes de Petri são técnicas baseadas em modelos de redes. De acordo com Murata (1989), essas técnicas de rede modelam esquemas por meio de mecanismos implícitos de fluxo de *tokens* (marcações) na rede. Este fluxo é controlado por condições que habilitam a realização de tarefas, chamados de eventos.

Uma rede de Petri é composta por uma estrutura de rede, por inscrições associadas a essa estrutura e uma marcação. De acordo com Gorgônio (2001), a estrutura da rede e as inscrições definem a sintaxe de uma rede de Petri. A evolução de suas marcações, de acordo com uma regra de ocorrência (especificação), estabelece a sua semântica.

Custódio (2009) cita que essa estrutura pode ser vista como um grafo com comentários. Este grafo possui nós de posição, nós de transição e arcos direcionados conectando posições com transições e vice-versa. Godoy *et al.* (2010) descrevem essa estrutura conforme mostra a Figura 2.3:

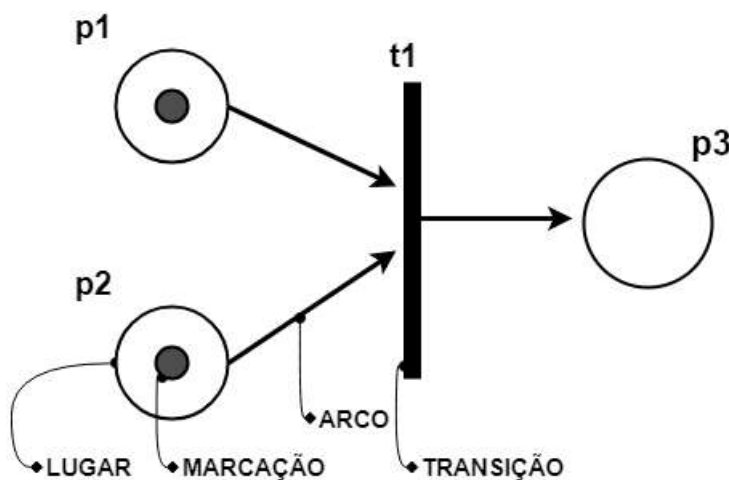


Figura 2.3: Exemplo da estrutura de uma rede de Petri.

Em que:

- Lugares (posição): graficamente são representados por um círculo e modelam os estados do sistema. Representam as condições para que determinado evento aconteça;
- Transições: graficamente são representadas por um retângulo (ou uma barra) e representam os eventos do sistema, ou seja, as mudanças de estados;
- Arcos: graficamente são representados por uma seta, interligando um lugar e uma transição, e representam como é a sequência com que acontecem as mudanças de estado dentro do sistema;
- Marcas: graficamente são representadas por um círculo de cor preta e representam o comportamento dinâmico do sistema, indicando os estados ativos, ou seja, que estão sendo utilizados pelo sistema.

Essa descrição pode ser complementada por Custódio (2009) e Barreto (2005), explicando que essas transições agem em marcações por um processo denominado disparo. Quando uma transição é disparada, ela consome as marcações de seus lugares de entrada, realiza alguma tarefa de processamento, e realoca um número específico de marcações nos seus lugares de saída.

2.4.2 Definição formal

Uma rede de Petri, segundo Barreto (2005) e Custódio (2009) baseados em Petri (1966), pode ser descrita formalmente como sendo uma quintupla composta por (P, T, F, W, M_0) , onde:

- $P = \{p_1, p_2, \dots, p_m\}$, um conjunto finito de lugares;
- $T = \{t_1, t_2, \dots, t_n\}$, um conjunto finito de transições;
- $F \subseteq (P \times T) \cup (T \times P)$, um conjunto de arcos onde lugares se ligam a transições ou transições a lugares, nunca dois elementos da mesma natureza tem uma ligação direta;
- $W : F \rightarrow \mathbb{N}$, uma função de atribuição de pesos dos arcos, sobre o conjunto dos números naturais;
- $M_0 : P \rightarrow \mathbb{N}$, uma função de mapeamento da marcação inicial da rede, sobre o conjunto dos números naturais.

Tomando como exemplo a Figura 2.3, agora usando a definição formal de uma rede de Petri, podemos descreve-la como sendo:

- $P = \{p_1, p_2, p_3\}$;
- $T = \{t_1\}$;
- $F = \{(p_1, t_1), (p_2, t_1), (t_1, p_3)\}$;
- $W = \{((p_1, t_1) = 1), ((p_2, t_1) = 1), ((t_1, p_3) = 1)\}$;
- $M_0 = \{1, 1, 0\}$.

2.4.3 Propriedades matemáticas

O estado de uma rede de Petri pode ser representado por um vetor M , no qual o primeiro valor é a quantidade de marcações na primeira posição da rede, o segundo é a quantidade de marcações na segunda posição, e assim por diante. Tal representação descreve completamente o estado de uma rede Petri.

Conforme explicam Barreto (2005) e Custódio (2009), uma lista de transição de estados $\sigma = \langle M_{i_0}, M_{i_1}, \dots, M_{i_n} \rangle$, que pode ser simplificada em $\vec{\sigma} = \langle t_{i_1}, \dots, M_{i_n} \rangle$ é chamada de uma sequência de disparo se cada transição satisfaz o critério de disparo, isto é, se existem marcações o suficiente na entrada de cada transição. Nesse caso, a lista de transição de estados de $\langle M_{i_0}, M_{i_1}, \dots, M_{i_n} \rangle$ é chamada de trajetória, e M_{i_n} é chamado de alcançável a partir de M_{i_0} por meio da sequência de disparos de $\vec{\sigma}$. Matematicamente: $M_{i_0}[\vec{\sigma}] > M_{i_n}$. Todas as sequências de disparo que pode ser atingida em uma rede N com estado inicial M_{i_0} são denotadas como $L(N, M_{i_0})$.

A propriedade $M_{i_0}[\vec{\sigma}] > M_{i_n}$ é complementada por Custódio (2009) utilizando a matriz de transição de estados. Essa matriz é dada por W^- composta por $|T|$ e $|P|$ e representa a quantidade de marcações passadas por cada transição de cada posição. Similarmente, W^+ representa a quantidade de marcações dadas por cada posição para cada transição. A soma dos dois, $W^- + W^+$, pode ser utilizada para calcular $M_{i_0}[\vec{\sigma}] > M_{i_n}$.

Essa expressão pode ser escrita como $M_{i_0} - M_{i_n} = W^T \cdot \sigma$, onde σ é um vetor de quantas vezes cada transição foi disparada na sequência, levando em consideração que existem marcações o suficiente para cada transição ser disparada. Essa alcançabilidade de estados por matriz, pode ser representada por um grafo de alcançabilidade, o qual tem seus destinos direcionados representando estados (M , por exemplo), e transições de arcos entre dois dos estados.

2.4.4 Extensões das redes de Petri

Conforme Barreto (2005), de Souza *et al.* (2007) e Gorgônio (2001) afirmam, existem diversas extensões das redes de Petri. Com o tempo, o termo redes de Petri passou a ser utilizado para generalizar um conjunto de modelos dessa classe. Dentre os principais estão:

- As redes de Petri elementares, as redes de Petri lugar/transição e as redes de Petri temporais, classificadas como redes de baixo nível por terem marcações indistintas, consequentemente tendo transições com ocorrências instantâneas.
- As redes de Petri predicado/transição e as redes de Petri coloridas, que são conhecidas como redes de Petri de alto nível. Esse tipo de rede caracteriza-se, principalmente, pela incorporação da teoria de tipos de dados. As marcações usadas nessas redes podem carregar informação complexa, manipulada pelo uso de uma linguagem.

As redes classificadas como alto nível expressam informação complexa aumentando o poder de descrição dessas redes e, consequentemente, modelos mais compactos podem ser obtidos em relação as redes de baixo nível. Essa flexibilidade na manipulação da informação permite distribuir a complexidade do modelo entre as inscrições e a estrutura da rede.

2.4.5 Redes de Petri como técnica de avaliação

Como citado anteriormente, de um ponto de vista pedagógico, a utilização de formalismos matemáticos pode ajudar a capturar modelos cognitivos de aprendizagem e juntamente com técnicas algorítmicas pode ajudar o próprio aluno a construir seu próprio conhecimento. Isso também permite ao professor dedicar parte do seu tempo a atividades mais produtivas em um ambiente escolar ao invés de avaliar individualmente cada aluno.

Dentro desse contexto, de Souza *et al.* (2007) revelam que as redes de Petri podem ser facilmente adaptáveis a esse cenário por elas permitirem a formalização de processos, implicando em uma maior inserção das características de determinada tarefa, por meio das suas especificações, podendo ser analisadas.

De fato, de acordo com Posada *et al.* (2016), as redes de Petri vêm sendo aplicadas cada vez mais para avaliações pedagógicas e avaliações formativas, além de estar presente em áreas que buscam a representação do conhecimento.

O processo de uma avaliação usando as redes de Petri, adaptado de Shen *et al.* (2012), pode seguir um fluxo, como visto na Figura 2.4:

Como pode ser visto na Figura 2.4, primeiramente faz-se necessário que o professor defina o tipo de critério que ele pretende utilizar para avaliar o desempenho do aluno. A partir dessa informação, serão geradas as especificações do modelo da rede de Petri que será sintetizada em seguida. Uma vez que o modelo esteja pronto, a nota do aluno será calculada pela execução da estrutura da rede.

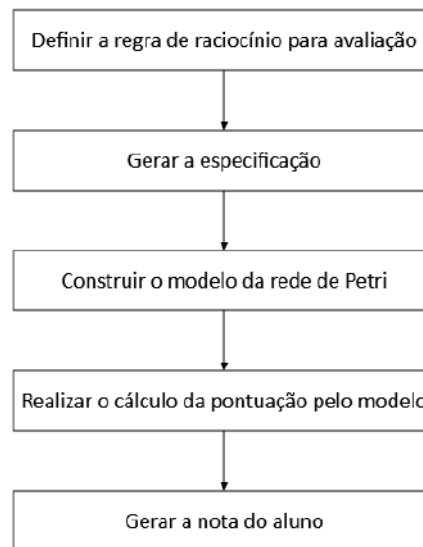


Figura 2.4: Fluxo do processo de avaliação utilizando redes de Petri.

2.5 A automatização do *feedback*

A tarefa de trabalhar os dados avaliados e enviá-los de volta ao aluno e professor foi classificada por Seshia (2015) como *feedback* ajustável. Por exemplo, apenas revelando uma sugestão de alto nível para o aluno, permitindo que ele descubra a falha na própria solução, em vez de revelar uma linha detalhada de informações a nível numérico sobre seus erros.

Definido o processo de automatização da tarefa de avaliação no contexto da educação digital utilizando redes de Petri, precisamos lembrar das características definidas por Valente (1999) de que a aplicação de tecnologias dentro do ambiente escolar deve fornecer ao aluno e professor informações claras e consistentes para que estes possam usá-las para construir conhecimento.

Em vista disso, o *feedback* abordado neste trabalho, busca utilizar as informações formais geradas nas avaliações automáticas e propor meios do estudante compreender seus erros para que possa adquirir consciência das suas falhas e possa ser direcionado para um conteúdo específico.

Entretanto, conforme citado anteriormente, corrigir e classificar o desempenho individual de cada aluno sobre um determinado tema é uma tarefa demorada, tediosa e propensa a erros. Então, é irrelevante automatizar a tarefa de avaliar e não prover recursos para que esse *feedback* seja, por tabela, automatizado também.

Tomando como premissa a utilização de métodos formais como meio de automatizar o processo de avaliação, mais especificamente a utilização de redes de Petri, Cunha *et al.* (2011) apontam a verificação de propriedades como um apoio ferramental para esse propósito. Segundo Baier *et al.* (2008), esse tipo de técnica de verificação baseada em modelos permite descrever o comportamento do sistema de forma matematicamente precisa e inequívoca. Essas propriedades podem estar relacionadas a estrutura do modelo da rede de Petri ou a características semânticas que o modelo deveria atingir.

As redes de Petri modelam esquemas por meio de mecanismos implícitos de fluxo de *tokens* (marcações) na rede. Este fluxo é controlado por condições que habilitam a realização de tarefas (eventos). Essa estrutura pode ser vista como um grafo com comentários. Este grafo possui nós de posição, nós de transição e arcos direcionados conectando posições com transições e vice-versa.

Uma das principais propriedades a ser verificada, no contexto desta proposta, para as redes de Petri é a alcançabilidade. Essa propriedade é utilizada para representar o desdobramento de todos os estados possíveis alcançáveis das redes de Petri. A propriedade é verificada definindo o estado inicial como M_0 , juntamente com todas as possibilidades possíveis de transição a partir desse estado e dos subsequentes, à medida que a execução da rede progride.

Dessa forma, é possível percorrer todos os estados da rede, enquanto seus eventos e marcações vão sendo trabalhados pela estrutura e por fim, teremos os estados finais que são alcançáveis a partir da marcação inicial. A propriedade de alcançabilidade é melhor definida no contexto desta proposta no Capítulo 5, em que o método utilizado é explicado.

Com isso, é possível extrair informações relevantes sobre o modelo da rede e sua execução, tais como quantas vezes ele respondeu uma questão ou quais as respostas que ele marcou para uma questão específica, entre outras. Essas informações acarretam em dados que podem ser trabalhados pelos professores, de forma a identificar os pontos em que o estudante é falho, seja em suas respostas ou suas dúvidas.

Essa verificação é automatizada em cima do modelo de avaliação, ajudando a propor recomendações que façam uso das informações extraídas para direcionar o foco e a atenção dos estudantes para seus pontos de maior dificuldade.

Prover *feedbacks* individualizados sobre o desempenho dos estudantes acerca de um determinado assunto é uma das grandes contribuições que os sistemas de avaliação e geração de *feedback* automático fornecem.

No entanto, conforme explicam Edmison *et al.* (2017), mesmo que esses sistemas sejam extremamente úteis ao fornecer o *feedback*, a maneira como essa informação é mostrada pode ser um problema.

De acordo com Danilov *et al.* (2016), existem diversas técnicas que buscam facilitar a exibição de dados abstratos por meio do conceito de cores, buscando aumentar o nível de visualização de dados e acelerar o processo de análise destes.

Edmison *et al.* (2017) afirmam que há muitas vantagens em utilizar esse tipo método visual em cima de *feedbacks*, dentre elas:

- Ele fornece *feedback* abrangente sobre o objeto avaliado, em vez de comentários textuais que são muito mais limitados;
- Fornece informações de localização dentro da solução do aluno, e não uma descrição orientada a recursos;
- Ao fornecer informações sobre a avaliação inteira, permite que o aluno escolha qual área deseja investigar;
- Pode ser totalmente automatizado, evitando trabalho adicional dos professores para escrever sugestões ou mensagens de *feedback* para os estudantes, ou para considerar como os alunos podem interpreta-lo.

2.6 Considerações finais

Este capítulo descreveu os principais conceitos necessários para o entendimento deste trabalho. Na Seção 2.1 foi dada uma visão geral sobre a origem e definições do termo educação digital, bem

como a apresentação do principal meio de aplicação de tecnologia à educação, os ambientes virtuais de aprendizagem.

Na Seção 2.2 é explicado como funciona o processo de avaliação automática, bem como seus fundamentos. Seguidamente na Seção 2.3, são apresentados os conceitos aplicáveis da tecnologia sobre a educação e as definições de métodos formais, mostrando como os métodos formais podem ser aplicados a esta tarefa por meio de um panorama entre sua função de origem e sua função com a educação.

A Seção 2.4 apresenta os conceitos e definições formais sobre os modelos conhecidos como rede de Petri. É apresentado sua descrição matemática juntamente com suas propriedades e extensões, além de mostrar como o modelo se aplica a educação digital, no contexto da avaliação automática.

Na Seção 2.5 são apresentados os conceitos necessários para o entendimento de como funciona o processo de geração automática de *feedback* com base na verificação de propriedade do modelo de rede de Petri e é descrito como o complemento visual ao método (buscando o simbolismo de cores), que busca reduzir a complexidade de visualização dos *feedbacks* gerados pode agregar contribuições a proposta.

Capítulo 3

Revisão Sistemática sobre métodos, técnicas e metodologias usadas para avaliação e *feedback* automático

Este capítulo tem como objetivo apresentar um levantamento bibliográfico, baseado na técnica de revisão sistemática da literatura, com o objetivo de identificar e conhecer quais métodos, metodologias ou técnicas estão sendo aplicados para automatizar o processo de avaliação e *feedback* automático em ambientes educacionais. Essa identificação é feita extraíndo suas principais características e avaliando suas vantagens e desvantagens dentro dos cenários em que foram aplicadas. Um dos principais objetivos desta revisão sistemática é selecionar métodos para serem aplicados neste trabalho.

3.1 Execução da revisão sistemática

A revisão sistemática é uma forma de identificar, avaliar e interpretar todas as pesquisas disponíveis relevantes para uma questão de pesquisa particular. Uma das razões para a realização de revisões sistemáticas é que esta resume as evidências existentes em relação a um tratamento ou tecnologia (Kitchenham, 2004).

A aplicação de revisão sistemática da literatura requer que seja seguido um conjunto bem definido e sequencial de passos, segundo um protocolo de pesquisa desenvolvido apropriadamente. Este protocolo é construído considerando um tema específico que representa o elemento central da investigação. Os passos da pesquisa, as estratégias definidas para coletar as evidências e o foco das questões de pesquisa são definidas explicitamente, de tal forma que outros pesquisadores sejam capazes de reproduzir o mesmo protocolo de pesquisa e de julgar a adequação dos padrões adotados no estudo.

Diante disso, essa revisão foi conduzida de acordo com o objetivo descrito pela Tabela 3.1, fundamentada no paradigma GQM (*goal, question and metric*) descrito por Basili *et al.* (1994).

Esse levantamento bibliográfico buscou responder a seguinte questão principal de pesquisa:

Tabela 3.1: Objetivo da revisão segundo o paradigma QM.

Analisar	Publicações científicas relacionadas a métodos de avaliação e geração de <i>feedback</i> automático
Com o propósito de	Identificar/Caracterizar
No que diz respeito a	Conceitos, metodologias e métodos de avaliação e geração automática de <i>feedback</i>
Do ponto de vista do	Pesquisador
No contexto	Ambientes virtuais de aprendizagem

- **QP:** Quais conceitos, métodos, metodologias, técnicas e ferramentas relacionados a avaliação e geração de *feedback* automático podem ser encontrados na literatura dentro do contexto de ambientes virtuais de aprendizagem?

Essa questão pode ser fragmentada em três sub indagações, são elas:

- Q1: Quais os métodos, metodologias, técnicas e ferramentas aplicados para produzir a avaliação automática de alunos em ambientes virtuais de aprendizagem?
- Q2: Quais métodos, metodologias, técnicas e ferramentas são capazes de gerar *feedbacks* automáticos baseados nos métodos usados como avaliação automática, satisfazendo critérios previamente definidos?
- Q3: Em que cenários de ambientes virtuais de aprendizagem esses conceitos são aplicados na literatura?

Para que um trabalho fosse selecionado como relevante neste levantamento sistemático, ele deveria estar escrito no idioma inglês e estar de acordo com pelo menos um dos seguintes critérios de inclusão:

- C.I 1: Descrever tecnologias (técnicas, métodos, modelos, ferramentas, abordagens) utilizadas para prover avaliação automática;
- C.I 2: Descrever tecnologias (técnicas, métodos, modelos, ferramentas, abordagens) utilizadas para prover *feedback* automático;
- C.I 3: Descrever customização de técnicas ou métodos aplicados a geração de avaliação automática em cenários específicos;
- C.I 4: Descrever customização de técnicas ou métodos aplicados a geração de *feedback* automático em cenários específicos.

Trabalhos rejeitados são justificados por pelo menos um dos seguintes critérios de exclusão:

- C.E 1: Trabalho não é aplicável a nenhum dos critérios de inclusão;
- C.E 2: Publicação não disponível;
- C.E 3: Publicação não atende a delimitação de idioma;

- C.E 4: Publicação não apresenta ambientes virtuais de aprendizagem (em uma visão educacional) como população;
- C.E 5: Publicação dispõe apenas de conceitos em seu conteúdo;
- C.E 6: A metodologia usada é restrita a uma área específica.

O ambiente utilizado para execução da revisão foi o de bibliotecas digitais. Logo, optou-se pela biblioteca digital *Scopus*¹. Com esta biblioteca foi possível consultar e acessar diferentes fontes digitais via *web* por meio de expressões de busca pré-estabelecidas.

A expressão de busca utilizada neste levantamento bibliográfico é fundamentada na estrutura descrita na Tabela 3.2.

Tabela 3.2: Estrutura usada para elaborar a expressão de busca.

População	Ambientes virtuais de aprendizagem
Intervenção	Definir uma abordagem (formal ou não) para avaliação automática e geração de <i>feedback</i> em ambientes virtuais de aprendizagem.
Controle	N/A
Outputs (Resultados)	Técnicas/Métodos/Ferramentas para geração de avaliação e <i>feedback</i> automático

Com isso, a expressão usada para obter a base de dados inicial do levantamento bibliográfico foi:

- (“*education*” OR “*educational environment*” OR “*virtual learning environment*” OR “*e-learning*” OR “*e-education*” OR “*engineering education*”) AND ((“*grading*” OR “*automatic grading*” OR “*automatic evaluation*” OR “*grading system*” OR “*learning evaluation*” OR “*computerized grading*”) AND (“*feedback*” OR “*automatic feedback*”)) AND (“*approach*” OR “*method*” OR “*technique*” OR “*methodology*” OR “*formal*”)

Com base nas publicações retornadas pela execução dessa expressão de busca e nos critérios de inclusão e exclusão, uma série de dois filtros foram aplicados com o intuito de selecionar os trabalhos considerados relevantes para esta revisão.

O primeiro filtro consiste na seleção dos trabalhos por meio da análise preliminar do título, resumo (*abstract*) e palavras chaves. O segundo filtro consiste na leitura íntegra das publicações selecionadas no primeiro filtro, visando identificar se as publicações de fato se aplicam aos critérios de inclusão, ou seja, se descrevem alguma contribuição relevante sobre métodos usados para o propósito do estudo (Tabela 3.1).

Ao final do segundo filtro foi realizada a fase de extração das seguintes informações dos trabalhos selecionados:

1. Título: o título do trabalho;
2. Resumo: uma breve descrição do estudo;
3. Palavras-Chave: palavras-chaves do trabalho;

¹<https://www.scopus.com/>

4. Autor(es): o(s) autor(es) do trabalho;
5. Fonte: onde o trabalho foi publicado;
6. Ano de publicação: ano de publicação do trabalho;
7. Tipo de estudo: pesquisa formal, empírica ou exploratória;
8. Método utilizado para produzir a avaliação automática;
9. Método utilizado para gerar *feedback* automático;
10. Cenário de aplicação;
11. Arquitetura apresentada;
12. Teste estatístico;
13. Limitações do(s) método(s);
14. Subárea(s) da pesquisa;
15. Perspectivas futuras: questão de pesquisa sugerida para trabalhos futuros, caso haja;
16. Comentários: observações gerais sobre o estudo, caso necessário.

O item “Tipo de estudo” é um critério de classificação qualitativo e baseia-se na definição de estudo definida por Wazlawick (2017), onde:

- Pesquisas formais exigem a elaboração de uma teoria e de uma prova formal de que essa teoria é correta. A lógica formal é a grande ferramenta de trabalho desse tipo de pesquisa e embora seja mais difícil de realizar, seus resultados, quando obtidos são mais difíceis de refutar;
- Pesquisas empíricas apresentam uma nova abordagem que é comparada com outras por meio de testes aceitos pela comunidade. Os métodos estatísticos são a grande ferramenta desse tipo de pesquisa. No entanto, mesmo baseada em testes estatísticos, poderá ser refutável se não estiver também embasada em uma boa teoria, porque a estatística não explica as causas;
- Pesquisas exploratórias não conseguem provar uma teoria ou resultados estatisticamente aceitos. São comuns desse tipo de estudo abordagens do tipo estudo de caso, análises qualitativas e pesquisas em áreas emergentes. A argumentação e o convencimento são a principal ferramenta desse tipo de pesquisa.

3.2 Análise e publicação dos resultados

A revisão foi conduzida por meio do manuseio da ferramenta *StArt*² e iniciou com 130 trabalhos retornados pela máquina de busca. Na Figura 3.1 é possível observar que dos 130 trabalhos retornados e aplicado o primeiro filtro, 82 foram aceitos para o próximo, 45 foram rejeitados e 3 foram considerados duplicados. O primeiro filtro analisa as seguintes informações: título, resumo/*abstract* e palavras-chave. Já no segundo filtro, 25 trabalhos foram aceitos e 57 foram

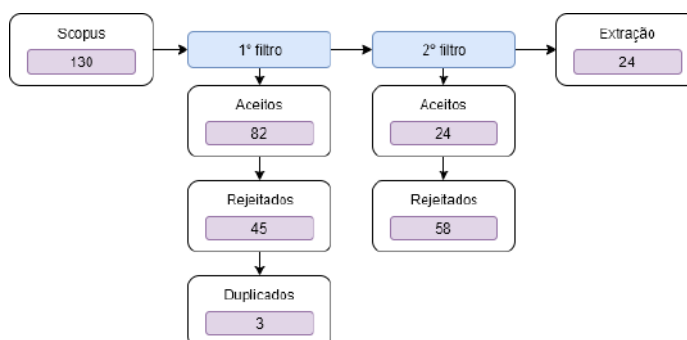


Figura 3.1: Processo de seleção das publicações.

rejeitados, não havendo duplicados durante essa etapa. Essa fase consistiu em uma leitura minuciosa dos trabalhos. Esses 25 trabalhos aceitos passaram então para a fase de extração.

O gráfico apresentado pela Figura 3.2 apresenta o total de publicações por ano, das 24 classificadas para a fase de extração.

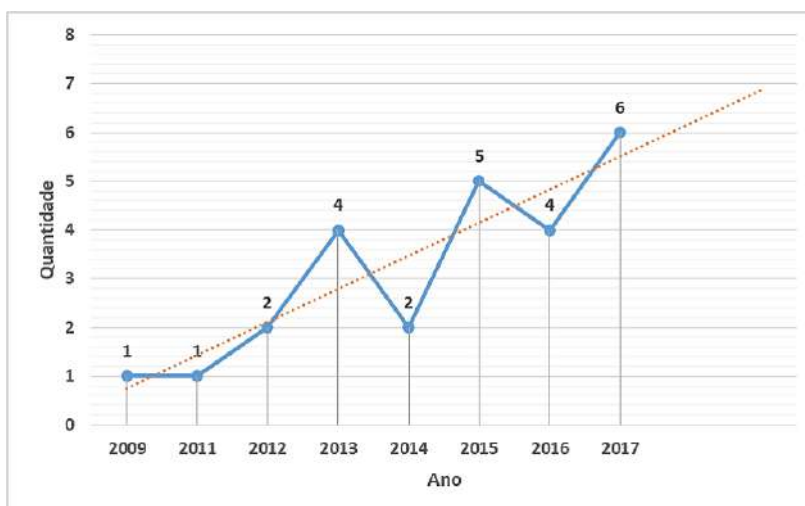


Figura 3.2: Relação da quantidade de trabalhos retornados por período.

É possível observar uma linha de tendência linear crescente com o passar dos anos. Para realização da revisão sistemática, o ano de publicação não foi delimitado. Sendo selecionados trabalhos publicados entre 2009 e 2017, demonstrando que as pesquisas nessa área são recentes principalmente por ter o maior índice no ano de 2017.

A Figura 3.3 demonstra a relação dos critérios de aceitação atribuídos a esses 25 trabalhos resultantes do segundo filtro.

Como pode ser notado, a maior parte dos trabalhos descreve alguma customização de método para automatizar o processo de avaliação e de geração de *feedback*. Em contrapartida, os trabalhos rejeitados no segundo filtro, em que foi feita a leitura na íntegra do texto, apresentam altos índices de aplicações de técnicas específicas para essas duas tarefas, mas são acrescidos de algum critério de exclusão, conforme mostra a Figura 3.4.

²http://lapes.dc.ufscar.br/tools/start_tool

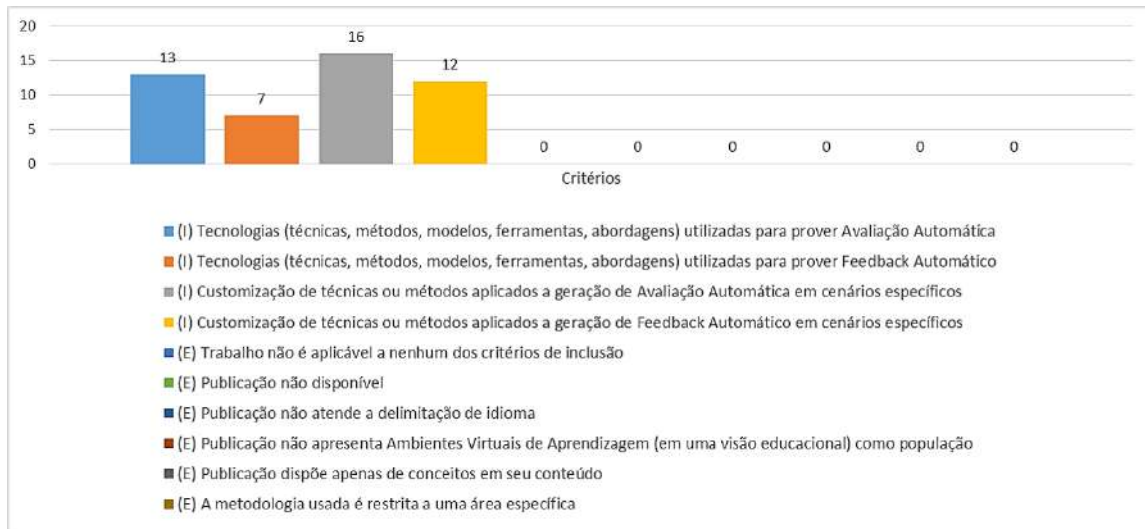


Figura 3.3: Quantificação dos critérios de aceitação em relação aos trabalhos aceitos para extração.

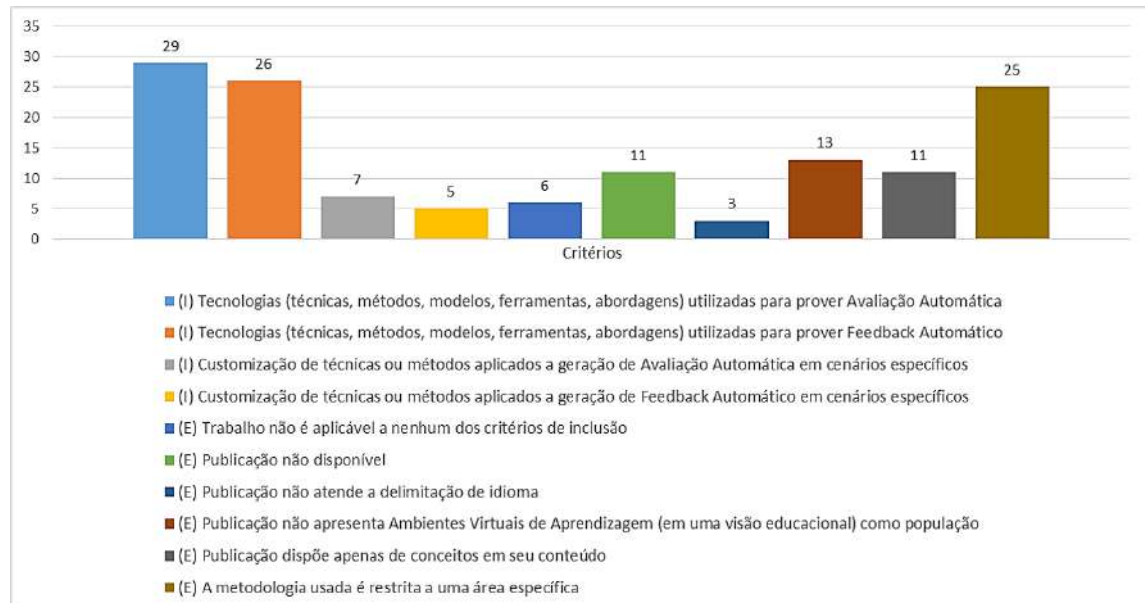


Figura 3.4: Quantificação dos critérios de rejeição em relação aos trabalhos excluídos da extração.

É possível notar que grande parte dos trabalhos rejeitados utilizam uma metodologia restrita a alguma área específica, impossibilitando a utilização dela para outros tipos de estudos. Isto justifica o alto índice dos dois primeiros critérios de aceitação e porque esses trabalhos foram rejeitados.

3.2.1 Coeficiente Kappa

Com o intuito de agregar confiabilidade a pesquisa, o método de coeficiente Kappa foi utilizado, com base na configuração apresentada por Nakagawa *et al.* (2017), para calcular o grau de concordância entre dois pesquisadores sobre a seleção dos trabalhos retornados por esta revisão, comparando resultados e solucionando conflitos por consenso. A concordância pode variar entre

valores abaixo de zero e um. De acordo com Gisev *et al.* (2013), o coeficiente Kappa é um dos mais originais e conhecidos índices de concordância utilizado pela academia.

Esse coeficiente pode variar entre -1 e $+1$, onde $+1$ indica concordância perfeita. A interpretação dos valores dentro desse limite varia de autor para o autor, mas o mais comum é a definição de Landis e Koch (1977), que está na Tabela 3.3.

Tabela 3.3: Interpretação dos níveis de concordância pelo coeficiente Kappa.

Valor	Interpretação
< 0.00	Sem concordância
$0.00 - 0.20$	Concordância pobre
$0.21 - 0.40$	Concordância justa
$0.41 - 0.60$	Concordância moderada
$0.61 - 0.80$	Concordância substancial
$0.81 - 1.00$	Concordância quase perfeita

Foram selecionados aleatoriamente 20% do total de trabalhos (26 trabalhos), sobre os quais os dois filtros foram aplicados por dois pesquisadores separadamente. No primeiro filtro obteve-se uma concordância de 0.76 e no segundo filtro, uma concordância de 0.72. Ambos indicam uma concordância substancial. Com isso, este mapeamento apresenta um índice satisfatório de confiabilidade e precisão.

3.3 Discussão sobre os resultados

Com a fase de extração de informações foi possível coletar todos os dados necessários para responder as questões de pesquisa propostas na Seção 3.1. Analisando as publicações selecionadas identificou-se resultados positivos na avaliação das abordagens propostas, mesmo considerando as respectivas limitações. Com isso, as próximas seções buscam apresentar os dados coletados que respondem as questões de pesquisa propostas.

3.3.1 QP: Quais conceitos, métodos, metodologias, técnicas e ferramentas relacionados a avaliação e geração de *feedback* automático podem ser encontrados na literatura dentro do contexto de ambientes virtuais de aprendizagem?

O gráfico da Figura 3.5 mostra todos os métodos identificados nos trabalhos retornados na revisão. Como pode ser visto, 11 métodos foram identificados ao todo, sendo que em alguns casos mais de um método é usado como proposta do trabalho.

É notável que as técnicas mais aplicadas atualmente para a finalidade da questão principal de pesquisa são as metodologias que utilizam grafos, métricas de similaridade e aprendizado de máquina. No entanto, a revisão mostrou que há evidências da aplicação de métodos formais tanto como meio de avaliação quanto geração automática de *feedback*.

As seções seguintes buscam responder cada uma das sub-questões de pesquisa relacionando os métodos apresentados aqui com os devidos trabalhos de origem.

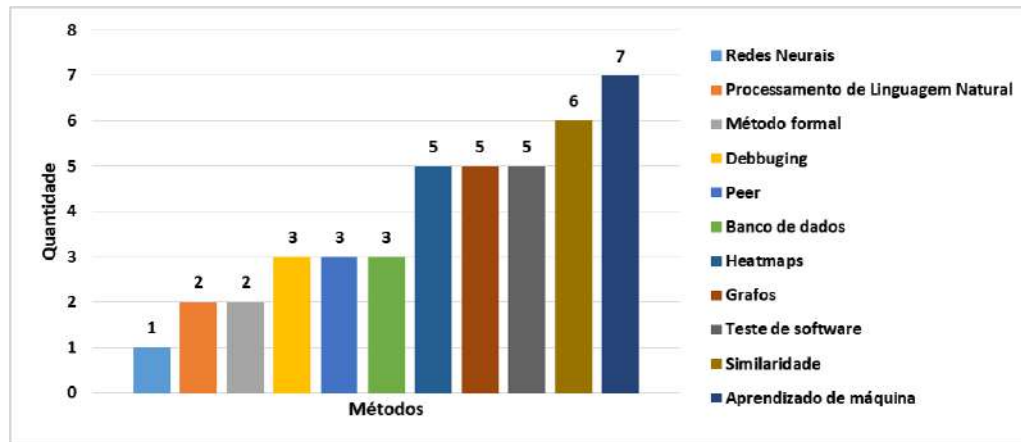


Figura 3.5: Métodos encontrados durante a revisão em função da quantidade de trabalhos presentes.

3.3.2 Q1: Quais os métodos aplicados para produzir a avaliação automática de alunos em ambientes virtuais de aprendizagem?

Conforme explicado anteriormente, parte dos métodos identificados que são usados no contexto da avaliação automática é trabalhado em conjunto com alguma outra técnica secundária. Dessa forma, há uma customização das metodologias abordadas pelos autores. A Tabela 3.4 mostra as aplicações desses métodos em função dos autores de cada um dos 25 trabalhos finais da revisão.

É possível observar que os três métodos mais buscados para esse contexto, conforme demonstrado na Figura 3.5, são acompanhados por aplicações conjuntas de outros métodos. Também é possível perceber que métodos que não possuem inferência de outros, como no caso dos métodos formais, dificilmente aparecem com muita frequência. O gráfico da Figura 3.6 mostra a quantidade de trabalhos classificados por método e suas aplicações de acordo com o ano.

Como pode ser visto na Figura 3.6, não são todos os métodos identificados na QP principal que são aplicados a avaliação automática. Nesse caso, somente 10 métodos puderam ser identificados como meios de fornecer avaliação automática. Dando destaque para aprendizado de máquina e cálculos de similaridade, seguidos de grafos e teste de software. Os métodos pouco identificados nas abordagens foram redes neurais e processamento de linguagem natural. No entanto, são as mais recentes, o que nos leva a refletir que ambas podem estar no advento de suas aplicações a esse contexto.

3.3.3 Q2: Quais técnicas são capazes de gerar *feedbacks* automáticos baseados nos métodos usados como avaliação automática, satisfazendo critérios previamente definidos?

As abordagens identificadas para geração automática de *feedback* são mais discretas que as utilizadas para avaliação. Parte dos trabalhos mostram ser limitados a meios mais convencionais de apresentar informações resultantes de uma avaliação. A Tabela 3.5 demonstra essas abordagens em função dos autores.

É possível notar que alguns trabalhos estão marcados pela cor vermelha. Esses trabalhos utilizam a abordagem de geração de relatório, contendo informações pré-definidas e que não são acrescentadas

Tabela 3.4: Métodos aplicados para avaliação automática em função dos autores.

Autor(es)	Redes Neurais	Processamento de Linguagem Natural	Debugging	Gráficos	Teste de software	Similaridade	Aprendizado de máquina	Peer	Métodos formais	Banco de dados
Edmison <i>et al.</i> (2017)			x							
Shehab <i>et al.</i> (2017)	x									
Correia <i>et al.</i> (2017)						x				
Inventado <i>et al.</i> (2017)										x
Bolón-Canedo <i>et al.</i> (2017)							x	x		
Király <i>et al.</i> (2017)					x					
Reischmann e Kuchen (2016)				x		x				
Thaler <i>et al.</i> (2016)						x				
Hsiao e Govindarajan (2016)		x								
Pape <i>et al.</i> (2016)					x					
Lan <i>et al.</i> (2015)							x			
Waters <i>et al.</i> (2015)							x	x		
Raman e Joachims (2015)							x	x		
Basitere e Ivala (2015)										x
Aziz <i>et al.</i> (2015)			x							
Kyrilov e Noelle (2014)							x			
Striewe e Goedicke (2014)				x						
Alur <i>et al.</i> (2013)						x				
Smith <i>et al.</i> (2013)						x				
Vujosevic-Janivic <i>et al.</i> (2013)			x	x	x					
Fast <i>et al.</i> (2013)									x	
Shen <i>et al.</i> (2012)				x					x	
Gamalel-Din (2012)							x			
AlShamsi e Elnagar (2011)				x	x					

de qualquer outro método para prover *feedback*. Nesses casos, a informação recebida após a etapa de avaliação é fruto de processos imutáveis contidos em ambientes virtuais de aprendizagem.

Em contrapartida, os outros métodos seguem da mesma forma que os métodos identificados para avaliação. Suas aplicações são determinadas pelo uso conjunto de mais de um método. Dentre esses trabalhos, existem abordagens que propõem a customização dos modelos convencionais de geração de relatório com outras abordagens e propostas de geração de *feedback* mais visual e formal do que textos.

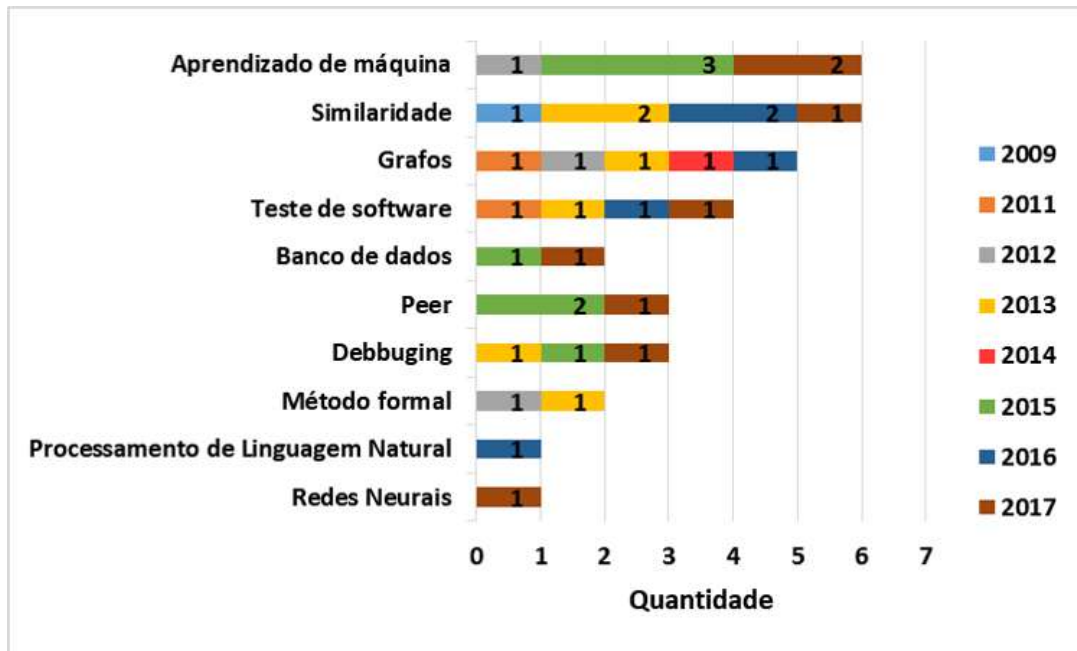


Figura 3.6: Quantidade dos métodos identificados aplicados ao contexto de avaliações automáticas distribuídos por ano.

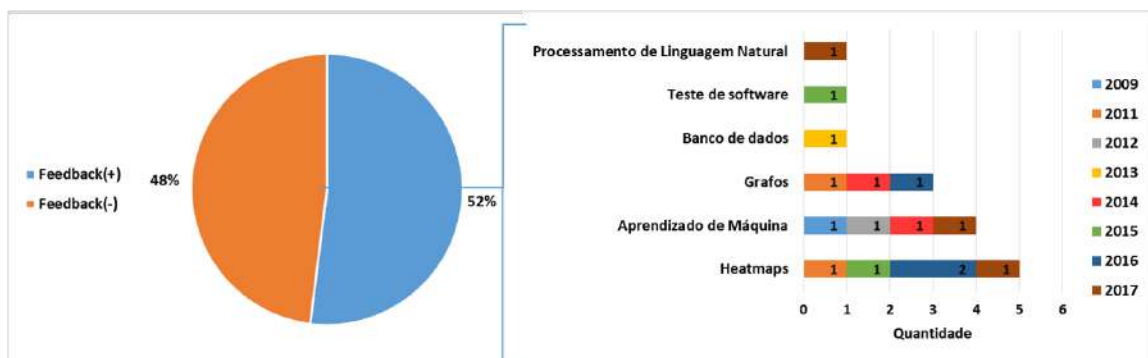


Figura 3.7: Quantidade dos métodos identificados aplicados ao contexto de avaliações automáticas distribuídos por ano.

Como pode ser visto na Figura 3.7, dos 25 trabalhos aceitos, 48% identificados como “*Feedback(-)*” são trabalhos que não abordam uma metodologia de geração automática de *feedback*, esses trabalhos apenas apresentam o resultado de suas avaliações para que seja trabalhado pelo responsável competente.

Os outros 52% compõem o grupo dos trabalhos que utilizam um método para gerar *feedback* automático com base na avaliação (distribuídos por ano). Dentro desse grupo foi possível distinguir 6 dos 11 métodos identificados para a QP principal. Dá-se destaque para a técnica de *heatmaps*, seguida de aprendizado de máquina e grafos.

Tabela 3.5: Métodos aplicados para geração automática de *feedback* em função dos autores.

Autor(es)	Processamento de Linguagem Natural	Aprendizado de Máquina	Grafos	Teste de software	Banco de dados	Heatmaps	Feedback
Edmison <i>et al.</i> (2017)						X	X
Shehab <i>et al.</i> (2017)	X						X
Correia <i>et al.</i> (2017)							X
Inventado <i>et al.</i> (2017)							X
Bolón-Canedo <i>et al.</i> (2017)		X					X
Király <i>et al.</i> (2017)							X
Reischmann e Kuchen (2016)			X			X	X
Thaler <i>et al.</i> (2016)							X
Hsiao e Govindarajan (2016)							X
Pape <i>et al.</i> (2016)						X	X
Lan <i>et al.</i> (2015)						X	X
Waters <i>et al.</i> (2015)							X
Raman e Joachims (2015)							X
Basitere e Ivala (2015)							X
Aziz <i>et al.</i> (2015)				X			X
Kyrilov e Noelle (2014)		X					X
Striewe e Goedicke (2014)			X				X
Alur <i>et al.</i> (2013)							X
Smith <i>et al.</i> (2013)							X
Vujosevic-Janicic <i>et al.</i> (2013)							X
Fast <i>et al.</i> (2013)					X		X
Shen <i>et al.</i> (2012)							X
Gamalel-Din (2012)		X					X
AlShamsi e Elnagar (2011)			X			X	X

3.3.4 Q3: Em que cenários de ambientes de virtuais de aprendizagem esses conceitos são aplicados na literatura?

Ao analisar as publicações aceitas, e após identificar quais os métodos utilizados para avaliar o desempenho de estudantes e gerar automaticamente *feedback*, foi possível definir quais os cenários de aplicação presentes na literatura.

Como pode ser observado, os quatro cenários identificados foram: (i) Programação; (ii) Modelagem; (iii) Textos e (iv) Outros. Onde:

Tabela 3.6: Classificação dos trabalhos em função do cenário de aplicação dos métodos.

Autor(es)	Cenário
Király <i>et al.</i> (2017)	Programação
Edmison <i>et al.</i> (2017)	
Hsiao e Govindarajan (2016)	
Pape <i>et al.</i> (2016)	
Aziz <i>et al.</i> (2015)	
Kyrilov e Noelle (2014)	
Striewe e Goedicke (2014)	
Vujosevic-Janicic <i>et al.</i> (2013)	
Gamalel-Din (2012)	
AlShamsi e Elnagar (2011)	
Correia <i>et al.</i> (2017)	Modelagem
Reischmann e Kuchen (2016)	
Thaler <i>et al.</i> (2016)	
Alur <i>et al.</i> (2013)	
Smith <i>et al.</i> (2013)	
Bolón-Canedo <i>et al.</i> (2017)	Texto
Shehab <i>et al.</i> (2017)	
Basitere e Ivala (2015)	
Inventado <i>et al.</i> (2017)	Outros
Lan <i>et al.</i> (2015)	
Waters <i>et al.</i> (2015)	
Raman e Joachims (2015)	
Fast <i>et al.</i> (2013)	
Shen <i>et al.</i> (2012)	

- Programação – refere-se aos cenários em que o processo de avaliação e *feedback* automático é realizado sobre atividades que envolvem produção de códigos-fontes, programação, teste e áreas afins;
- Modelagem – trata dos cenários em que o processo de avaliação e *feedback* automático é realizado sobre atividades de cunho criativo. Essas atividades envolvem esquematização, projeto visual, descrições técnicas, entre outras;
- Texto – está relacionado aos cenários cuja a avaliação e o *feedback* automático são aplicados sobre atividades de cunho dissertativo, ou seja, atividades cuja a solução será algum tipo de texto a ser verificado;
- Outros – caracteriza os cenários de propósitos gerais. Nesse caso, o processo de avaliação e *feedback* automático é feito com base em atividades cuja taxonomia de execução é definida por questionários de múltipla-escolha.

3.4 Tipos de estudos

A classificação dos 11 métodos em função do seu tipo de estudo, conforme visto na Seção 3.1 e apresentado pela Figura 3.8, corrobora que a maior parte dos trabalhos trata de estudos

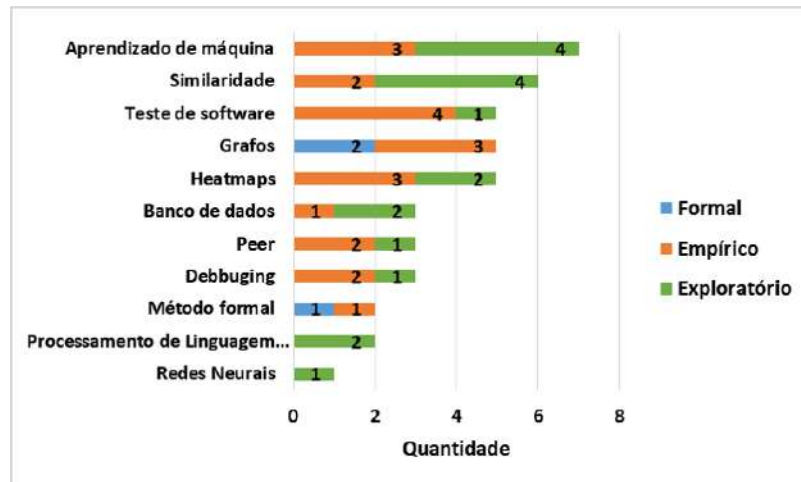


Figura 3.8: Distribuição dos métodos por tipo de estudo.

exploratórios, ou seja, não provam uma teoria mas apresentam argumentações qualitativas de seus resultados por meio de estudos de caso.

Em seguida, os estudos empíricos caracterizam a segunda maior quantidade de aplicações. Esses estudos apresentam uma nova abordagem e comparam-na com outras presentes na literatura, além de apresentarem testes estatísticos como evidências de seus resultados.

Já os estudos formais constituem a menor parcela dos métodos. O que é justificável uma vez que é preciso a elaboração de uma teoria e de uma prova formal de que tal teoria é correta (o que não é trivial), comprovando seus resultados.

Enfatiza-se com as observações acima que os estudos exploratórios são mais comuns devido aos métodos utilizados arbitrariamente exigirem testes estatísticos que comprovem seus argumentos ao contrário das pesquisas empíricas e formais que requerem uma quantificação de seus resultados.

3.5 Limitações desta revisão sistemática

Um problema recorrente em revisões sistemáticas da literatura é encontrar todas as publicações relevantes para um determinado assunto (Kitchenham *et al.*, 2009). Para aliviar esse problema optou-se por uma busca automatizada usando uma biblioteca digital para encontrar as publicações. Ainda assim, ao analisar a execução desta revisão sistemática foram identificadas duas limitações.

Primeiro, existe a possibilidade de alguma publicação relevante não ter retornado durante a revisão. Em vista disso, optou-se por utilizar a biblioteca digital *Scopus* por esta ser uma das maiores bases de dados de publicações, incluindo editoras internacionais amplamente reconhecidas (como *Springer*³, *IEEE Xplore Digital Library*⁴ e *ACM Digital Library*⁵), Kitchenham e Brereton (2013) adotaram e recomendam a biblioteca *Scopus*.

Segundo, por mais que apenas uma pessoa tenha extraído os dados de todas as publicações e isto possa ter introduzido viés, outros pesquisadores somaram com sutis contribuições ao revisar os

³<http://www.springer.com/br/>

⁴ieeexplore.ieee.org/

⁵[https://www.acm.org/](http://www.acm.org/)

dados reunidos. Adicionalmente, o índice de concordância Kappa foi aplicado para mensurar o grau de concordância sobre o MSL.

3.6 Considerações finais

Neste capítulo foi apresentada uma revisão da literatura que visou identificar quais métodos podem ser utilizados como solução do processo de automatização de avaliação e geração de *feedback*. A revisão foi realizada com base em dois filtros de seleção e uma etapa de extração de dados. São apresentadas as informações de execução da revisão, análises e publicações de resultados, discussões sobre os resultados e as limitações da revisão.

A execução da revisão utilizou a biblioteca digital *Scopus* como fonte para busca de publicações, o que resultou em 130 trabalhos retornados pela consulta, vale ressaltar que, com base nestes trabalhos, a explosão de pesquisas na área se deu em meados de 2010, sendo que o maior índice de publicações são datadas de 2017.

Dentre os 11 métodos identificados nos trabalhos analisados destacam-se os métodos de aprendizado de máquina, similaridade, grafos e teste de software para o contexto de avaliações automáticas. Ainda assim, os métodos de processamento de linguagem natural e redes neurais mostraram ser pesquisas mais recentes.

Já relacionado ao *feedback* gerado, o destaque se dá para a técnica conhecida como *heatmaps*, seguida de aprendizado de máquina e grafos. Vale ressaltar que uma parte dos trabalhos não apresentou uma metodologia científica de geração de *feedback*.

Foi identificado que a busca e publicação dentro da área cresça, principalmente se redes neurais estiver de fato em seu despertar nesse contexto. O mapeamento ainda mostrou ser válido por meio da aplicação do coeficiente kappa que obteve um índice substancial de confiabilidade em ambos os dois filtros de seleção aplicados. Contribui-se para a sociedade acadêmica como fonte de busca de trabalhos bases de aplicações de cada um dos métodos identificados.

Ainda na fase de extração foram identificados quatro cenários de aplicação dos métodos de avaliação e geração de *feedback* automático: (i) programação, (ii) modelagem, (iii) textos e (iv) outros.

Por fim, são apresentadas duas limitações presentes durante a execução e finalização da revisão da literatura. Tais limitações são discutidas e argumentadas sobre suas justificativas.

Capítulo 4

Trabalhos Correlatos

Neste capítulo serão apresentados e discutidos os principais trabalhos existentes na literatura, resultantes da revisão apresentada no Capítulo 3, que se relacionam com os tópicos que formam a base para o desenvolvimento deste trabalho. É dada ênfase em: métodos, metodologias e técnicas que são usadas para avaliação e *feedback* automático, bem como seus cenários de aplicação.

4.1 Métricas de Similaridade

A maioria dos trabalhos que usam métodos baseados em cálculos de similaridade tem cenários de modelagem de diagramas gráficos. Alur *et al.* (2013), por exemplo, usam computação de entropia para avaliar automaticamente a semelhança entre os modelos de autômatos finitos deterministas construídos por alunos e professores. Esse cálculo é realizado transformando os modelos gráficos em linguagens formais (usando uma linguagem conhecida como MOSEL¹) e vice-versa. Os resultados desse trabalho foram positivos no sentido de proporcionar uma avaliação minuciosa do desempenho do estudante, uma vez que trata da análise de modelos e provê como *feedback* contraexemplos que comprovam que a solução do aluno está incorreta. Em relação a este trabalho, essa abordagem mostra desvantagem por limitar-se a um contexto específico, dificultando sua expansão/aplicabilidade para outros campos.

Thaler *et al.* (2016) demonstram uma aplicação mais completa do conceito de similaridade. O trabalho apresenta uma metodologia em que um conjunto de métricas desse tipo é aplicado para avaliar automaticamente os modelos gerados pelo aluno, ao invés de aplicar apenas um procedimento de cálculo de similaridade. Cada uma delas gera um determinado valor e, no final, uma pontuação média é atribuída ao aluno com base em seu desempenho em todas as métricas. Exemplos de métricas usadas por eles são mostradas na Figura 4.1, dentre elas: média do grau de conectores que descrevem o número médio de arcos e saídas de conectores, complexidade do fluxo de controle do modelo, número de estados diferentes após uma divisão do conector (*token split*), entre outras.

Infelizmente, as métricas selecionadas pelos autores limitam a aplicabilidade da proposta ao contexto ao qual foi aplicada. Em comparação a este trabalho, buscamos utilizar métricas que possam ser aplicadas a diferentes contextos.

¹[http://www.maths.ed.ac.uk/hall/Xpress/FICO_Docs/mosel/mosel_lang\).html](http://www.maths.ed.ac.uk/hall/Xpress/FICO_Docs/mosel/mosel_lang).html)

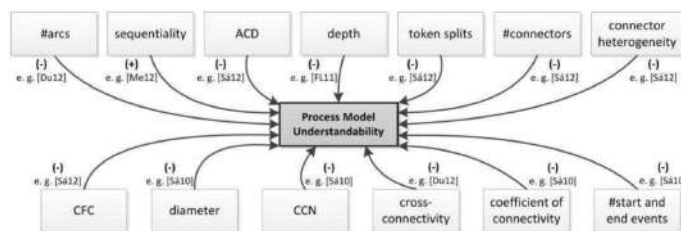


Figura 4.1: Métricas utilizadas no trabalho de Thaler *et al.* (2016).

Outras medidas de similaridade mais complexas podem ser usadas para melhorar o resultado da avaliação. Smith *et al.* (2013) propõem o uso de um cálculo de similaridade, no qual um processo de operações matemáticas é usado para comparar as soluções propostas por alunos e professores. Essas operações sempre buscam a unidade de diferença menos significativa entre os dois objetos comparados, a partir dos quais os créditos dos alunos são atribuídos. Esse processo de similaridade foi aplicado ao cenário de modelagem de diagramas de relacionamento entre entidades e diagramas de classes UML.

Este trabalho, em comparação com os apresentados nessa seção, também busca um meio matemático de avaliar o desempenho dos estudantes. No entanto, buscamos meios que possam representar o pensamento do aluno e que possam ser aplicados de maneira mais flexível, sendo expansível para outras propostas.

4.2 Aprendizado de Máquina

Técnicas de aprendizado de máquina foram identificadas em aplicações de cenários específicos. A maioria deles para avaliação de conteúdo ou artefato pedagógico.

Lan *et al.* (2015) discutem um estudo empírico sobre o uso de aprendizado de máquina através de algoritmos Bayesianos como um meio de agrupar estudantes por seu desempenho. Uma abordagem de eficiência comparativa foi apresentada ao aplicar algoritmos genéricos e usar o algoritmo *Naive Bayes* para tais agrupamentos. Essa abordagem permitiu que critérios específicos fossem adotados para avaliar diferentes aspectos dos alunos, agrupando-os por nível de compreensão. Como a proposta visa atender um público alvo da matemática, não apresentou aspectos de flexibilidade em sua descrição, o que dificulta sua expansão para outros cenários.

Os trabalhos de Raman e Joachims (2015) e Waters *et al.* (2015) também aplicam o algoritmo *Naive Bayes*, mas em um contexto mais flexível e generalizado. Em suas abordagens, os autores utilizaram dados de avaliações taxonômicas de questões de múltipla escolha para treinar seus modelos de algoritmos. O diferencial de suas propostas é que elas fazem uso de uma segunda técnica, chamada de *peer*. Com a aplicação do algoritmo de *Naive Bayes*, os alunos são agrupados e recebem uma avaliação prévia, depois essas avaliações são distribuídas pelo método *peer* entre os componentes de mesmo grupo para que possam ser incrementadas manualmente por estes, por fim sendo entregue aos devidos alunos.

Kyrilov e Noelle (2014) descrevem uma abordagem diferente para aprendizado de máquina aplicada a ambientes de programação virtual. Ele usa um método conhecido como raciocínio baseado em casos (CBR - *Case-Based Reasoning*). Essa técnica permite a solução de novos problemas utilizando adaptações de problemas antigos já solucionados. O método foi proposto conforme mostra a Figura 4.2, sempre que uma solução é considerada correta o banco de casos é

atualizado e caso a solução seja incorreta, um *feedback* contendo informações sobre a resposta correta encontrada no banco de casos é gerado, caso o instrutor veja necessidade de intervir nesse *feedback* ele pode alterá-lo.

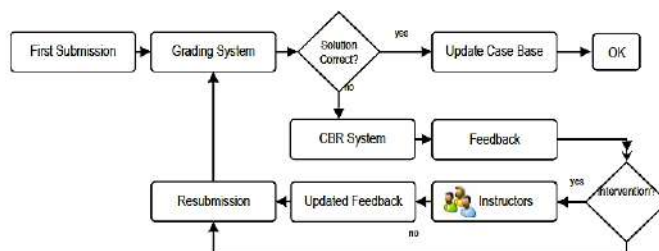


Figura 4.2: Metodologia proposta por Kyrilov e Noelle (2014).

Em geral, esse tipo de metodologia de avaliação busca principalmente melhorar o ambiente de ensino para que o professor tenha *feedback* e possa escolher qual conteúdo pedagógico precisa de mais atenção. Outro fator comum dessas abordagens é a flexibilidade do método que o torna aplicável a qualquer área de conhecimento, mas apenas a um tipo de objeto de aprendizado. No entanto, estes trabalhos apresentaram uma certa dependência de uma base de dados para que suas avaliações sejam feitas. Enquanto que nossa proposta é baseada na modelagem do comportamento do estudantes ao responder ao questionário e essas informações são obtidas diretamente dos *logs* de execução da plataforma educacional, dispensando o trabalho e/ou tratamento com qualquer base de dados.

4.3 Grafos

Um dos métodos identificados mais utilizados no contexto de avaliações automáticas em ambientes virtuais de aprendizagem foi o uso de grafos. Como é um modelo matemático, mas adaptativo, os grafos são usados em vários meios de aplicação. Reischmann e Kuchen (2016), em um estudo empírico, usaram a teoria dos grafos para avaliar a modelagem de questões criativas de linguagem UML usando uma medida conhecida como isomorfismo de subgrafo tolerante a erros (ETSI - *Error-Tolerant Subgraph Isomorphism*).

Outro cenário em que a teoria dos grafos é amplamente utilizada é em ambientes de programação virtual. Striwe e Goedicke (2014), Vujosevic-Janicic *et al.* (2013) e AlShamsi e Elnagar (2011) demonstram este tipo de metodologia usando grafos como um meio para avaliar a execução de código de estudante, seja por meio de fluxo de chamadas de funções, seja por árvores de processo de execução, entre outros meios.

A principal limitação desses métodos é que, mesmo o *feedback* sendo produzido como produto do uso de grafos, transformar o modelo em texto ou mesmo apresentar o modelo em si, é bem comum à técnica e pode não ser entendido corretamente pelos alunos e professores. Este é um dos problemas que esta proposta busca trabalhar. Neste caso, também foi utilizado modelos para gerar avaliações. Entretanto, esta proposta vai transformar as diversas informações em algo compreensível para os estudantes.

4.4 Outros Métodos

Basitere e Ivala (2015) buscaram a recuperação de informações, comumente associadas a bancos de dados, como forma de fornecer avaliações automáticas em ambientes virtuais que utilizam textos como forma de respostas. Nesse método, os autores buscam apresentar um estudo empírico sobre o uso de modelos de recuperação de informação em ambientes virtuais de educação web. Assim como os métodos de aprendizado de máquina, os modelos de recuperação da informação dependem de uma fonte fixa e extensa de dados, o que acaba dificultando sua aplicabilidade.

Shen *et al.* (2012) adotaram conceitos de rede de Petri para realizar avaliação automática em ambientes virtuais que trabalham com questões de múltipla escolha e utilizam a Teoria de Resposta ao Item como critério de avaliação. Este modelo é baseado em fundamentos proposicionais da lógica difusa. Em particular, este modelo é conhecido na literatura como uma rede de Petri difusa. As respostas dos alunos são dadas como entradas no modelo de rede, depois são elaboradas por sua estrutura, levando em conta todo o conceito de teoria de resposta a itens e, finalmente, as notas do aluno são geradas. Uma desvantagem da proposta de Shen é que, por mais que ele utilize um modelo matemático para realizar avaliação, seu foco está em uma só métrica. Nossa proposta é semelhante a dele, mas ao invés de modelarmos métricas, modelamos o comportamento do aluno ao responder o questionário. Assim, várias métricas podem ser aplicadas sobre o mesmo objeto.

Pode-se perceber que há um interesse constante no desenvolvimento de novas metodologias de avaliação e geração de *feedback* automático, independente do cenário. Esse interesse é cada vez mais instigado pela possibilidade de integrar mais de um método para esse fim e adaptação de técnicas existentes na literatura que se mostraram capazes de produzir bons resultados.

4.5 Resumo

Este capítulo demonstrou proposta similares a deste trabalho. Sendo descrito em quatro áreas de maior impacto, com base na revisão apresentada no Capítulo 3.

Acerca dos métodos de similaridade, a principal vantagem é a facilidade de trabalhar com os cálculos e a vasta opção de cálculos de similaridade disponíveis na literatura. Essas propostas mostraram a possibilidade de integração de mais de uma métrica de similaridade para obter um resultado melhor de avaliações. Mas seus cenários de aplicações são limitados a atividades gráficas ou de simples comparações de respostas.

Os métodos que buscam aprendizado de máquina demonstram um grande poder computacional para os mais diversos cenários. No entanto, mostram-se dependentes de uma base de dados fixa e incrementável para que seus resultados sejam mais consolidados.

As propostas que buscam os grafos como meio de avaliação são comuns a várias áreas, por se tratar de uma representação abstrata de problemas, e por terem a capacidade de representar diferentes situações. No entanto, os grafos por si só não possuem uma maneira de servir de avaliação e precisam de algum outro meio que complemente sua aplicação, geralmente cálculos de similaridade. Ainda assim, mostram ser mais adaptáveis a mudanças de paradigmas do que os outros métodos descritos.

Por fim, entram os métodos de recuperação da informação que, similar aos métodos de aprendizado de máquina, necessitam de uma fonte de dados consolidada para que seus processos

sejam aplicados. As redes de Petri mostram uma pequena participação nessa área, mas que mostrou ser útil na modelagem de métricas para avaliação, juntando a formalização e sistematização matemática para prover avaliações e servir de ponte para que recomendações possam ser feitas.

Assim, a proposta deste trabalho busca suprir esta formalização e sistematização dos métodos de avaliação utilizando rede de Petri de forma que seja possível modelar, não métricas, mas sim o comportamento do aluno e aplicar métricas no modelo, independente de base ou da necessidade de um método secundário para avaliação.

Capítulo 5

Método Proposto

Este capítulo descreve o conjunto de métodos propostos para este trabalho, complementando a visão apresentada na Seção 1.5. Os métodos apresentados propõem o uso de técnicas de interpretação e transformação de informação, formalizações matemáticas e algoritmos formais para construção de modelos de redes de Petri, que buscam representar o comportamento de estudantes ao passar pela fase de avaliação automática e geração de *feedback*.

5.1 Visão geral sobre o método

O método proposto é apresentado na Figura 5.1 que mostra o fluxo geral utilizado para compor o método. Nela é apresentado um conjunto de métodos composto por três abordagens que visam: (i) analisar e interpretar as informações de entrada; (ii) sintetizar e executar modelos formais baseados nessas informações; e (iii) apresentar informações relevantes sobre o processo que foi executado, apontando falhas e possíveis melhorias. As caixas enumeradas representam os componentes do método e os itens com linhas tracejadas representam as possibilidades de execução, caso venham a acontecer.

É possível notar que o processo tem início a partir do momento que os *logs* de respostas dos alunos e as métricas selecionadas pelo professor são obtidas do servidor do ambiente educacional. Uma vez que essa tarefa seja concluída, têm início as atividades de análise e interpretação dessas informações pelo componente “Interpretador”. A saída dessa primeira fase são os componentes do modelo, baseados no comportamento do aluno ao responder ao questionário.

A segunda etapa consiste no processo de avaliação automática. Esta fase possui dois componentes principais. O primeiro componente é o “Sintetizador de rede de Petri” que é responsável por receber a estrutura da rede gerada na etapa anterior e construir as matrizes de marcações de cada aluno para serem executadas. O segundo componente, o “Executor da rede”, é responsável por realizar os disparos e ações dentro da estrutura da rede por meio das matrizes, de onde serão extraídos os dados necessários para verificação das métricas sobre o modelo e de onde será gerada a pontuação do aluno.

A terceira etapa só é alcançável se a pontuação gerada pela fase anterior não for máxima, ou seja, se o aluno acertou todas as questões avaliadas esta etapa não é executada, pois ele obteve nota máxima em sua avaliação. Esta etapa é responsável por elaborar o *feedback* que será entregue ao aluno e ao professor. Primeiramente, algumas propriedades do modelo gerado para avaliação serão verificadas, tais propriedades podem variar de acordo com a métrica que busca ser aplicada. Uma vez terminada

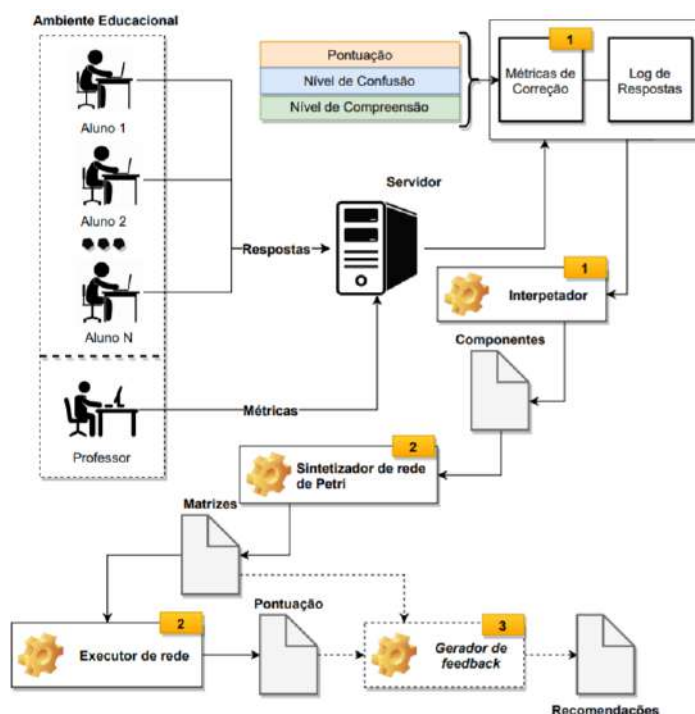


Figura 5.1: Fluxo geral da estrutura do método.

essa verificação, o componente “Gerador de *feedback*” irá transformar as informações geradas em uma recomendação, contendo informações relevantes que podem ser utilizadas para melhorar o desempenho dos estudantes.

As seções seguintes explicam de forma mais detalhada cada uma das etapas descritas e enumeradas presentes na Figura 5.1, sendo elas: etapa de Pré-processamento juntamente com os critérios de correção, etapa de avaliação automática e etapa de geração automática de *feedback*.

5.2 Critérios de avaliação

Segundo Leitão (2017), os critérios de avaliação também podem ser entendidos como métricas que viabilizam a avaliação do impacto de uma aula em uma turma por parte do professor. Biswas e Ghosh (apud (Leitão, 2017)) propõem algumas métricas básicas para medir tanto a interação do estudante com o material didático quanto seu desempenho em questionários avaliativos, são elas: nota tradicional, nota ponderada, nível de dúvida, nível de desvio e nível de confusão.

5.2.1 Nota tradicional

A “Nota tradicional” (NT) calcula a nota obtida por um estudante como o produto da divisão do total de questões corretas ($|c|$) pelo total de questões do questionário ($|q|$), pelo máximo de pontos que podem ser obtidos. Por exemplo, considerando um questionário de seis questões, cuja pontuação total é de 10 pontos, um estudante que acerta seis questões obtém 10 pontos, enquanto um estudante que responde corretamente quatro questões, obtém 6,66 pontos. Esse cálculo é feito pela Equação 5.1:

$$NT = \frac{|c|}{|q|} \times 10 \quad (5.1)$$

5.2.2 Nota ponderada

A "Nota ponderada" (NP) é baseada no peso p de cada resposta marcada pelo aluno em função do máximo de pontos por pesos (mp), supondo uma nota entre 0 e 10. O peso p das respostas varia de 0 a 4 (em inteiros) e $|q|$ é o total de questões do questionário. A "Nota ponderada" pode ser obtida pela Equação 5.2 e 5.3.

$$mw = |q| \times 4 \quad (5.2)$$

$$NP = 10 \times \left(\frac{\sum_{i=1}^n p_i}{mw} \right) \quad (5.3)$$

5.2.3 Nível de desvio

Essa métrica busca mensurar o quão próximo o raciocínio do estudante está do que o professor espera, ou seja, ela mede em níveis o índice de corretude das respostas do aluno. Os níveis variam de 0 a 4, onde:

- Nível 4 (Correto): nível em que a resposta dada pelo estudante condiz com a que o professor espera. Nesse caso, não há qualquer desvio da resposta correta;
- Nível 3 (Próximo ao Correto): nível mais próximo do que o professor espera. É o que mais implica na suposição de que o estudante está retendo algum conhecimento, apresentando um desvio mínimo do correto;
- Nível 2 (Valor Médio): esse nível representa que o aluno está um pouco mais distante do que o professor espera. Nesse caso, sua resposta não está completamente errada, mas também começa a indicar que o aluno está tendo um viés maior;
- Nível 1 (Valor Abaixo da Média): esse nível já indica a situação alarmante de que o aluno está retendo o mínimo de conhecimento sobre determinado conteúdo, mas apresenta um grande viés em relação ao que professor espera;
- Nível 0 (Incorreto): o nível mais baixo é utilizado nos casos em que o estudante errou completamente sua resposta. Nesse caso, representando que o aluno não fazia ideia 'do que' ou 'sobre o que' ele estava respondendo, pois dentre respostas mais corretas, ele optou pela incorreta.

Uma vez que esta métrica visa classificar o percentual da distância entre as respostas corretas e as respostas marcadas pelo aluno em um questionário, ela pode ser classificada como sendo uma nota complementar da "Nota ponderada" (NP), representando o quanto falta da nota ponderada para se chegar a 10, e pode ser obtida pela Equação 5.4.

$$D = (10 - NP) \times 10 \quad (5.4)$$

5.2.4 Nível de dúvida

O nível de dúvida é utilizado para medir o nível de incerteza, ou dúvida, de um aluno ao responder uma questão. De acordo com Leitão (2017), essa incerteza é dada em função da quantidade de vezes que uma alternativa diferente da atual foi marcada, isto, é, quanto mais vezes a resposta da questão tiver sido trocada, mais incerteza demonstrou.

Por exemplo, no mesmo questionário com seis questões, a aluno mudou de alternativa na quarta questão mais de uma vez, sempre alternando entre três possíveis respostas. Ao utilizar como critério o nível de dúvida juntamente com a pontuação do aluno, será possível delinear um limite do que o aluno de fato absorveu do conteúdo e em que aspectos ainda lhe restam dúvidas.

5.2.5 Nível de confusão

O nível de confusão é uma medida de precisão baseada na relação entre o total de respostas corretas ($|c|$) e a quantidade de marcações de respostas (T) em um conjunto de questões (q), ou seja, enquanto o nível de dúvida é medido em relação à uma questão, o nível de confusão é medido em relação a um questionário. Essa relação pode ser vista na Equação 5.5 e 5.6.

$$T = \sum_{i=1}^{|q|} mQ_i \quad (5.5)$$

$$NC = 1 - \frac{|c|}{T} \quad (5.6)$$

5.3 Etapa de pré-processamento

A etapa de pré-processamento compreende o item referenciado como “1” na Figura 5.1, bem como suas entradas e saídas. Esta etapa realiza a tarefa de análise das informações adquiridas do ambiente educacional.

O componente “Interpretador” constrói os componentes da rede ao identificar questões e respostas na saída de cada questionário dentro da plataforma. As Definições 1 (que define o que são os lugares de questão, respostas e verificação na estrutura da rede) e 2 (que define o que é a multiplicidade dos arcos na rede), juntamente do Algoritmo 1 especificam como o modelo de rede de Petri é gerado.

Definição 1. Seja $P_Q \subset P$, onde $p_{q_i} \in P_Q$ é um lugar que representa que a questão i foi respondida; $P_R \subset P$, em que $p_{a_{ij}} \in P_R$ é um lugar que representa uma resposta j para a questão i ; e $P_V \subset P$, onde $p_{v_i} \in P_V$ é um local de verificação que, quando marcado, representa que a resposta correta foi marcada.

Definição 2. Seja c_q a multiplicidade dos arcos de saída de p_{q_i} ; c_a é a multiplicidade dos arcos de saída de $p_{a_{ij}}$; e c_v é a multiplicidade dos arcos de saída de p_{v_i} .

Pelas Definições 1 e 2 juntamente com o Algoritmo 1, é possível notar que sempre que uma pergunta/resposta é identificada, é verificado se ela está ou não no conjunto P . Se a pergunta/resposta já estiver incluída, os contadores serão incrementados. Os arcos são então gerados e seus pesos são baseados nos c_q e c_a de seus respectivos lugares.

```

for each  $row(q, a) \in LOG$  do
  if  $pq_q \notin P$  then
     $P \leftarrow P \cup \{pq_q, pv_q\}$ 
     $c_q \leftarrow 1$ 
     $c_v \leftarrow 0$ 
     $flag_q \leftarrow false$ 
    if  $pa_{qa} \notin P$  then
       $P \leftarrow P \cup \{pa_{qa}\}$ 
       $c_a \leftarrow 1$ 
    else
       $c_a \leftarrow c_a + 1$ 
    end if
  else
     $c_q \leftarrow c_q + 1$ 
  end if
   $T \leftarrow T \cup \{ta_q, to_q\}$ 
   $F \leftarrow F \cup \{(pq_q, ta_q), (ta_q, pa_{qa}), (pq_{qa}, to_q)\}$ 
   $W \leftarrow W \cup \{(pq_q, ta_q) \leftarrow c_q, (ta_q, pa_{qa}) \leftarrow c_q, (pa_{qa}, to_q) \leftarrow c_a\}$ 
  call verification ( $pa_{qa}, pv_q$ )
end for

```

Algoritmo 1: Geração dos componentes da rede com base no *log*.

A primeira etapa da geração de PN é a modelagem do comportamento do aluno ao responder o questionário. Neste momento, não estamos levando em conta o tempo gasto em responder a cada pergunta. Como exemplo, considere as seguintes entradas de log: $\langle q1, b \rangle$, $\langle q3, a \rangle$, $\langle q1, c \rangle$, $\langle q2, a \rangle$, $\langle q1, e \rangle$ e $\langle q5, d \rangle$, o que significa que o aluno respondeu a alternativa “ b ” da questão 1, alternativa “ a ” para a pergunta 3, alterou a resposta de a questão 1 da alternativa “ c ”, respondida como alternativa “ a ” da questão 2, mudou novamente a resposta da questão 1 para a alternativa “ e ” e respondeu à alternativa “ d ” da questão 5.

A Figura 5.2 ilustra a aplicação do Algoritmo 1 no exemplo anterior. Sempre que um aluno retorna a uma pergunta que já foi respondida e marca uma alternativa diferente, o interpretador aumenta o valor do arco de transição do lugar da pergunta para o lugar da nova resposta, de forma que a marcação nunca siga o caminho anterior, levando à execução para a escolha mais recente.

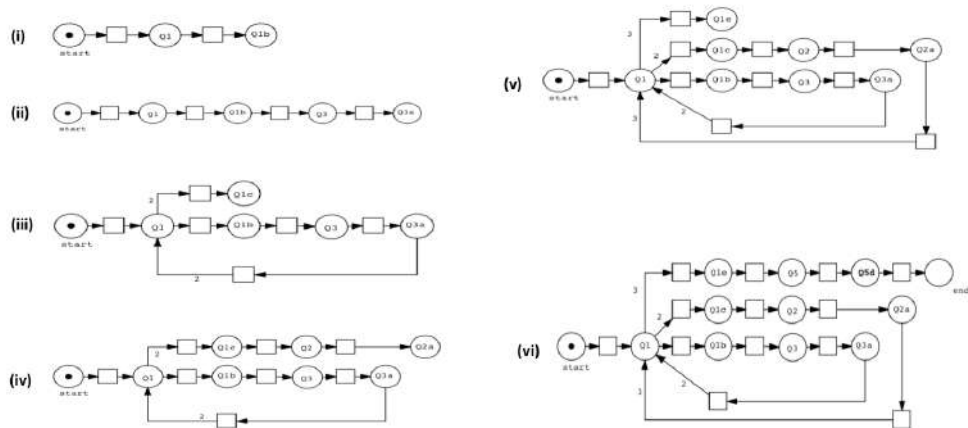


Figura 5.2: Estrutura gerada pelo Interpretador.

```

verification( $p_a, v_q$ ) :
if  $p_a \in LA$  then
   $c_v \leftarrow c_v + 1$ 
   $flag_q \leftarrow true$ 
   $F \leftarrow F \cup \{(ta_q, v_q)\}$ 
   $W \leftarrow W \cup \{(ta_q, v_q) \leftarrow c_v\}$ 
else
  if  $flag_q$  then
     $flag_q \leftarrow false$ 
     $F \leftarrow F \cup \{(v_q, to_q)\}$ 
     $W \leftarrow W \cup \{(v_q, to_q) \leftarrow c_v\}$ 
  end if
end if

```

Algoritmo 2: Criação dos lugares de verificação e seus arcos.

A função $verification(p_a, p_{v_q})$ é detalhada no Algoritmo 2; essa função demonstra a conexão entre os lugares de pergunta, resposta e verificação. Seja LA uma lista das respostas corretas e $flag_q$ um atributo para verificar se p_{v_q} foi a última resposta correta do aluno da pergunta q . Nesse caso, um novo arco é criado a partir da transição t_{a_q} para o local p_{v_q} para representar o fato de que a resposta da pergunta q está correta.

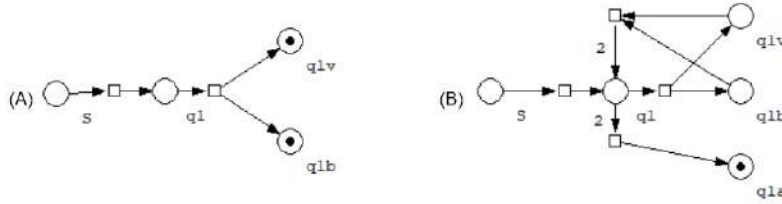


Figura 5.3: Estrutura dos lugares de verificação.

A Figura 5.3 (a) ilustra como modelar quando a resposta da pergunta $\langle q1 \rangle$ está correta, ou seja, o lugar $\langle q1v \rangle$ possui uma marcação. Por outro lado, a Figura 5.3 (b) demonstra que a questão $\langle q1 \rangle$ teve sua resposta alterada de correta (b) para incorreta (a), ou seja, lugar $\langle q1v \rangle$ não tem marcação de tal forma que o local de verificação $\langle q1v \rangle$ esteja inacessível, implicando que os alunos falharam nessas questões.

Este procedimento é repetido para cada linha de *log*. Como mostramos, para cada questão, ela pode gerar transições com multiplicidade de uma ou mais marcações, dependendo do estudante ter mudado de opinião nessa questão. Cada componente da rede de Petri possui atributos, descritos na Figura 5.4, que armazenam informações utilizadas em etapas posteriores, tais como: índices de dificuldade, tópico relacionado à questão, desvio de resposta, entre outros.

Ao final do processo, serão gerados os componentes da estrutura da rede de Petri de cada aluno, ou seja, os objetos gerados das informações retiradas dos *logs* se transformam em representações do comportamento do aluno ao realizar o questionário, garantindo a individualidade durante a avaliação e geração de *feedback*.

Uma vez que todas as informações necessárias são obtidas e transformadas na estrutura da rede, tem início a próxima etapa do método, contendo dois componentes: o “Sintetizador de rede de Petri” e o “Executor da rede”.

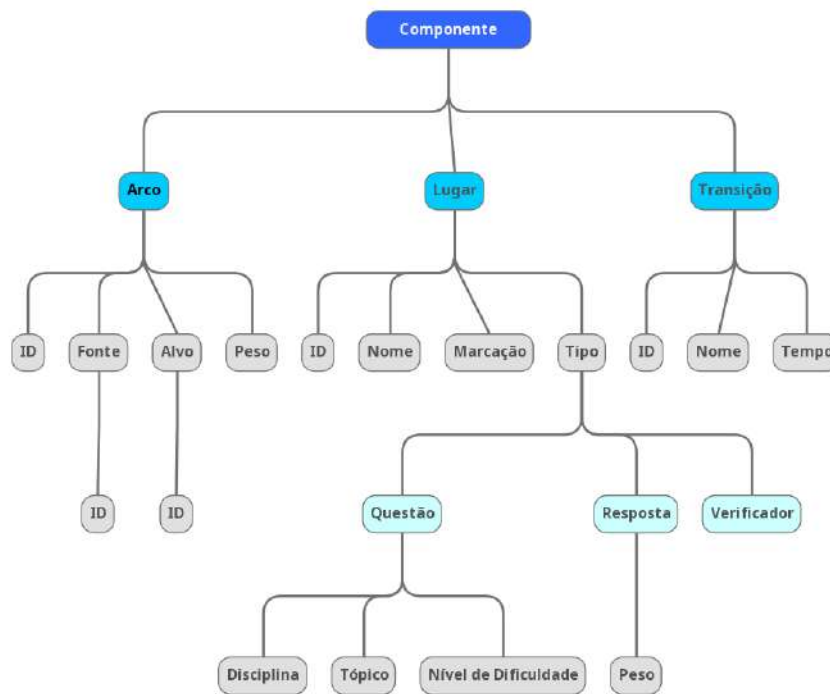


Figura 5.4: Objetos e atributos preenchidos pelo “Interpretador”.

5.4 Etapa de avaliação automática

A fase de Avaliação Automática consiste em uma análise formal realizada sobre o comportamento do aluno modelado em rede de Petri e possui dois componentes principais: o **Sintetizador** e o **Executor** do rede de Petri.

5.4.1 Sintetizador de rede de Petri

O Sintetizador adiciona dois lugares ao modelo, *start* e *end*, conecta os dois lugares a rede e, para fins de implementação, constrói as matrizes I (entrada) e O (saída), ambas de dimensões $m \times n$, onde m é a cardinalidade do conjunto de lugares; e n é a cardinalidade do conjunto de transições. A matriz I representa as pré-condições para disparar cada transição, enquanto a matriz O representa as pós-condições depois de disparar cada transição. As pré-condições representam os requisitos para o disparo de transição, enquanto as pós-condições representam o estado da rede após o disparo.

A Definição 3 indica como gerar matrizes I e O com base na multiplicidade de arcos de um lugar para uma transição (matriz I) e de uma transição para um lugar (matriz O).

Definição 3. Para gerar a matriz I , basta verificar os lugares de entrada em cada transição, para que o valor inserido na matriz corresponda à multiplicidade do arco, ligando o lugar p_i à transição t_j ($w(i, j)$). Da mesma forma, para gerar a matriz O , basta verificar os lugares de saída em cada transição, e o valor colocado na matriz é a multiplicidade do arco, conectando a transição t_j ao lugar p_i ($w(j, i)$). Se não houver arco, o valor na matriz será definido como zero.

5.4.2 Executor da rede de Petri

Após gerar as matrizes I e O , o **Executor** atua no modelo comportamental do aluno, gerando assim a marcação inicial (M_0), que pode ser vista como vetor de dimensão m , onde apenas o local *start* tem um *token*. A Figura 5.5 (A) ilustra a definição da pré-condição para disparar uma transição ativada, e a Figura 5.5 (B) mostra a definição da pós-condição após disparar uma transição habilitada.

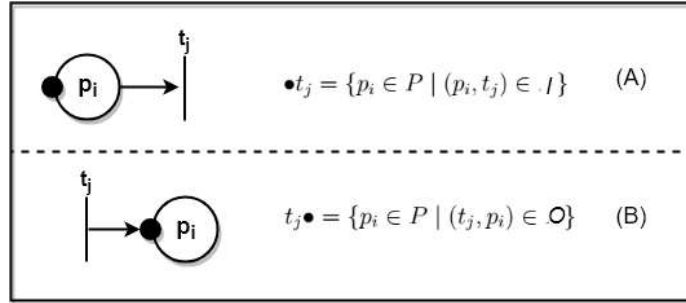


Figura 5.5: Representação das pré e pós condições para disparo de transições.

Uma transição t_j é habilitada em uma marcação m se houver *tokens* suficientes para disparar, ou seja, se $m(p_i) \geq w(p_i, t_j) \forall p_i \in P$. Para essa proposta, uma transição t_j pode ser habilitada se estiver disparável e, adicionalmente, t_j ter a maior prioridade entre todas as transições habilitadas. A transição t_j tem a prioridade mais alta se a Equação 5.7 for verdadeira.

$$w(p_i, t_j) = p_i, \forall p_i \in \bullet t_j \quad (5.7)$$

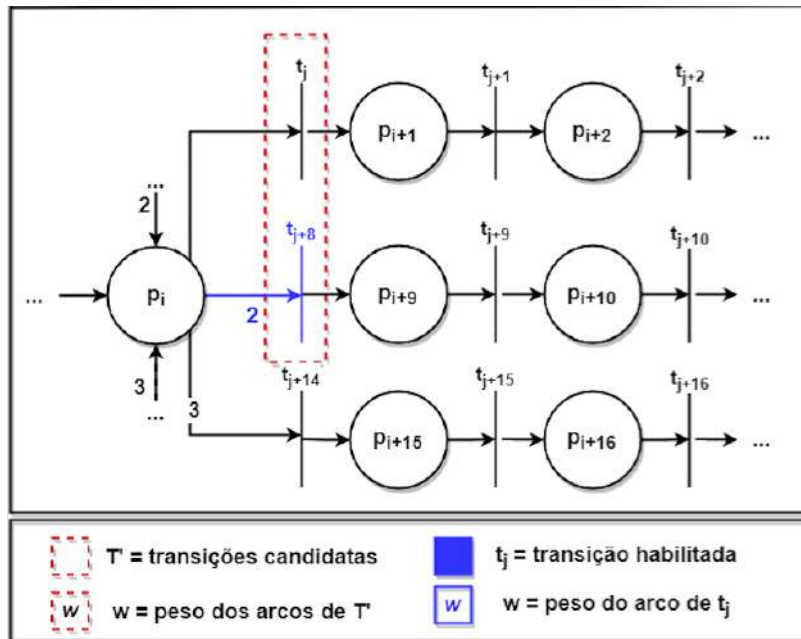


Figura 5.6: Exemplo de prioridades de disparo.

A Figura 5.6 explica a prioridade de disparo de uma transição. O principal objetivo dessa prioridade é lidar com lugares concorrentes, ou seja, não permite que múltiplas transições sejam disparadas concorrentemente na mesma marcação. Essa situação é melhor explicada pelo fato de um aluno não poder responder duas perguntas ao mesmo tempo.

Na Figura 5.6, vamos considerar que no lugar p_i existam dois *tokens*. Embora as transições t_j e t_{j+8} estejam habilitadas, somente a transição t_{j+8} é disparável, pois tem a maior prioridade. No entanto, suponha que o lugar p_i tenha três *tokens*; Neste caso, a única transição que pode ser disparada seria a transição t_{j+14} .

Uma vez identificada a transição disparável, o Executor da rede gera uma nova marcação M_k por meio da Equação 5.8.

$$M_k = M_{k-1} - I_{ij} + O_{ij}, k = 1, 2, \dots \quad (5.8)$$

Definição 4. M_F só terá valores $M_F(p_i) > 0$ para perguntas cuja a resposta está correta e para o lugar final. Como $P_V \subset P$ é um subconjunto de locais de verificação (ver Seção 5.3). Assumindo que $\overline{P_V}$, são todos os lugares $p_{v_i} \in P_V$ cujos *tokens* são iguais a zero, isto é, $p_{v_i} = 0$. Nesse caso, podemos dizer que $\overline{P_V}$ contém as respostas erradas.

Vamos supor que M_F seja a marcação final desejável. A Equação 5.8 é aplicada enquanto $M_k \neq M_F$, ou seja, até que o lugar final seja alcançável. Este lugar final representa que o aluno concluiu todas as suas respostas e, portanto, M_F só pode conter $M_F(p_i) > 0$ nos lugares de verificação e no lugar final. Assim, é possível obter as questões corretas totais do aluno (consulte Definição 4) e avaliar seu desempenho pela métrica ‘Pontuação’ descrita na Seção 5.6.2.

Por exemplo, suponha uma marcação final M_F gerada pela execução da rede de um estudante que respondeu um questionário com cinco questões, conforme segue abaixo:

$$M_{SS} = [1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \dots 0 \ 0 \ 1] \quad (5.9)$$

Nessa marcação, os itens em **negrito** são os lugares de verificação correspondentes as questões. De acordo com a Definição 4, isso indica as questões que o estudante conseguiu acertar, no caso $\langle q1, q3, q5 \rangle$ e o último valor de ‘1’ que indica o lugar final. Os valores em *itálico* indicam os lugares que compõem o restante da rede (questões e respostas), então a dimensão desses valores pode variar muito de aluno para aluno, dependendo de como o estudante respondeu ao questionário.

Dessa forma, a pontuação tradicional do aluno pela métrica da Pontuação (veja Seção 5.6.2) é:

$$Pts = (10/5) * 3 = 6.0 \text{ pontos} \quad (5.10)$$

Para avaliar pela nota ponderada nesse caso, precisamos supor valores de pesos para as respostas dadas pelo aluno. Então suponhamos que, apesar de o aluno ter errado as questões $\langle q2, q4 \rangle$, as respostas que ele marcou tem um peso 3, que não é correto mas chega próximo (peso 4). Nesse caso, a nota tradicional do aluno é:

$$mw = 5 \times 4 = 20 \quad (5.11)$$

$$NP = 10 \times \left(\frac{(4 + 3 + 4 + 3 + 4)}{mw} \right) \quad (5.12)$$

$$NP = 10 \times \left(\frac{18}{20} \right) \quad (5.13)$$

$$NP = 10 \times 0,9 \quad (5.14)$$

$$NP = 9,0 \text{ pontos} \quad (5.15)$$

Nesse ponto já podemos observar a diferença que a mudança da métrica de avaliação impacta na nota do estudante. Para avaliar o nível de desvio do estudante, devemos levar em consideração a nota ponderada. Então tomemos como base o exemplo da Equação 5.15, o nível de desvio do estudante é:

$$D = (10 - 9,0) \times 10 \quad (5.16)$$

$$D = 1,0 \times 10 \quad (5.17)$$

$$D = 10\% \text{ de desvio} \quad (5.18)$$

Se formos fazer uma avaliação individual das questões, podemos usar o nível de dúvida que estudante teve. No entanto, como parte da avaliação individual de uma questão, o nível de dúvida serve mais como apoio para outras métricas. Por exemplo, supondo que na estrutura da rede de Petri o lugar da questão 3 tenha sido marcado 4 vezes, então seu nível de foi 3, já que a primeira marcação foi seu palpite inicial, e ele ainda assim não marcou a alternativa correta. Mas, pela pontuação ponderada, a questão 3 tem um nível de desvio 3, logo não está tão errada. Esse mesmo processo pode ser feito para todas as questões, e levar em consideração outras métricas para fazer uma avaliação das questões.

A última avaliação a ser feita é do nível de confusão do aluno. Nesse caso, tomando a Equação 5.10, tiveram 3 respostas corretas. Uma vez que o nível de dúvida é avaliado em relação ao questionário como um todo, o nível de confusão do estudante é:

$$T = 1 + 1 + 4 + 1 + 1 \quad (5.19)$$

T é a quantidade de marcações de respostas, nesse caso, assume-se que somente a questão 3 foi trocada 4 vezes, logo:

$$T = 8 \quad (5.20)$$

$$NC = 1 - \frac{3}{8}, \text{ onde } 3 \text{ o total de respostas corretas} \quad (5.21)$$

$$NC = 1 - 0,375 \quad (5.22)$$

$$NC = 0,625 \quad (5.23)$$

5.5 Gerador de *feedback*

Além das métricas de ‘Nota Tradicional’ e ‘Nota Ponderada’, outras podem ser obtidas a partir das informações contidas na rede, conforme mostrou a Figura 5.4. Assim, é possível propor recomendações para tópicos, onde o aluno teve uma taxa de erro maior. Por exemplo, na Figura 5.7, assumamos que $p_i = “Q51V”$ seja um lugar de verificação e $p_i \in \overline{PV}$. De acordo com a Definição 4, o aluno não marcou a resposta correta para a questão 51 porque “Q51V” não é alcançável, uma vez que a resposta correta não foi selecionada, não gerando um arco para o local de verificação, o que pode gerar uma recomendação de estudo para o tópico da questão.

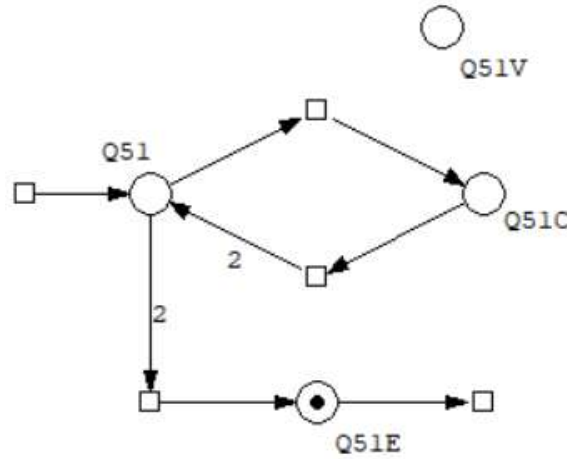


Figura 5.7: Exemplo de $m_i \in \overline{M_{SS}}$.

Outra informação que pode ser obtida com o método é a quantidade de vezes que o aluno mudou de resposta para uma questão. Por exemplo, na Figura 5.7, além do aluno não ter respondido corretamente, ele mudou sua resposta duas vezes, a primeira para “Q51C” e a segunda para “Q51E”, indicando o nível de dúvida do aluno sobre essa questão, que também é uma métrica descrita na Seção 5.2.4. O nível de dúvida é caracterizado dentro da estrutura da rede pela multiplicidade dos arcos de saída dos lugares de questões. Se uma questão tem dois arcos de saída, seu nível de dúvida é dois, se possui três, seu nível de dúvida é três, e assim por diante.

O *feedback* pode ser incrementado analisando o desvio das respostas dadas pelo estudante (métrica descrita na Seção 5.6.4). Por exemplo, assumindo que na questão “Q51” da Figura 5.7 nenhuma das respostas dadas pelo estudante é correta, mas “Q51C” tem um desvio de nível 3 e “Q51E” tem um desvio de nível 4, então supõem-se que o estudante está retendo algum conhecimento sobre o conteúdo dessa questão, já que suas respostas estão próximas do correto.

Os dados gerados pela rede de Petri podem ser transformados em informações de *feedback* visual, ou em relatórios, e apresentadas aos alunos como um meio de recomendar que seus esforços sejam direcionados para conteúdos específicos nos quais a taxa de erros é maior. Caso o *feedback* seja visual, cores podem ser usadas para melhorar a compreensão do modelo, indicando diferentes índices de acordo com a métrica, se um nível de dúvida é alto, ou se o desvio está próximo do correto, e assim por diante. Isso poderá ser melhor compreendido no próximo capítulo.

5.6 Formalização das métricas

Esta seção visa discretizar as métricas apresentadas na Seção X. A discretização delas é feita tomando como base a estrutura da rede Petri utilizada no desenvolvimento do método. O objetivo é confirmar que com o método proposto é possível realizar as ações propostas pelas métricas.

5.6.1 Cenário

Seja um estudante *A* respondendo a um questionário de múltipla-escolha para obter a nota máxima de 10 pontos, contendo 5 questões, com 5 alternativas de respostas (A, B, C, D, E). O gabarito das respostas corretas, como segue:

- 1 - B
- 2 - A
- 3 - C
- 4 - D
- 5 - A

O estudante *A* forneceu como resposta as seguintes alternativas:

- 1 - B
- 2 - C
- 3 - C
- 4 - A
- 5 - E

Dado o modelo de rede de Petri gerado pelo estudante conforme a Figura 5.8 :

Pela Figura 5.8, a marcação inicial (m_0) da rede, que representa o atual estado do estudante antes de responder ao questionário é como segue na Tabela

Para chegar ao estado final da rede de Petri do estudante, é necessário avançar na execução da rede conforme a marcação progride:

Nas Figuras 5.9 e 5.10 é possível observar que a marcação em seu primeiro estado já assume a primeira questão respondida como sendo correta, uma vez que o lugar de verificação $\langle Q1V \rangle$ foi alcançado. A marcação então se divide e uma delas segue para os próximos estados da rede.

Já nas Figuras 5.11 e 5.12 é possível notar que a resposta marcada está incorreta, pois o lugar de verificação $\langle Q2V \rangle$ não foi alcançado pela marcação em seu estado atual e foi passado para o próximo.

As Figuras 5.13, 5.14 e 5.15 mostram que o estudante não respondeu corretamente a questão e seu lugar de verificação não foi alcançado. No entanto, ao invés de seguir para a próxima questão, o

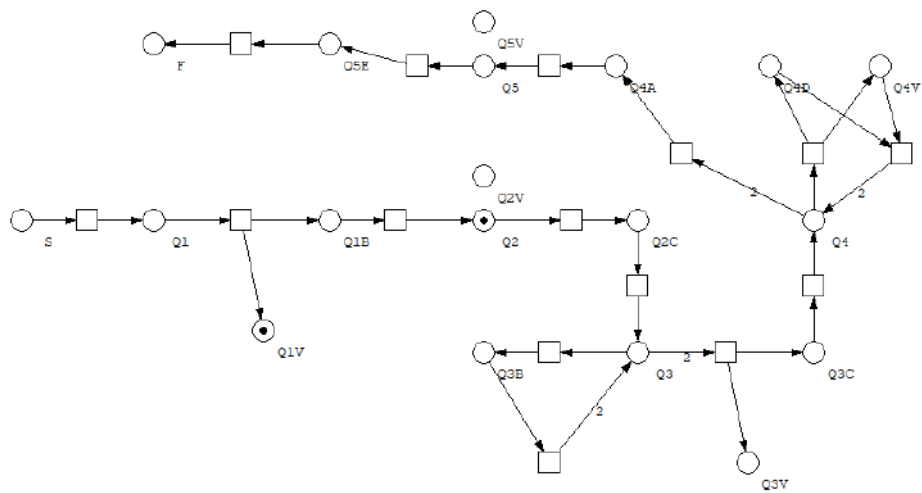


Figura 5.11: Estado m_3 da rede de Petri.

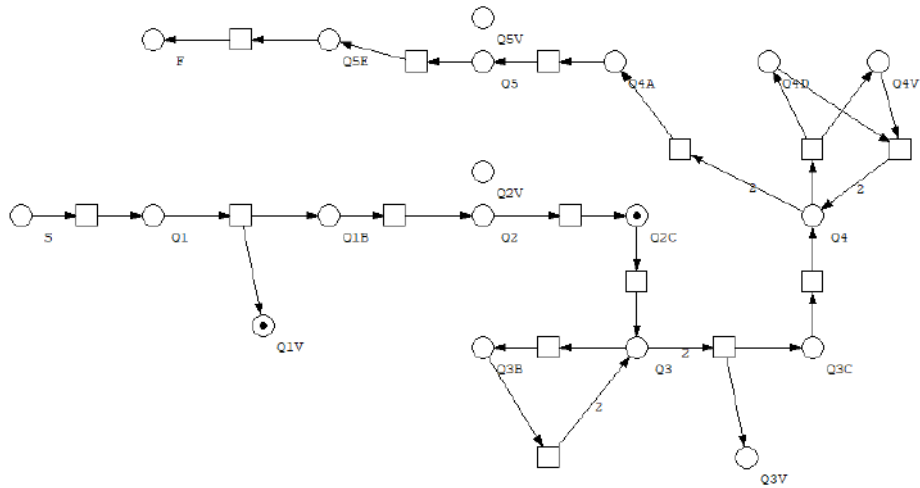


Figura 5.12: Estado m_4 da rede de Petri.

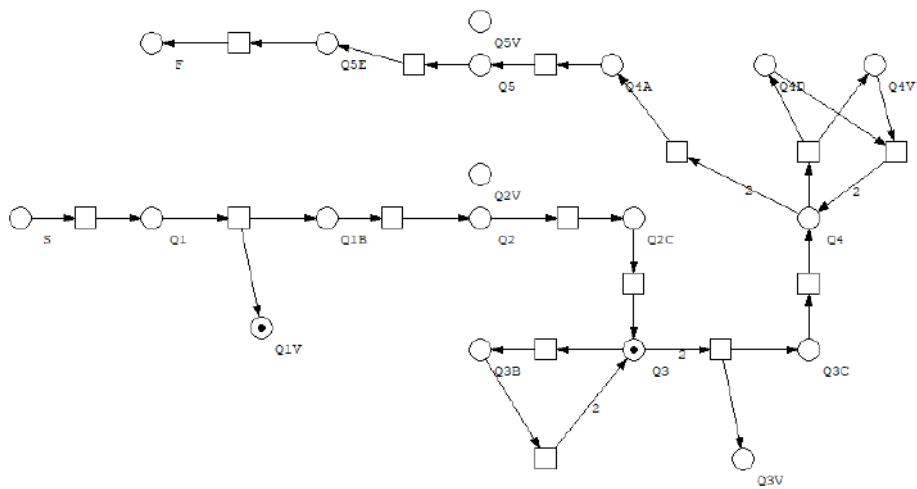
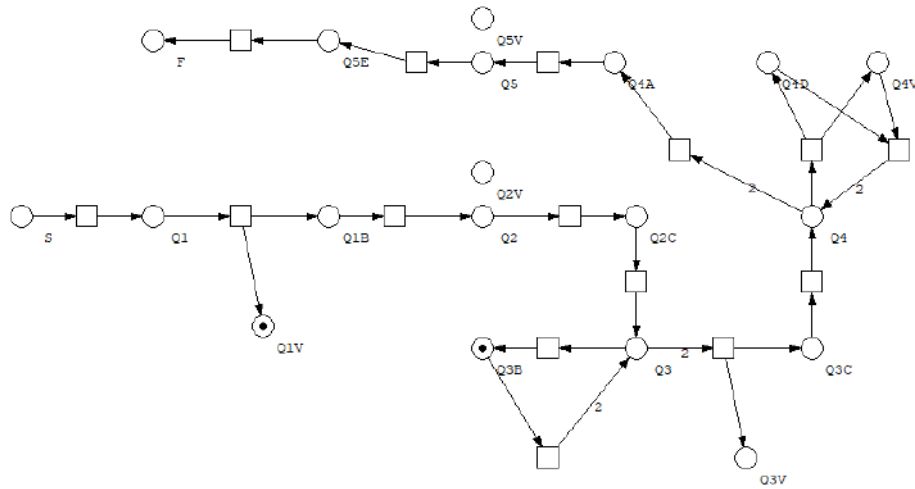
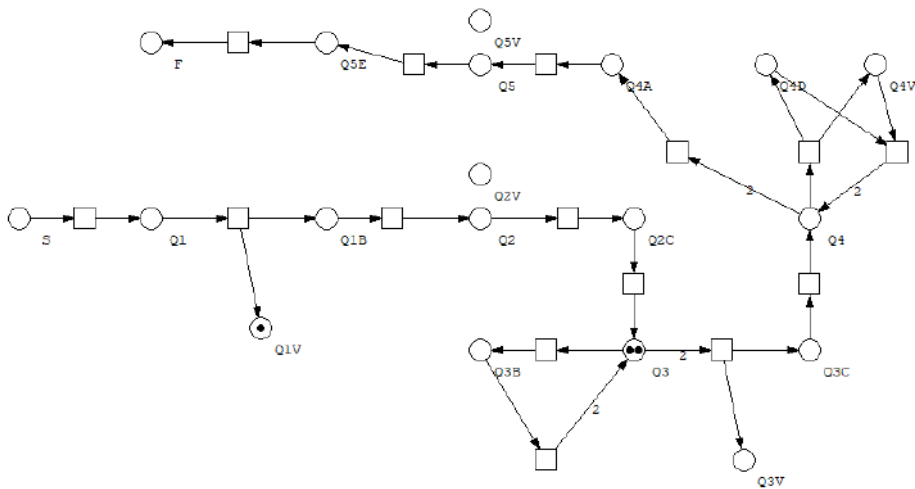


Figura 5.13: Estado m_5 da rede de Petri.

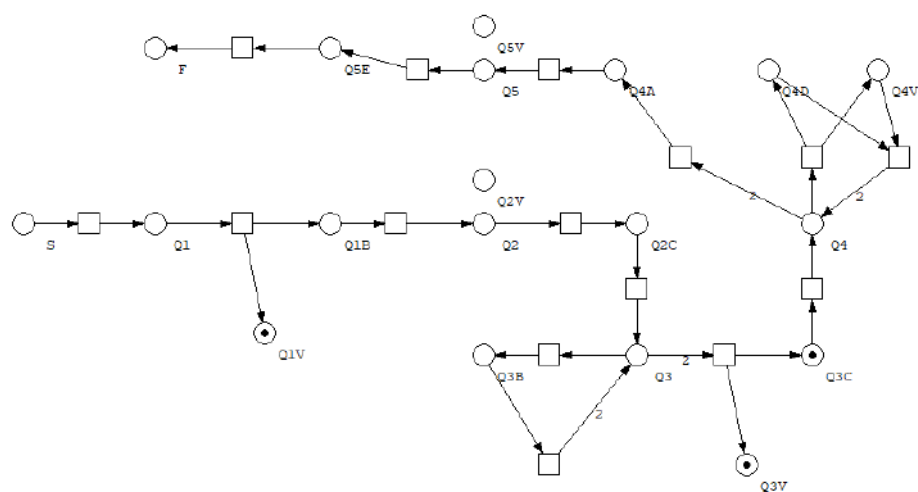
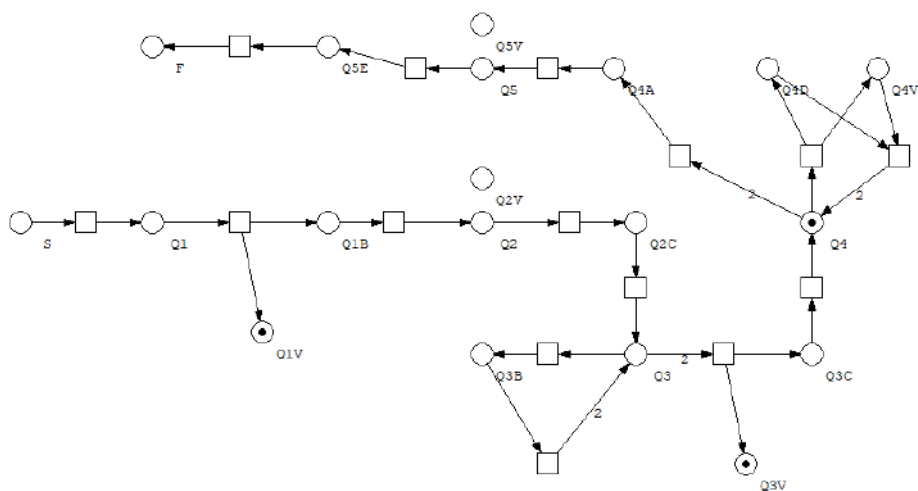
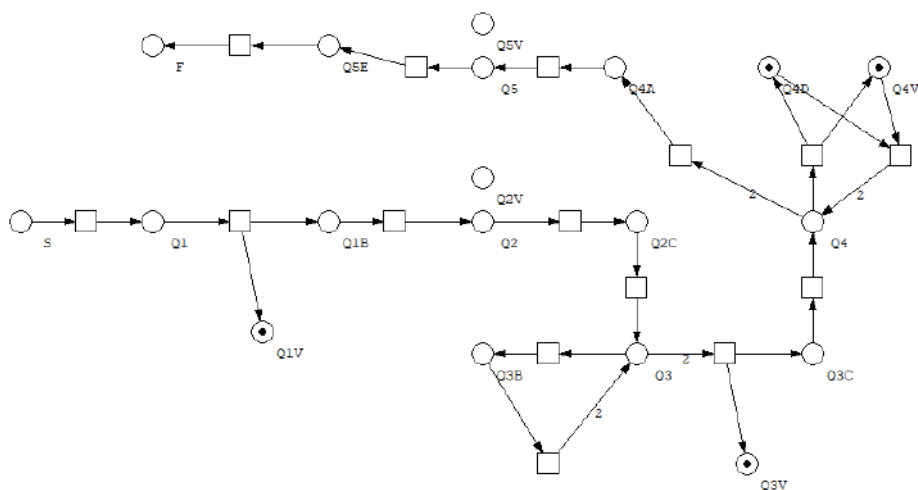
Figura 5.14: Estado m_6 da rede de Petri.Figura 5.15: Estado m_7 da rede de Petri.

estudante volta para a mesma questão e muda sua resposta. Assim, o lugar da questão recebe duas marcações e espera pelo próximo passo.

A Figura 5.16 mostra a segunda resposta marcada pelo estudante. Ele responde corretamente, então uma das marcações, que veio da resposta anterior, alcança o lugar de verificação $\langle Q8V \rangle$ enquanto a outra segue para o próximo estado da rede.

As Figuras 5.17 e 5.18 mostram uma situação contrária a anterior. Nesse caso, o estudante respondeu corretamente a questão e a rede alcança o lugar de verificação $\langle Q4V \rangle$. No entanto, a Figura 5.19 mostra que o aluno voltou para a mesma questão, o que fez com que a marcação que estava no lugar de verificação da questão saia e vá para $\langle Q4 \rangle$. Na Figura 5.20 vemos que mudar sua resposta da correta para uma incorreta ($\langle Q4A \rangle$) faz com que o lugar de verificação da questão não receba mais uma marcação e fique vazio.

As Figuras 5.21 e 5.22 mostram que novamente o estudante marcou a resposta incorreta, deixando o lugar de verificação $\langle Q5V \rangle$ inacessível. Ele não retorna para mudar sua resposta, então a Figura 5.23 mostra que o estudante finalizou seu questionário.

Figura 5.16: Estado m_8 da rede de Petri.Figura 5.17: Estado m_9 da rede de Petri.Figura 5.18: Estado m_{10} da rede de Petri.

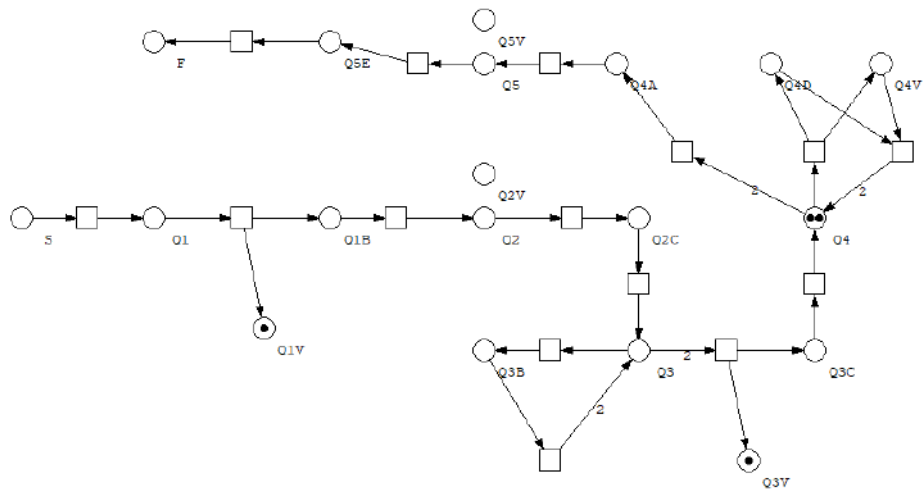


Figura 5.19: Estado m_{11} da rede de Petri.

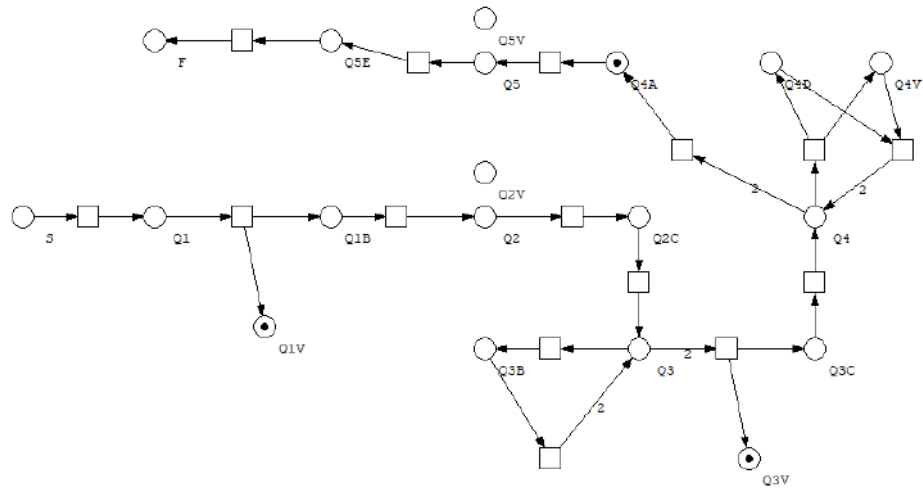


Figura 5.20: Estado m_{12} da rede de Petri.

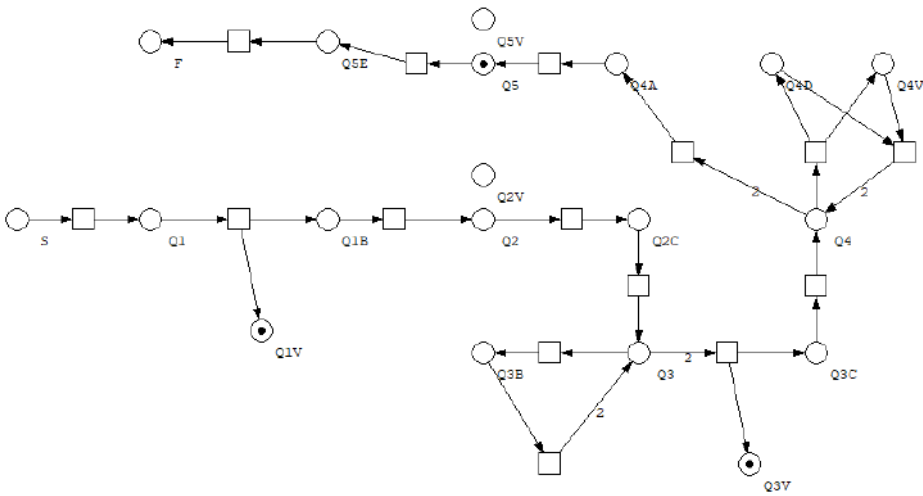


Figura 5.21: Estado m_{13} da rede de Petri.

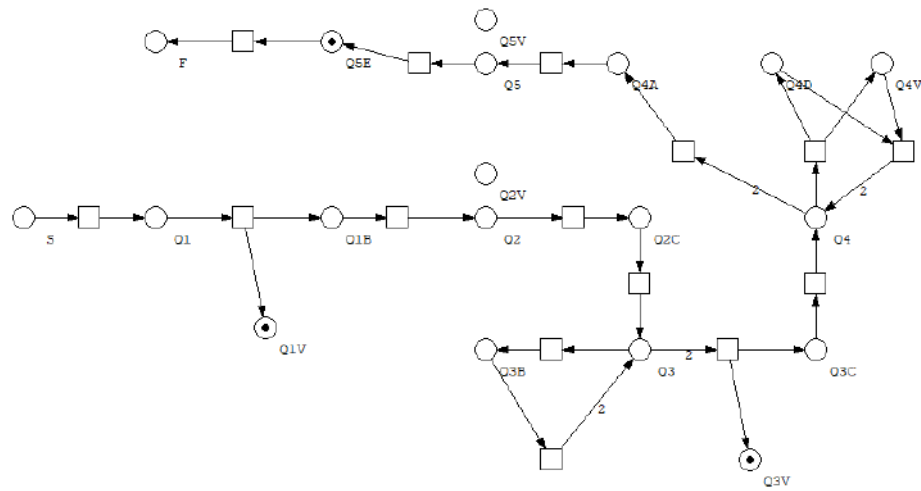


Figura 5.22: Estado m_{14} da rede de Petri.

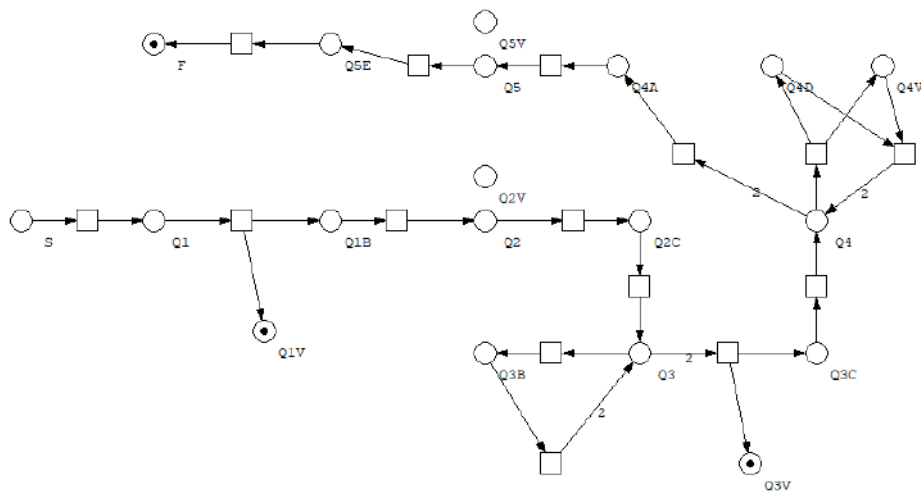


Figura 5.23: Estado m_{15} da rede de Petri.

A Tabela 5.1 mostra em cada um dos m^{15} estados da rede as marcações em seus respectivos lugares. Nela é possível notar como a marcação final da rede é obtida.

Com base no cenário descrito e nas figuras e tabelas que demonstram a execução de uma rede de Petri do início ao fim, tomemos a seguinte definição geral:

Definição 5. Seja $P = [p_1, p_2, p_3, \dots, p_{max}]$ o vetor de todos os lugares da rede de Petri, que representa as respostas de um estudante ao questionário. Dado n o número de questões do questionário, $P_Q = [p_{q1}, p_{q2}, p_{q3}, \dots, p_{qn}]$ é o vetor de lugares de questões e $P_Q \subset P$ com $P_Q \rightarrow \mathbb{N}$. $P_V = [p_{v1}, p_{v2}, p_{v3}, \dots, p_{vn}]$ é o vetor de lugares de verificação da rede de Petri de modo que $P_V \subset P$ com $P_V \rightarrow \{0, 1\}$. Dado m o número total de respostas dadas pelo estudante, $P_R = [p_{r1}, p_{r2}, p_{r3}, \dots, p_{rm}]$ é o vetor de respostas dadas pelo estudante tal que $P_R \subset P$, $P_R \rightarrow \mathbb{N}$ e $P_{QR} = P_Q \times P_R$ sendo a relação dos lugares de questão com lugares de respostas. $P_C = [s, f]$, $P_C \rightarrow 0, 1$ e $P_C \subset P$, sendo P_C o vetor de lugares de controle da rede de Petri, onde s o lugar de começo e f o lugar do fim. Dessa forma, $M_E = [m_V, m_s, m_Q, m_R, m_f]$, onde M_E é o vetor de marcação do estado final da rede de tamanho P após sua execução, tal que $m_V \subset P_V$, $m_s \rightarrow s$ e $m_s \subset P_C$, $m_Q \in P_Q$, $m_R \subset P_R$, $m_f \rightarrow f$ e $m_f \subset P_C$. Pelo cenário, M_E pode ser classificado como a Tabela 5.1.

Com base na Definição 5, as seções seguintes irão demonstrar matematicamente como as cinco métricas da Seção 5.2 podem ser alcançadas por meio da estrutura proposta pela rede de Petri e como seus valores são obtidos.

5.6.2 Nota Tradicional

Conforme a métrica descrita na Seção 5.6.2, a nota tradicional considera $m_V \in M_E$ como sendo uma função sobre o estado final da rede de Petri:

- $M_E(P_{Vq}) = 1$, se $p_{rq} \subset P_R$ é marcado como sendo a última escolha do estudante e é a resposta correta;
- $M_E(P_{Vq}) = 0$, se $p_{rq} \subset P_R$ é marcado como sendo a última escolha do estudante e é a resposta incorreta;

Com isso, C_C é o conjunto das questões corretas de forma que $C_C = \{p_{vq} \in M_E | p_{vq} = 1\}$ e C_I é o conjunto de questões incorretas de forma que $C_I = \{p_{vq} \in M_E | p_{vq} = 0\}$. A nota tradicional pode ser obtida por meio de:

$$NT = \frac{|C_C|}{|n|} \times 10 \quad (5.24)$$

Exemplo: Dado o cenário descrito na Seção 5.6.1, com $M_E = [m_{15}]$:

Tabela 5.2: m_{15} da Figura 5.23, considerando somente os lugares de verificação.

	QV1	QV2	QV3	QV4	QV5
m_{15}	1	0	1	0	0

As 5 primeiras posições de m_{15} são os valores de P_V . Logo, $C_C = \{p_{v1}, p_{v3}\}$, pois $M_E(p_{v1}) = 1$ e $M_E(p_{v3}) = 1$, deixando $C_I = \{p_{v2}, p_{v4}, p_{v5}\}$. Portanto, com $|C_C| = 2$

$$NT = \frac{2}{5} \times 10 \quad (5.25)$$

$$NT = 0,4 \times 10 \quad (5.26)$$

$$NT = 4,0 \text{ pontos} \quad (5.27)$$

5.6.3 Nota Ponderada

Conforme a métrica descrita na Seção 5.6.3, para obter a nota ponderada, é necessário ter um peso D , de modo que $F(p_q, p_r) \rightarrow D \mid p_q \in P_Q, p_r \in P_R$ e $D \in \{0..4\}$. Esse peso é obtido por meio da estrutura externa detalhada pela Figura 5.4 da Seção 5.3. Para a nota ponderada (NP), o máximo de pontos por peso D_{max} pode ser obtido por meio de:

$$D_{max} = |P_Q| \times 4 \quad (5.28)$$

Onde 4 é o peso para questões corretas. Dessa forma, NP pode ser obtida por meio de:

$$NP = 10 \times \left(\frac{\sum_{i=1}^n F(q_i, r_i)}{D_{max}} \right) \quad (5.29)$$

Exemplo: Dado o cenário descrito na Seção 5.6.1, com $P_Q = [p_{q1}, p_{q2}, p_{q3}, p_{q4}, p_{q5}]$, $|P_Q| = 5$. Com isso:

$$D_{max} = 5 \times 4 \quad (5.30)$$

$$D_{max} = 20 \quad (5.31)$$

Os estados de marcações em que o estudante dá suas últimas respostas são os das marcações $m2, m4, m8, m12$ e $m14$, vistas na Tabela 5.1. Supondo que o peso D das respostas dadas pelo estudante sejam conforme descritas a seguir: $F(p_{q1}, p_{r1}) = 4$; $F(p_{q2}, p_{r2}) = 2$; $F(p_{q3}, p_{r3}) = 4$; $F(p_{q4}, p_{r4}) = 0$; $F(p_{q5}, p_{r5}) = 3$, sua NP será:

$$NP = 10 \times \left(\frac{4 + 2 + 4 + 0 + 3}{20} \right) \quad (5.32)$$

$$NP = 10 \times \left(\frac{13}{20} \right) \quad (5.33)$$

$$NP = 10 \times 0,65 \quad (5.34)$$

$$NP = 6,5 \text{ pontos} \quad (5.35)$$

5.6.4 Nível de Desvio

Seja NP a nota ponderada, precisa-se de um peso D , de modo que $F(p_q, p_r) \rightarrow D \mid p_q \in P_Q, p_r \in P_R$ e $D \in \{0..4\}$. Esse peso é obtido por meio de uma estrutura externa, demonstrada na Figura 5.4. Conforme NP e D , o nível de desvio ND de um estudante pode ser obtido por meio de:

$$ND = (10 - NP) \times 10 \quad (5.36)$$

Exemplo: Seja o cenário descrito na Seção 5.6.1 e $NP = 6,5$, o nível de desvio ND de um estudante é de:

$$ND = (10 - 6, 5) \times 10 \quad (5.37)$$

$$ND = 4, 5 \times 10 \quad (5.38)$$

$$ND = 45\% \quad (5.39)$$

5.6.5 Dúvida por Questão

Dado um T , de modo que $T = [t_0, t_1, \dots, t_k]$ seja o vetor de transições da rede de Petri e $k \in \mathbb{N}^*$, e um A , de modo que $A = [a_0, a_1, \dots, a_u]$ seja o vetor de arcos da rede e $u \in \mathbb{N}^*$, onde $A \subseteq (P \times T) \cup (T \times P)$. Logo, $A_{In} = (T \times P)$ é o vetor de arcos de entrada da rede e $A_{Out} = (P \times T)$ o vetor de arcos de saída. Para obter o nível de dúvida que o estudante teve em uma questão é necessário avaliar a cardinalidade de A_{In} , ou seja:

$$A_{In}(p_q) = (t, p_q), \forall t \in T \quad (5.40)$$

Com isso, a dúvida DQ de um estudante é sempre igual ao A_{In} de maior cardinalidade. Logo:

$$DQ = \text{Max}(A_{In}(p_q)) - 1 \forall p_q \in P_Q \quad (5.41)$$

DQ é medido conforme o número de vezes que o estudante respondeu à uma questão, assim:

- Se $DQ = -1$, o estudante não respondeu àquela questão;
- Se $DQ = 0$, o estudante respondeu à questão uma única vez, não havendo dúvida;
- Se $DQ > 0$, o estudante obteve dúvida em p_q e $DQ \in \mathbb{N}$

Exemplo: Seja o cenário descrito na Seção 5.6.1 e observando a Figura 5.24, o estudante respondeu a p_{q1} uma vez (representado pelo círculo em vermelho), p_{q3} e p_{q4} duas vezes (representados pelos retângulos). Nesse caso, para p_{q1} :

$$A_{In}(p_{q1}) = (t_1, p_{q1}) = 1 \quad (5.42)$$

Logo:

$$DQ_1 = \text{Max}(A_{In}(p_{q1})) - 1 \quad (5.43)$$

$$DQ_1 = 1 - 1 \quad (5.44)$$

$$DQ_1 = 0 \quad (5.45)$$

$$DQ_4 = \text{Max}(A_{In}(p_{q4})) - 1 \quad (5.53)$$

$$DQ_4 = 2 - 1 \tag{5.54}$$

$$DQ_4 = 1 \quad (5.55)$$

5.6.6 Nível de Confusão

Assumindo a Definição 5 e tomando C_C como o conjunto das questões corretas de forma que $C_C = \{p_{vq} \in M_F | p_{vq} = 1\}$. Existe um T , de modo que $T = [t_0, t_1, ..., t_k]$ seja o vetor de transições da rede de Petri e $k \in \mathbb{N}^*$, e um A , de modo que $A = [a_0, a_1, ..., a_u]$ seja o vetor de arcos da rede e $u \in \mathbb{N}^*$, onde $A \subseteq (P \times T) \cup (T \times P)$. Logo, $A_{In} = (T \times P)$ é o vetor de arcos de entrada da rede. A quantidade de marcações MR das respostas pode ser obtida por do somatório dos arcos de maior cardinalidade para as respostas do estudante, ou seja:

$$MR = \sum Max(A_{In}(p_r)) \forall p_r \in P_R \quad (5.56)$$

Dessa forma, o nível de confusão NC pode ser calculado por meio de:

$$NC = (1 - \frac{|C_C|}{MR}) \times 100 \quad (5.57)$$

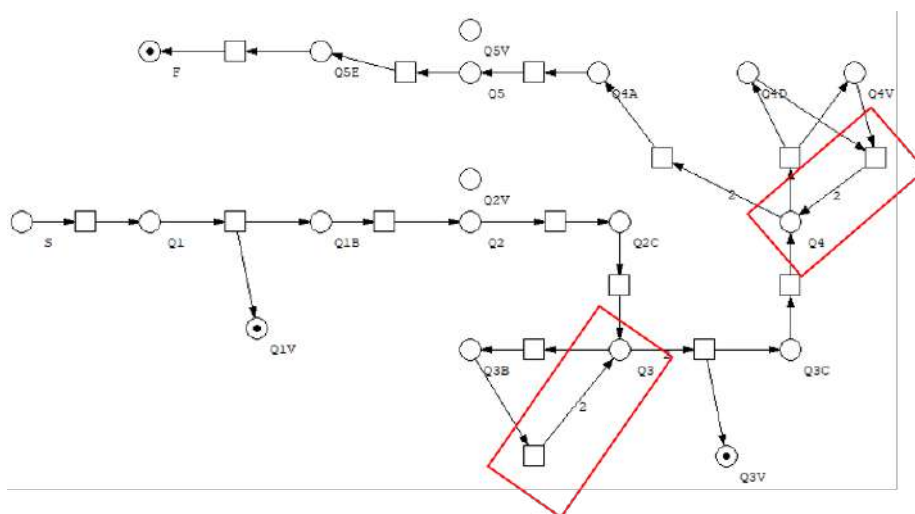


Figura 5.25: Estado m_{15} da rede de Petri com os lugares de questão que apresentaram confusão.

Exemplo: Exemplo: Seja o cenário descrito na Seção 5.6.1, uma vez que o estudante possua $C_C = \{p_{v1}, p_{v3}\}$, pois $M_F(p_{v1}) = 1$ e $M_F(p_{v3}) = 1$, seu $|C_C| = 2$. Para obter o MR , observamos pela Figura 5.25 que o estudante respondeu a todas as questões 1 vez, com exceção das questões Q_3 e Q_4 em que respondeu duas vezes, logo:

$$MR = 1 + 1 + 2 + 2 + 1 \quad (5.58)$$

Portanto, seu nível de confusão é de:

$$NC = (1 - \frac{2}{7}) \times 100 \quad (5.59)$$

$$NC = (1 - 0,29) \times 100 \quad (5.60)$$

$$NC = 0,71 \times 100 \quad (5.61)$$

$$NC = 71\% \quad (5.62)$$

O estudante ter obtido um nível de confusão de 71% se dá pelo fato de ser um questionário pequeno (com 5 questões) e de o aluno ter mudado de resposta 4 vezes (em Q_3 e Q_4).

5.7 Resumo

Este capítulo apresentou a metodologia proposta, sendo dividido em três etapas: Métricas e Pré-processamento, Avaliação automática e Geração de *feedback*. As métricas apresentadas são baseadas no trabalho de Leitão (2017), sendo elas: Nota Tradicional, Nota Ponderada, Nível de Desvio, Nível de Dúvida por questão, Nível de Confusão por questionário.

A etapa de Pré-processamento consiste na transformação das informações contidas nos *logs* de respostas dos alunos ao responderem um questionário avaliativo na plataforma educacional utilizada. É detalhado como os componentes da rede de Petri são criados e como as informações das questões/respostas são armazenadas dentro da estrutura da rede.

Na etapa de Avaliação Automática foi explicado a sintetização das matrizes de marcação para os modelos de rede de Petri juntamente com o processo de execução das redes por meios dessas matrizes. Ao final é dado um exemplo de como as métricas podem ser obtidas a partir do resultado final de execução da rede.

Por fim são apresentadas as formalizações das métricas com base no modelo de rede de Petri proposto para a avaliação automática. Esta formalização baseia-se nos conceitos abordados na Seção 2 sobre redes de Petri e definições da própria metodologia proposta, apresentando exemplos de cada métrica e relações destas com os componentes das redes.

Capítulo 6

Resultados

Este capítulo tem por objetivo descrever o planejamento e a aplicação dos estudos de caso para obtenção dos resultados da aplicação do método, contemplando itens como: objetivos, ameaças e resultados do caso de estudo aplicado. O método foi implementado usando a linguagem *Python* e foi aplicado ao banco de dados de questionários, que consistem em questões de múltipla escolha, com 40 questões pertencentes a diferentes áreas do conhecimento. Foi utilizada uma plataforma educacional desenvolvida por Leitão (2017). No experimento realizado, o questionário foi aplicado em 3 turmas, divididas em 2 dias de aplicação, e contemplou todas as métricas formalmente descritas na Seção 5.6. Além disso, é apresentado como resultado, a publicação da revisão sistemática da literatura, descrita no Capítulo 3, em um evento científico.

6.1 Publicações

A revisão sistemática da literatura, apresentada no Capítulo 3 foi submetida e aceita para publicação no Congresso Brasileiro de Informática na Educação - CBIE 2018¹, mais especificamente no XXIX Simpósio Brasileiro de Informática na Educação (SBIE), sob a trilha de Aspectos Teóricos, Metodológicos e Tecnológicos na construção de Ambientes e Sistemas Computacionais para Ensino/Aprendizagem (plataformas, arquiteturas, dispositivos móveis etc.) (Batista *et al.*, 2018).

6.2 Objetivos do estudo de caso

O objetivo principal do estudo de caso foi comprovar a aplicabilidade do método ao cenário proposto, com base em dados reais. Esta comprovação é possibilitada pelos seguintes objetivos específicos:

- Gerar as pontuações dos alunos conforme explicado na Seção 5.3, com base no modelo em rede de Petri de seu comportamento ao responder o questionário de múltipla escolha;
- Extrair informações relevantes sobre o desempenho do aluno e analisá-lo dentro da rede, a fim de fornecer recomendações sobre o nível de dúvida e desvio do aluno, além das métricas de nota tradicional e nota ponderada, e nível de confusão (consulte a Seção 5.2).

¹<http://cbie2018.virtual.ufc.br/index.php/sbie/>

6.3 Resultados experimentais

O estudo de caso foi feito com todas as métricas formalizadas e desenvolvidas. Para esse experimento, o método foi aplicado em 3 turmas, sendo duas delas divididas em 2 dias. Os questionários são divididos da seguinte forma:

Tabela 6.1: Distribuição das questões por turmas e por dias.

Turma	Dia	Número de Estudantes
01	1	33
	2	31
02	1	31
	2	17
03	1	34
Total		146

Observando a Tabela 6.1, tivemos um total de 146 alunos participantes do experimento. Os questionários aplicados tiveram um total de 44 questões cada. Essas 4 questões mutáveis são relacionadas à disciplina de língua inglesa ou língua espanhola.

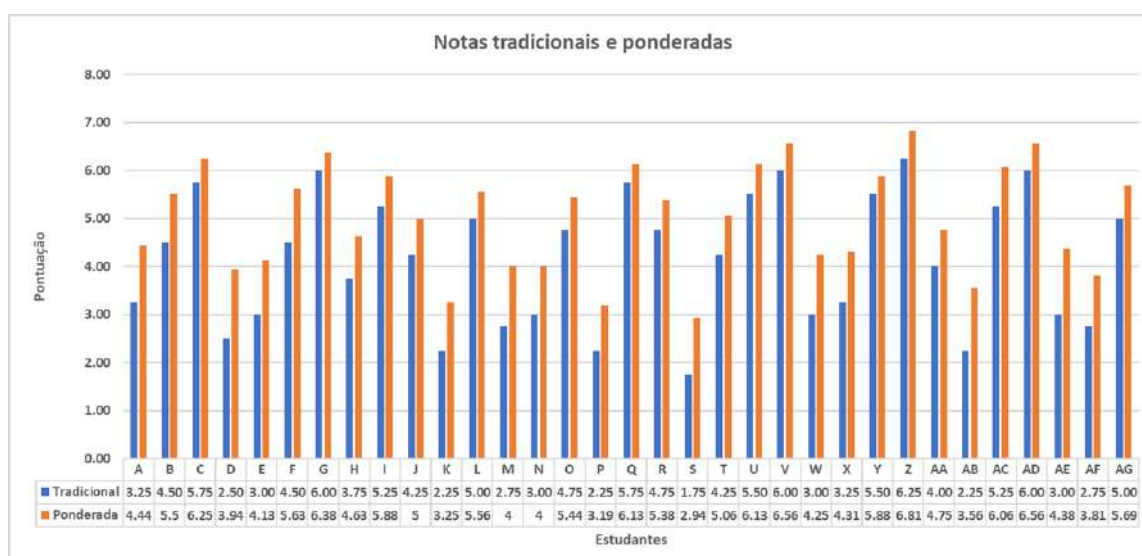


Figura 6.1: Relação entre as métricas de Nota Tradicional e Ponderada para a turma 01 do experimento no primeiro dia.

A Figura 6.1 mostra como foi o desempenho individual de cada estudante do primeiro dia da primeira turma, sendo comparadas duas métricas: Nota Tradicional e Nota Ponderada. Cada aluno respondeu 40 questões e foi avaliado automaticamente pelo método proposto.

A Nota Tradicional, apresentada pelas colunas azuis, mostra o resultado da avaliação dos estudantes levando em consideração somente o certo e o errado. É possível notar alguns picos em que as notas se sobressaem. No entanto, ao compararmos com a Nota Ponderada, representada pela coluna laranja, observamos alguns pontos de grande diferença ao levarmos em consideração os pesos das respostas dadas pelo estudante. Por exemplo, os estudantes *M* e *N* tiveram uma diferença mínima nos seus resultados da Nota Tradicional, mas ao avaliar seus resultados pela Nota Ponderada ficaram exatamente com a mesma nota, e uma nota maior que as suas tradicionais. Isso é consequência dos pesos das respostas que eles selecionaram no questionário. Esse efeito se aplica também aos que já

tinham uma nota alta pela Nota Tradicional e com a Ponderada acabaram por conseguir um aumento em seus resultados, como por exemplo estudantes *G*, *V*, *Z* e *AD*.

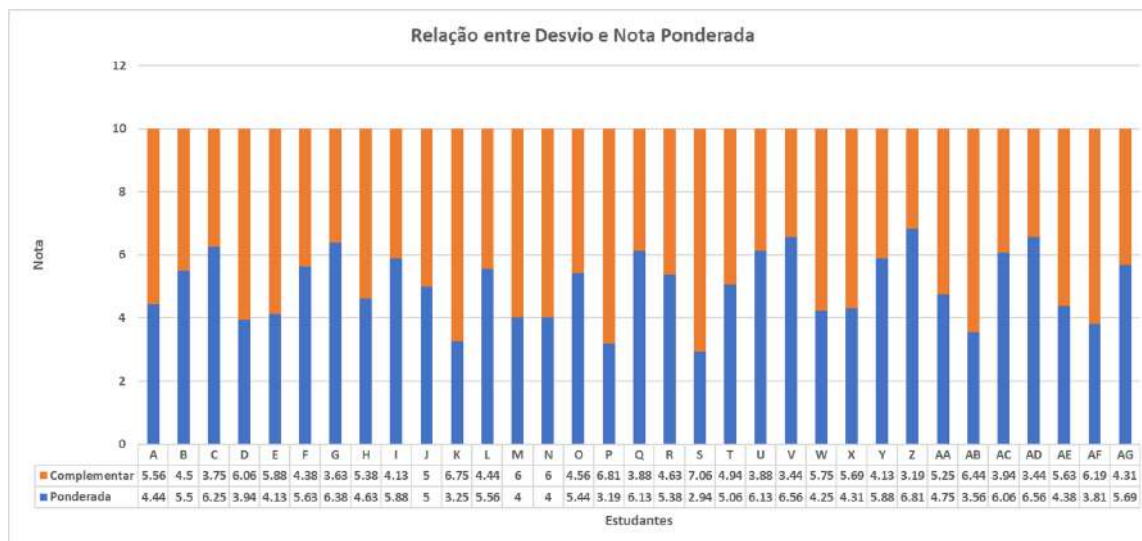


Figura 6.2: Relação entre as métricas de Nota Ponderada e Nível de Desvio para a turma 01 do experimento no primeiro dia.

Ainda sobre a turma 1 no primeiro dia, a Figura 6.2 mostra a comparação entre o Nível de Desvio e a Nota Ponderada obtida pelos estudantes. Como foi percebido, o desvio do estudante é complementar ao seu desempenho da Nota Ponderada. Isso é visível no estudante *S* que possui um valor altíssimo de nota complementar baseada no desvio quando comparado a sua nota ponderada, aproximadamente 70% de desvio. Então quanto maior o desvio de um estudante, menor sua nota ponderada.

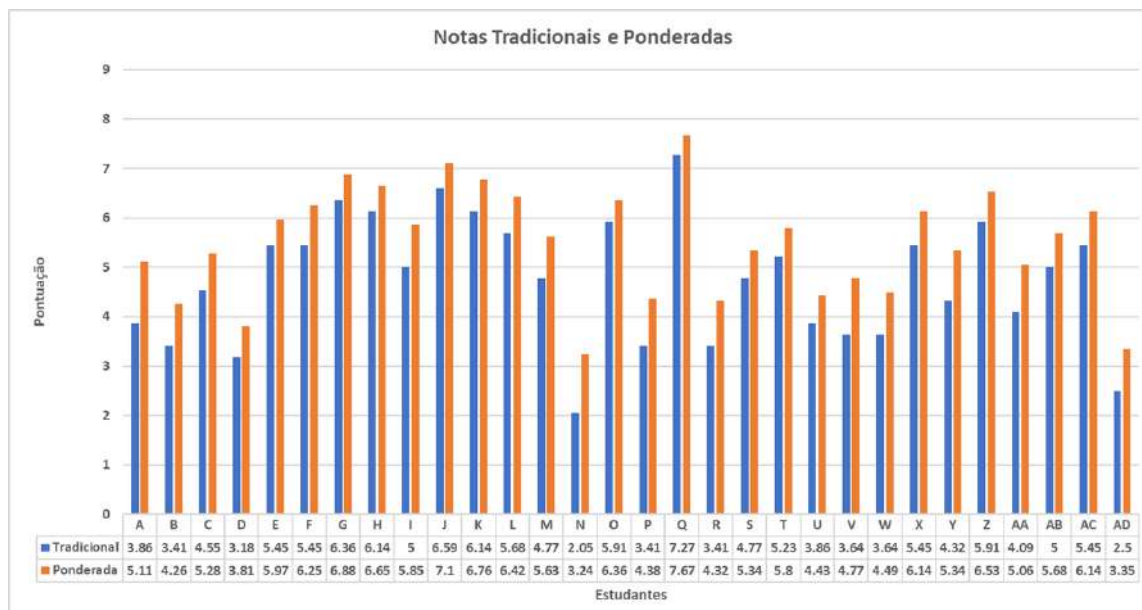


Figura 6.3: Relação entre as métricas de Nota Tradicional e Ponderada para a turma 01 do experimento no segundo dia.

Na Figura 6.3, que representa o segundo dia do experimento feito com a primeira turma tivemos notas mais altas, tanto notas tradicionais quanto ponderadas, mas ainda assim observamos

estudantes que tiveram uma nota bem abaixo da média dos demais, e mesmo com sua avaliação da nota ponderada não obtiveram um bom resultado, como é o caso do estudante *N* e *AD*.

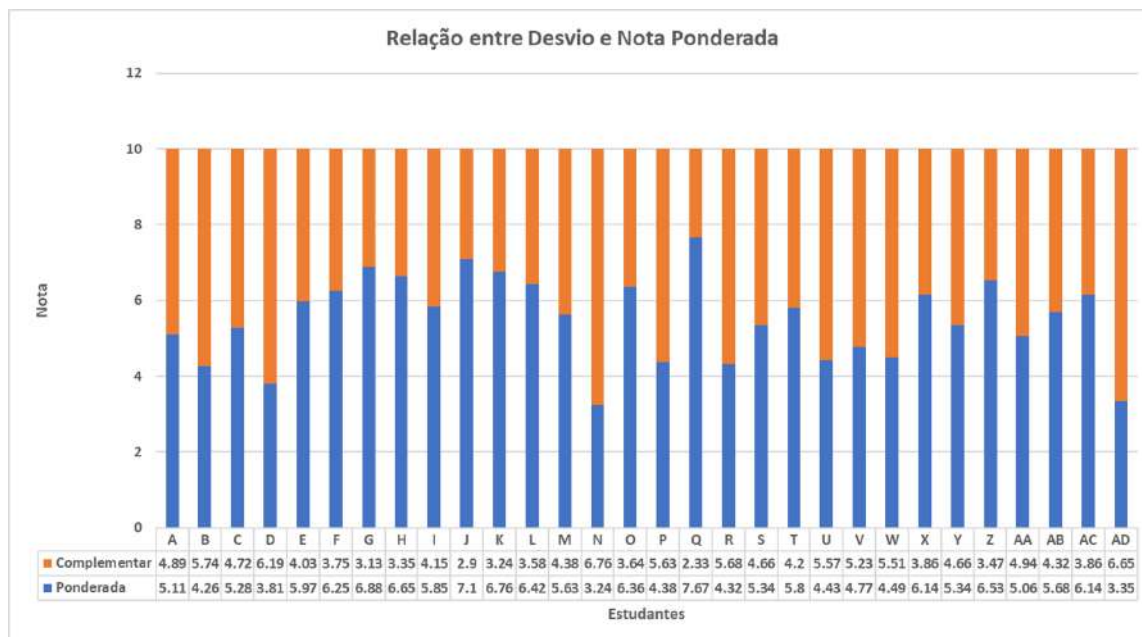


Figura 6.4: Relação entre as métricas de Nota Ponderada e Nível de Desvio para a turma 01 do experimento no segundo dia.

A Figura 6.4 nos dá um contraste interessante identificado durante a avaliação do Nível de Desvio, os estudantes *N* e *Q* tiveram seus desvios e notas ponderadas proporcionais. O estudante *N* teve um alto índice de desvio (67,6%) e uma nota ponderada muito baixa, enquanto o estudante *Q* obteve um índice de desvio de apenas 23,3% obtendo uma nota ponderada 7,67 pontos.

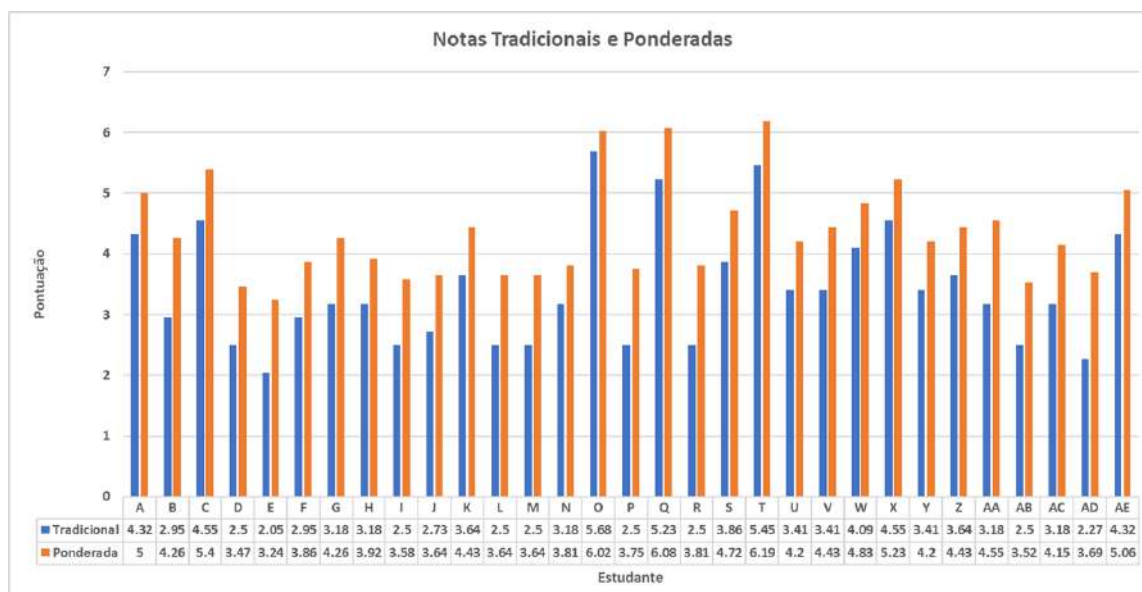


Figura 6.5: Relação entre as métricas de Nota Tradicional e Ponderada para a turma 02 do experimento no primeiro dia.

A turma 2 em seu primeiro dia conseguiu uma média de Notas Tradicionais (Figura 6.5) mais mistas se comparada a primeira turma em seu primeiro dia. As notas foram mais baixas no geral

(os estudantes *L* e *M* por exemplo só conseguiram chegar a nota ponderada de 3,0 pontos), havendo alguns casos em que a Nota Ponderada conseguiu ajudar o estudante a aumentar seu desempenho na avaliação automática (sendo o caso dos estudantes *O*, *Q* e *T*).

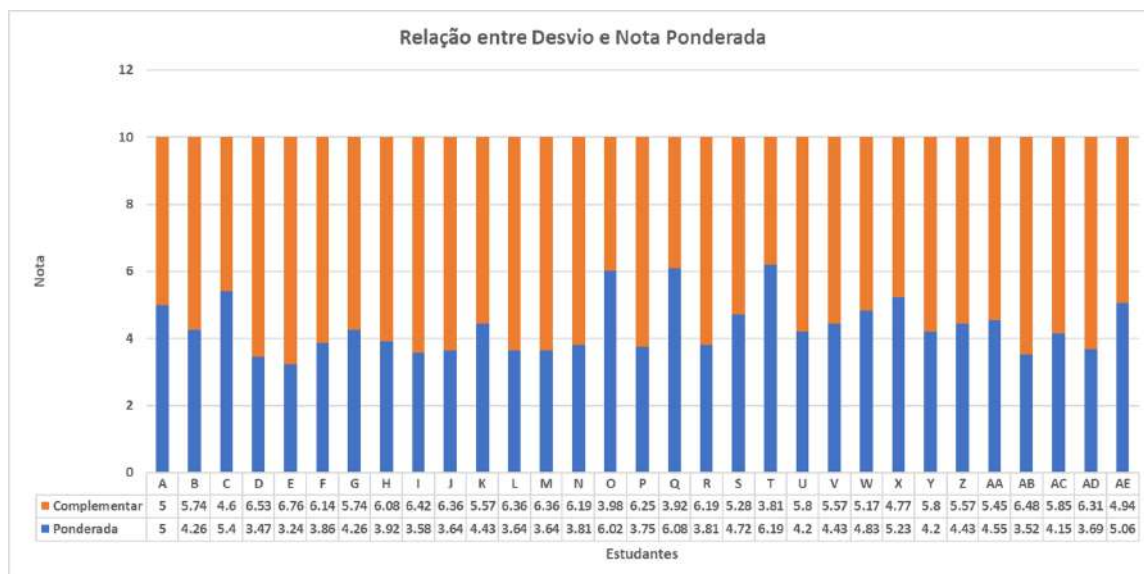


Figura 6.6: Relação entre as métricas de Nota Ponderada e Nível de Desvio para a turma 02 do experimento no primeiro dia.

Em consequência das Notas Ponderadas ainda serem baixas, conforme podemos ver na Figura 6.6, o Nível de Desvio dessa turma foi mais alto do que os outros dois dias da turma anterior, indicando que eles tiveram mais dificuldade no questionário, por exemplo, os estudantes entre o estudante *D* e *N* ficaram com um nível de desvio entre 60% e 70%.

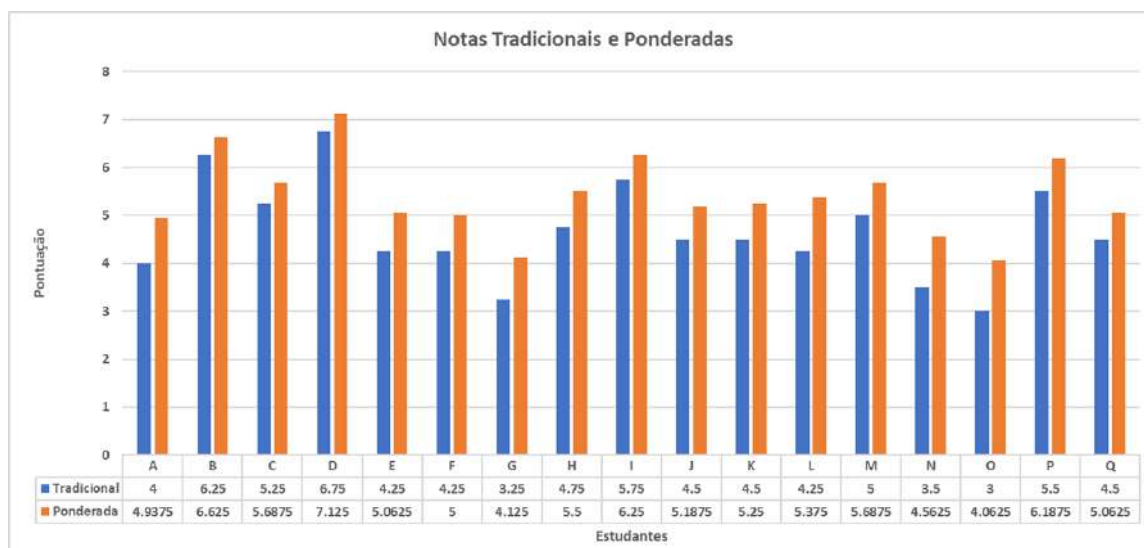


Figura 6.7: Relação entre as métricas de Nota Tradicional e Ponderada para a turma 02 do experimento no segundo dia.

O segundo dia da segunda turma teve menos participantes (como pode ser visto na Figura 6.7), logo o resultado do experimento realizado nesse dia teve uma coleta de dados menor. Ainda assim, é possível notar o desempenho dos estudantes em relação a suas avaliações, as notas foram mistas, não havendo um ponto muito fora da curva para mais ou para menos, exceto do estudante *D* que

obteve uma Nota Tradicional e Ponderada alta. A mesma coisa ao analisar o Nível de Desvio na Figura 6.8, existe um contraste ao avaliar como o estudante *D* se saiu e como os estudantes *N* e *O* saíram, por exemplo.

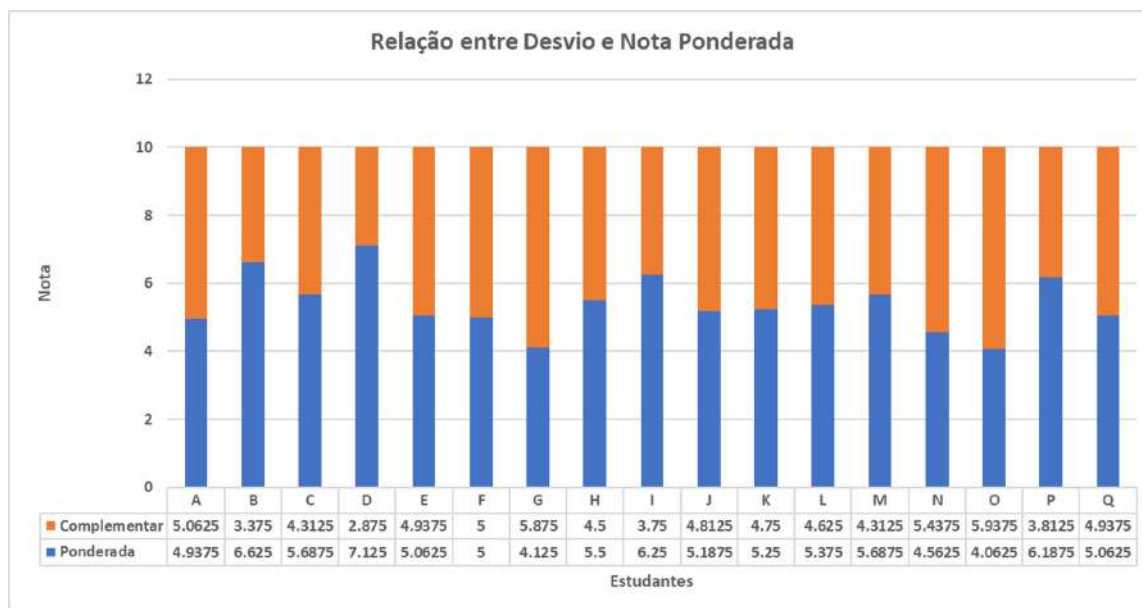


Figura 6.8: Relação entre as métricas de Nota Ponderada e Nível de Desvio para a turma 02 do experimento no segundo dia.

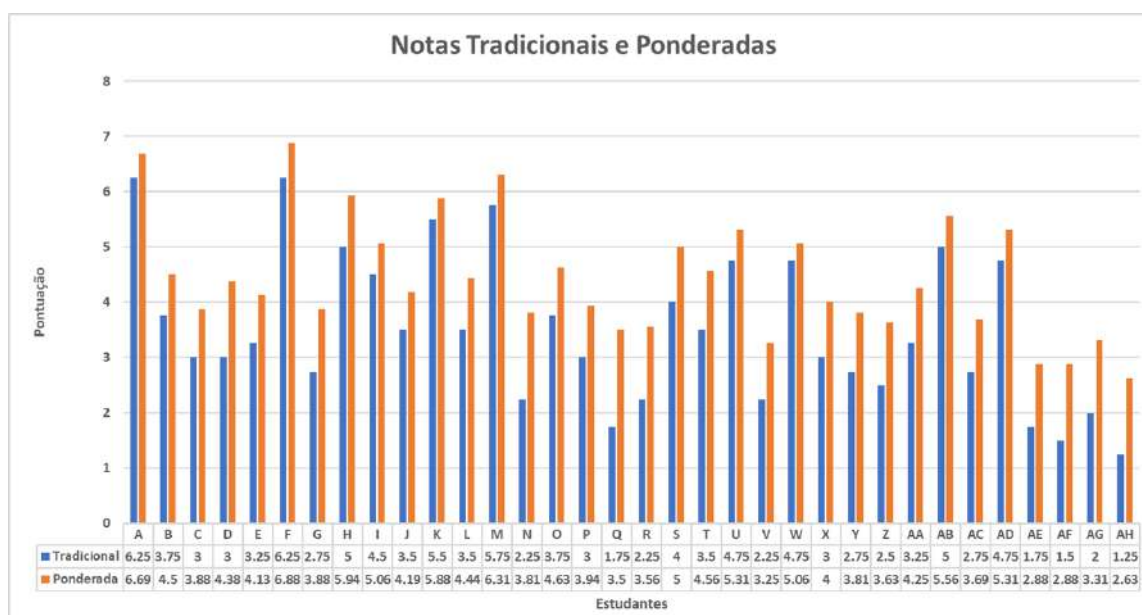


Figura 6.9: Relação entre as métricas de Nota Tradicional e Ponderada para a turma 03 do experimento no primeiro dia.

A terceira turma, representada pelas Figuras 6.9 e 6.10, teve somente um dia de participação no experimento. O interessante é que seus resultados foram um misto das 2 turmas anteriores, tendo seus picos (por exemplo os estudantes *B* e *D*) e vales (por exemplo os estudantes *G*, *N* e *O*) de notas durante a avaliação automática. Na Figura 6.9 é possível notar que a turma teve uma média de notas parecidas, variando entre 4,0 e 5,0 pontos, dando destaque para os estudantes *A* e *F* que tiveram

notas iguais em relação a Nota Tradicional. No entanto, se comparados pela Nota Ponderada, o estudante *F* teve um desempenho melhor que o estudante *A*.

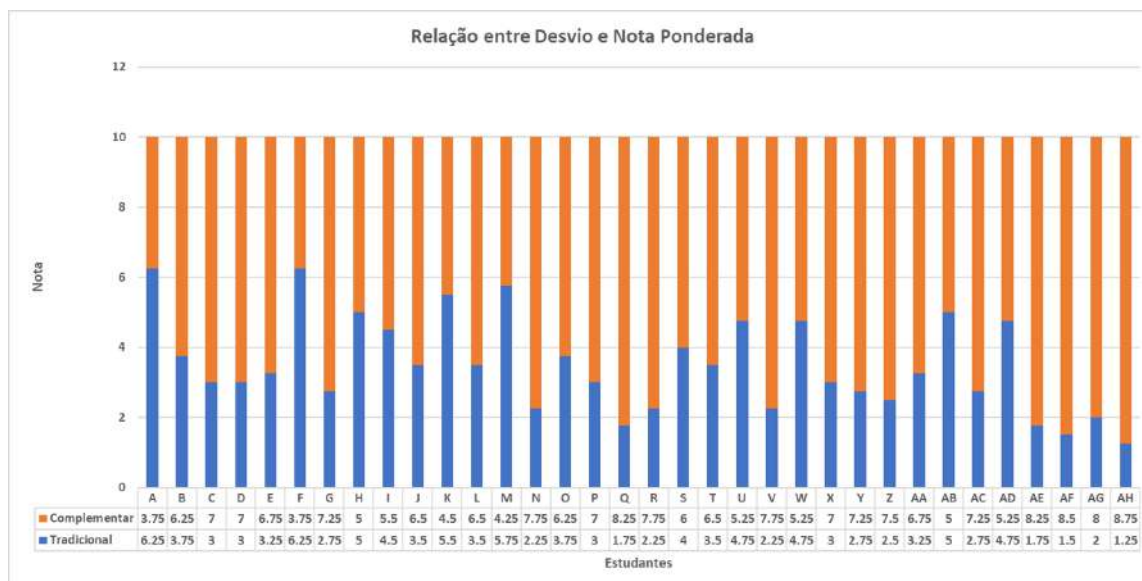


Figura 6.10: Relação entre as métricas de Nota Ponderada e Nível de Desvio para a turma 03 do experimento no primeiro dia.

A Figura 6.10 ajuda a compreender o porquê da terceira turma ser um misto das duas anteriores por meio dos níveis de desvios dos estudantes. Ao verificarmos o Nível de Desvio dos estudantes podemos ver mais uma vez que Nota Ponderada e Nível de Desvio são inversamente proporcionais, pois ao observar os estudantes *S* e *W* vemos que suas notas chegaram bem próximo do 5,0 exato, logo seus níveis de desvio também chegaram próximo de 50%. Além dos casos já observados nas turmas anteriores, desvios altos, notas ponderadas baixas, tendo ocorrido o maior nível de desvio observado entre as 3 turmas aqui, com o estudante *AH* obtendo 73,8% de desvio.

Devido ao fato de a métrica de Nível de Dúvida estar relacionada a dúvida por questão, apresentamos a seguir uma demonstração dos resultados que ela teve ao ser aplicado o método de avaliação automática. Essa demonstração toma como base a primeira turma em seus dois dias de exame.

A Figura 6.11 explana o número de estudantes que tiveram dúvida em uma questão em específico. As questões mostradas na Figura 6.11 foram as que mais tiveram ocorrência, dentre elas a questão 4 (*Q4*) foi a principal dúvida de 4 estudantes. No entanto, essa é uma visão geral em relação a todos os alunos.

Pela Figura 6.12, podemos ver que mesmo a questão 4 (*Q4*) ser a dúvida de 4 estudantes, ao avaliarmos individualmente cada questão vemos que a questão 22 (*Q22*) que foi dúvida de apenas 2 estudantes, foi, em um desses 2 estudantes, trocada de resposta 25 vezes. Essa informação só pode ser obtida por meio da métrica de Nível de Dúvida que verifica cada marcação da rede Petri feita pelo estudante durante seu exame. Vale ressaltar que esse valor é de um único estudante, em uma única questão, e que nesse caso apesar de mudar de resposta tantas vezes acabou respondendo corretamente.

Semelhante a Figura 6.11, a Figura 6.13 mostra as principais questões respondidas incorretamente e que foram dúvidas dos estudantes. Nela podemos perceber que a questão 6 (*Q6*) foi dúvida de 5 estudantes, e logo em seguida as questões 2 (*Q2*) e 32 (*Q32*) com 3 estudantes que tiveram dúvida.

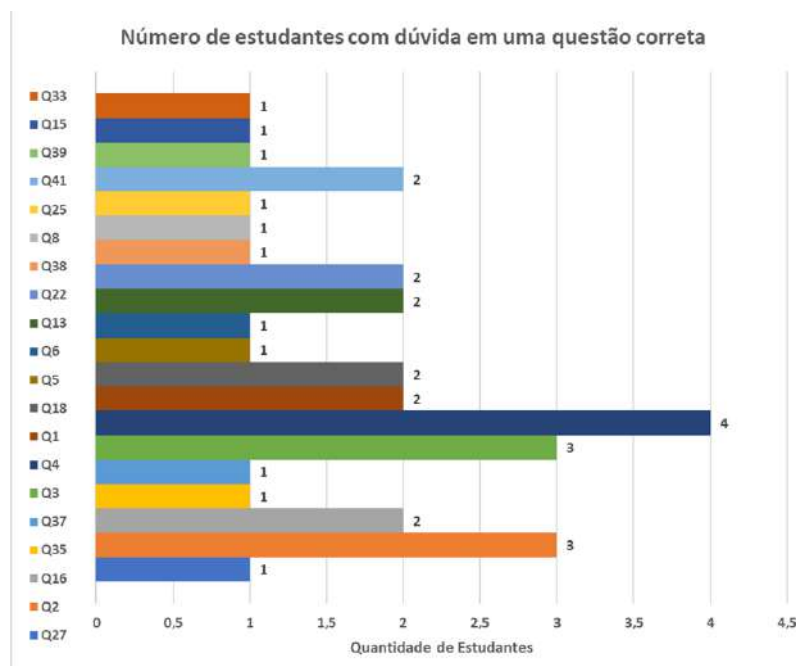


Figura 6.11: Principais questões respondidas corretamente em que os estudantes tiveram dúvida no primeiro dia.

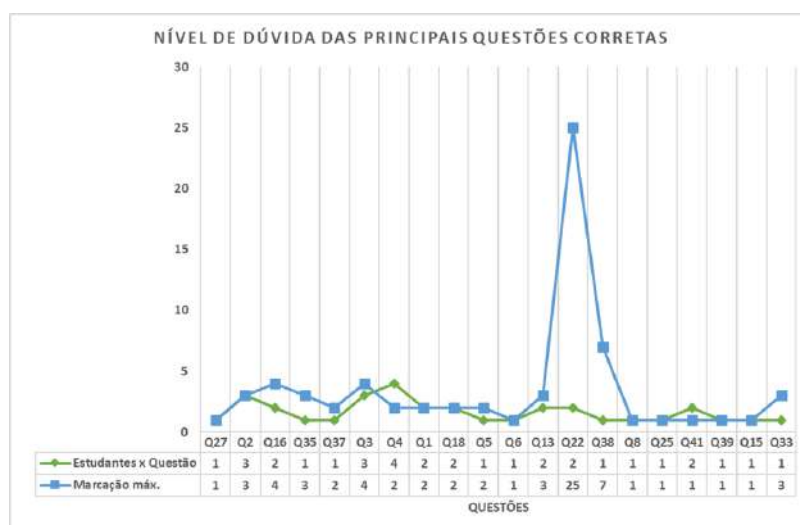


Figura 6.12: Nível de Dúvida dos estudantes avaliando individualmente as questões corretas no primeiro dia.

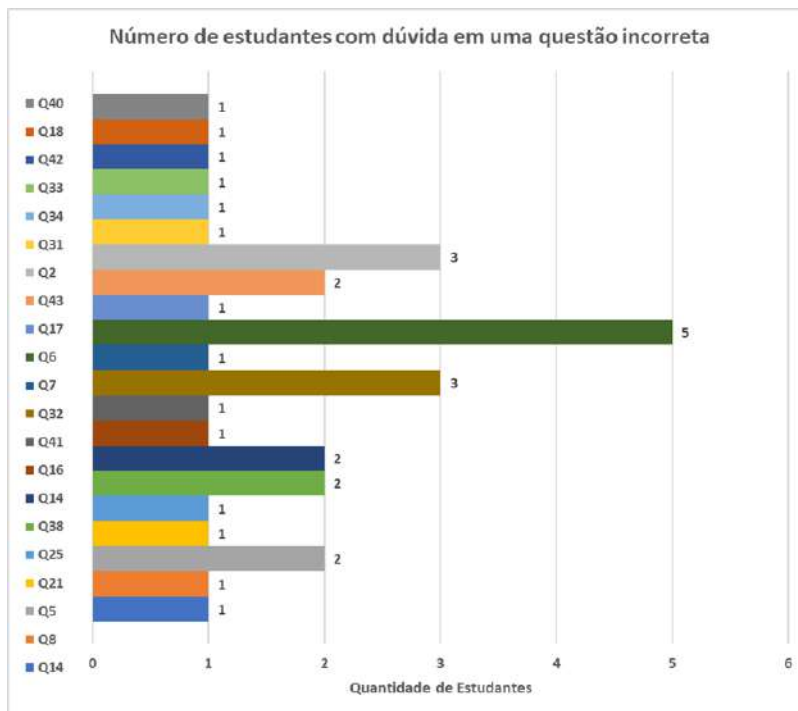


Figura 6.13: Principais questões respondidas incorretamente em que os estudantes tiveram dúvida no primeiro dia.

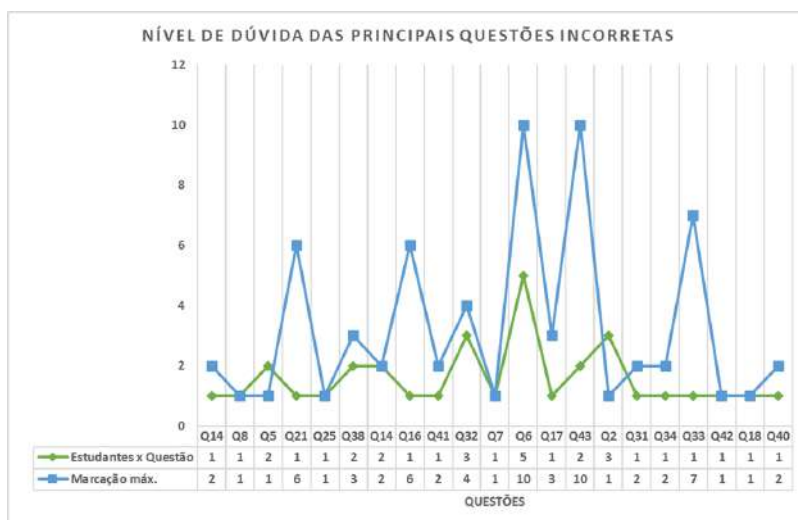


Figura 6.14: Nível de Dúvida dos estudantes avaliando individualmente as questões incorretas no primeiro dia.

Avaliando novamente individualmente cada questão, na Figura 6.14, vemos que a questão 6 (Q6) foi a que deixou mais estudantes em dúvida e também a que mais teve mudanças de respostas, sendo que um estudante chegou a trocar de resposta 10 vezes. O outro caso é da questão 43 (Q43) que foi dúvida de apenas 2 estudantes e um deles mudou de resposta 10 vezes.

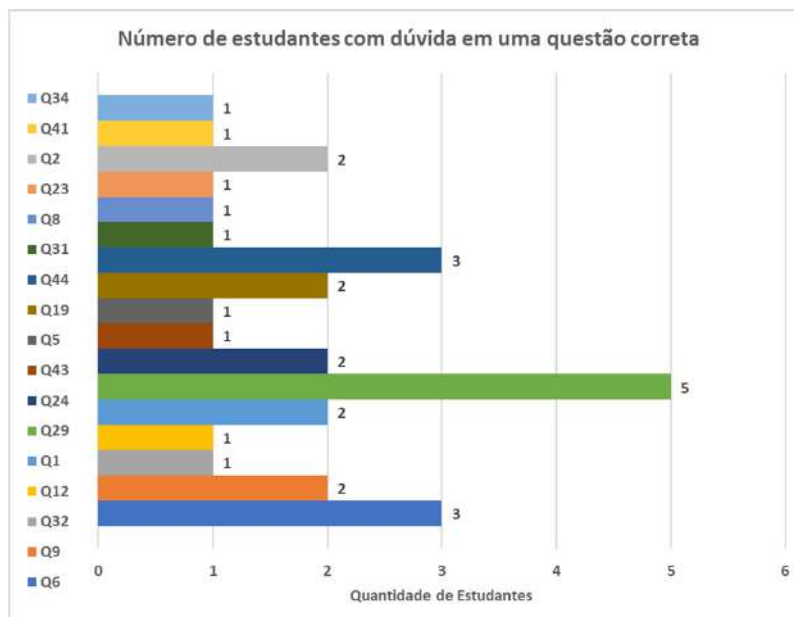


Figura 6.15: Nível de Dúvida dos estudantes avaliando individualmente as questões corretas no segundo dia.

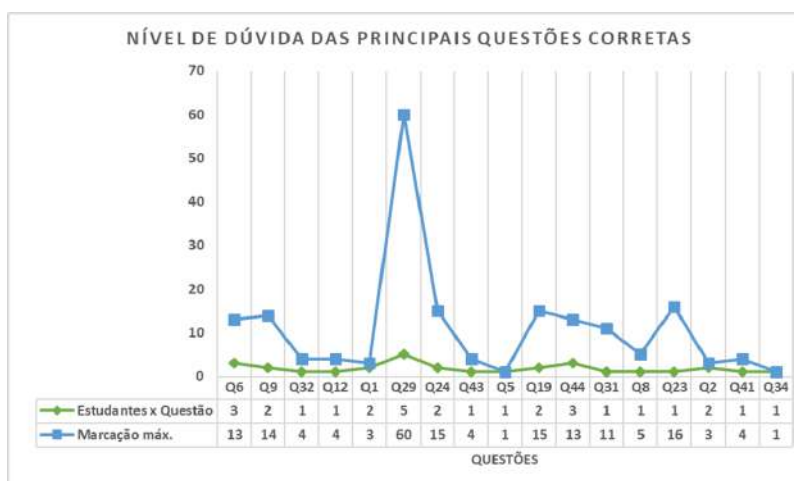


Figura 6.16: Nível de Dúvida dos estudantes avaliando individualmente as questões corretas no segundo dia.

Avaliando a turma no segundo dia por meio das Figuras 6.15 e 6.16, percebe-se que a questão correta com mais dúvida entre os estudantes foi a questão 5 (Q5) e nessa mesma questão um estudante chegou a mudar de resposta 60 vezes, sendo claramente um pico dentre as outras questões.

Já ao avaliarmos o nível de dúvida do segundo dia em relação as questões incorretas, Figuras 6.17 e 6.18, podemos notar que o número de alunos com mais dúvida em uma questão foi 3 estudantes, na questão 16 (Q16). Mas ao avaliarmos pela quantidade de marcações para medir o nível de dúvida

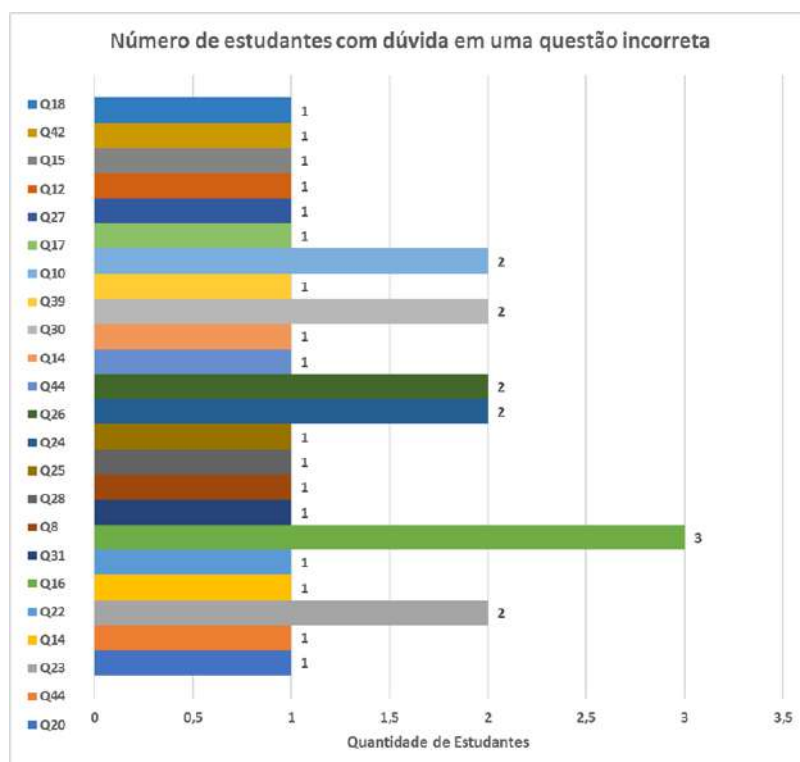


Figura 6.17: Nível de Dúvida dos estudantes avaliando individualmente as questões incorretas no segundo dia.

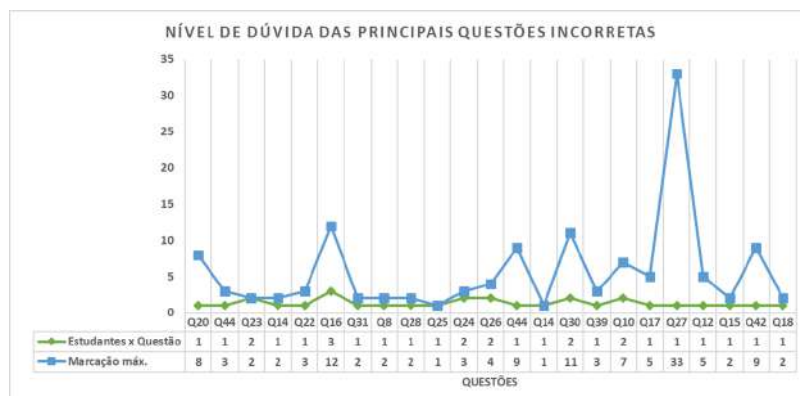


Figura 6.18: Nível de Dúvida dos estudantes avaliando individualmente as questões incorretas no segundo dia.

de uma questão, a questão 27 (*Q27*) que foi dúvida de um aluno teve sua resposta trocada 33 vezes, enquanto a questão 16 teve apenas 12 mudanças em sua resposta.

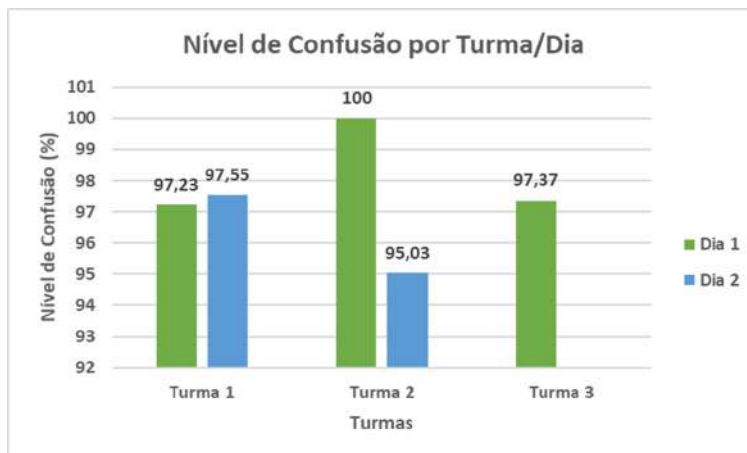


Figura 6.19: Nível de Confusão da turmas.

Ao avaliarmos as 3 turmas, na Figura 6.19, em relação a métrica de Nível de Confusão, podemos observar que a turma 2 em seu primeiro dia obteve um pico de 100% de confusão. Avaliando a primeira turma, vimos que na Figura 6.16, teve um pico de 60 alternativas que foram mudadas em uma questão, em vista disso, o nível de confusão da turma nesse momento foi de 91,55%. A terceira turma só participou do experimento no primeiro dia, por isso só possui uma coluna com 97,37% de confusão.

Em vista desse gráfico foi possível observar que o fator de decisão para mensurar o Nível de Confusão de uma turma, é o somatório da quantidade de vezes que os estudantes mudaram de alternativa durante a aplicação do questionário. A turma que menos obteve confusão ao final das avaliações foi a turma 2, em seu segundo dia de avaliação.

6.3.1 Limitações do experimento

Todo o método de avaliação pode ser aplicado com os dados obtidos por meio da plataforma educacional utilizada. Foi realizado somente uma análise dos próprios modelos das redes de Petri geradas dos estudantes. Como o método foi desenvolvido como se fosse parte de um *back-end* da aplicação, ele não possui uma interface em que o professor possa ter acesso a esses gráficos diretamente, tendo boa parte do seu processo feito por meio de linhas de comando.

6.4 Feedback

Uma vez que tenhamos feito o processo de Avaliação Automática dos estudantes, observamos pela própria explicação dos resultados, que muitos *feedbacks* podem ser retirados ao montar simples gráficos representativos de cada métrica utilizada para avaliação. No entanto, existem outros *feedbacks* que podem ser gerados, baseados na estrutura formal das redes de Petri apresentadas nas Seções 5.2 e 5.3.

A estrutura proposta na Figura 5.4 armazena informações importantes para a geração de *feedback*, dentre elas as informações de questão, como:

- Disciplina ao qual a questão está relacionada;
- Tópico que está relacionado a uma disciplina;
- Nível de dificuldade da questão.

Pelo segundo experimento realizado, as disciplinas utilizadas na avaliação da primeira turma, em seu primeiro dia foram:

Tabela 6.2: Conjuntos de questões por disciplinas da primeira turma, no primeiro dia, do segundo experimento.

Id da disciplina	Nome da disciplina	Conjunto de questões
1	Língua Portuguesa e Literatura	{Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8}
3	Língua Inglesa	{Q13, Q14, Q15, Q16}
4	História	{Q17, Q18, Q19, Q20}
5	Geografia	{Q21, Q22, Q23, Q24}
6	Matemática	{Q25, Q26, Q27, Q28, Q29, Q30, Q31, Q32}
7	Física	{Q33, Q34, Q35, Q36}
8	Biologia	{Q37, Q38, Q39, Q40}
9	Química	{Q41, Q42, Q43, Q44}

Com essas informações é possível criar relações na ordem como o estudante respondeu a questão, para demonstrar esse tipo de *feedback*, vamos levar em consideração a Figura 6.20 que é uma rede de Petri gerada pela avaliação automática de um estudante da primeira turma no primeiro dia do segundo experimento.

Pela Figura 6.20, é possível notar uma organização nos conjuntos de questões, conforme a Tabela 6.2. Logo de início o estudante foca nas questões de Língua Portuguesa e Literatura, conseguimos observar o quanto esse estudante mudou de resposta na questão Q3, que foi questão pela qual o estudante iniciou seu questionário (ver Figura 6.21). Em suma, ele passa pela resposta correta pelo menos duas vezes (Q3C), alcançando o lugar de verificação da questão (Q3V).

Ainda nessa rede de Petri, com suporte da Tabela 6.2, é possível observar que o estudante acaba indo e voltando nas questões. Por exemplo, ele chega a finalizar seu questionário de Língua Portuguesa, mas depois de certo tempo, respondendo à outras questões de História e Geografia (Q17 a Q20), acaba retornando para mudar suas respostas. As disciplinas possuem subitens, que são seus conteúdos, então pode ocorrer o caso de algumas respostas estarem relacionadas e o aluno conseguir assimilar essa situação.

Pela Figura 6.22 podemos ver uma estrutura de rede de Petri mais complexa que a rede anterior. Nesse caso, o estudante começa a responder seu questionário na questões de matemática (Q25 a Q32), depois vai verificando pelo menos uma questão de cada uma das outras disciplinas. Sendo que ao chegar na disciplina de Língua Portuguesa (Q1 a Q8), começam a surgir muitas dúvidas em sua questão Q5, chegando a mudar de resposta seis vezes. Mas sua principal questão foi a questão Q6 (ver Figura 6.23), onde chegou a mudar de resposta onze vezes, alterando entre a resposta correta Q6B (e seu estado de verificação Q6V) e mais duas respostas incorretas. Esse mesmo estudante respondeu fragmentavelmente seu questionário de Matemática, sempre alternando entre as outras disciplinas, finalizando nas questões de história (de Q17 a Q20).

A Figura 6.24 aborda outra visão sobre como analisar o desempenho dos alunos. Vale a pena notar que a sequência de transições (eixo X) mostra os disparos de acordo com a resposta da ordem de

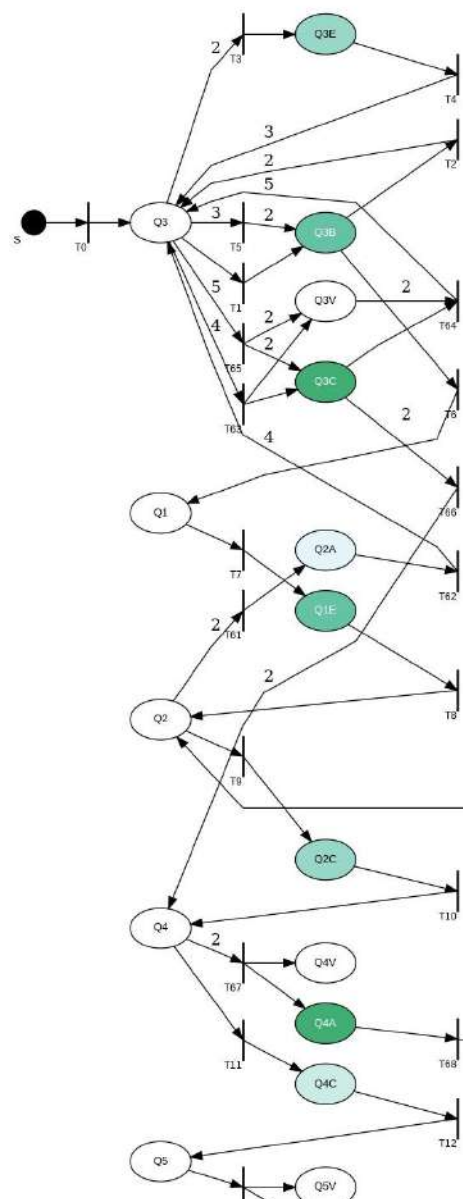


Figura 6.21: Dúvida em relação a questão Q3 da Figura 6.20.

perguntas e o número de marcações (eixo Y) são o nível de dúvida (métrica apresentada na seção 5.3) da questão. Os valores das marcações são obtidos com a Definição 3 (ver a Seção 5.4.1), sobre a matriz de entrada.

No gráfico do estudante 1 (Figura 6.24), o estudante permaneceu certo na maioria de suas respostas. Porém, ao responder a pergunta 5, sua resposta muda várias vezes; respondeu outras questões, mas voltou a questionar a 5 e continuou marcando as respostas erradas, exceto na 6ª vez em que respondeu, marcando a resposta correta, mas na 7ª marca outra resposta errada novamente. O professor pode utilizar essa observação para investigar o motivo disso ter acontecido. Por que o aluno mudaria de opinião, uma vez para a resposta certa e outra para a resposta errada? Seria falta de conhecimento? Outros alunos também apresentaram dúvida na mesma questão?

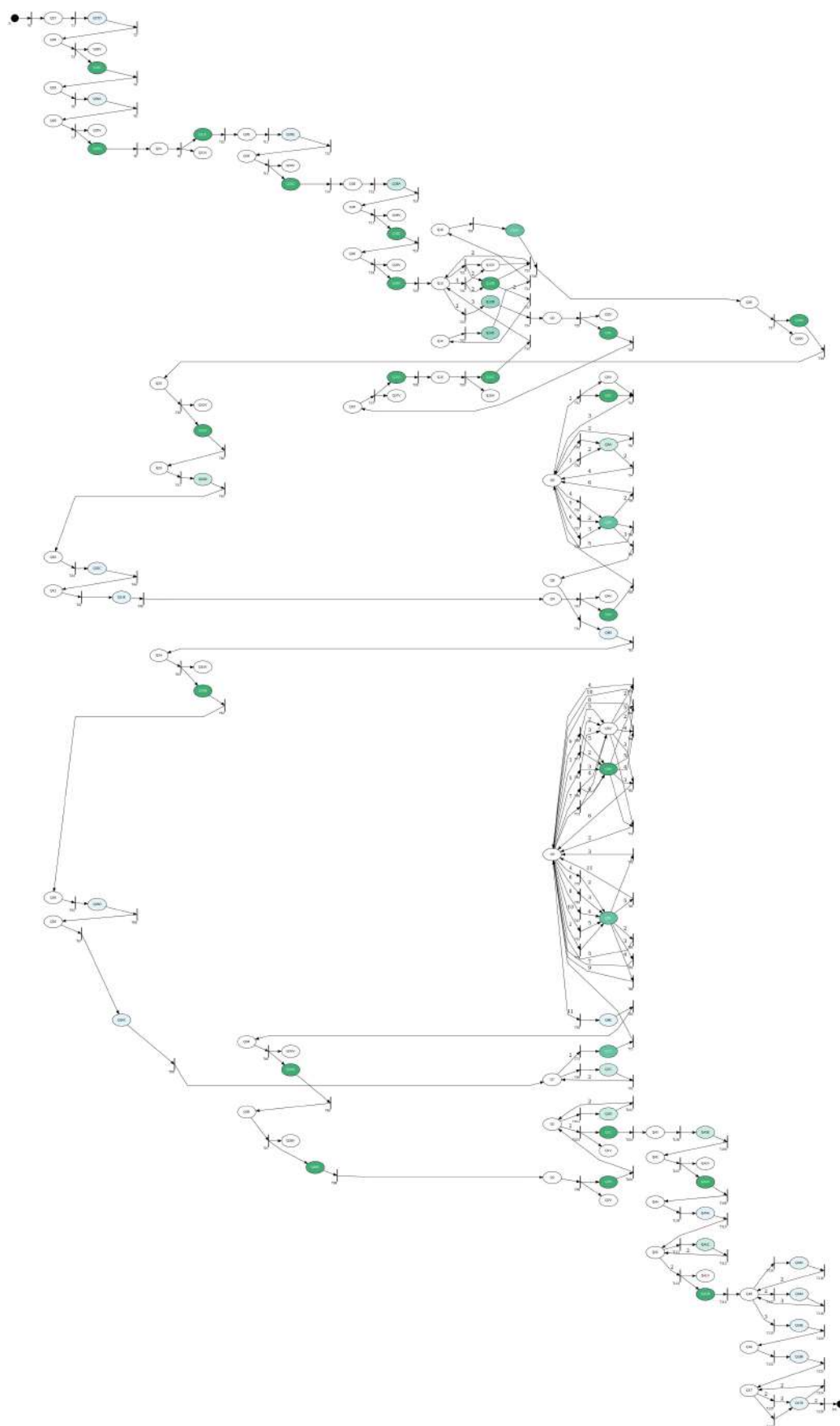


Figura 6.22: Rede de Petri gerada por um estudante do segundo experimento.

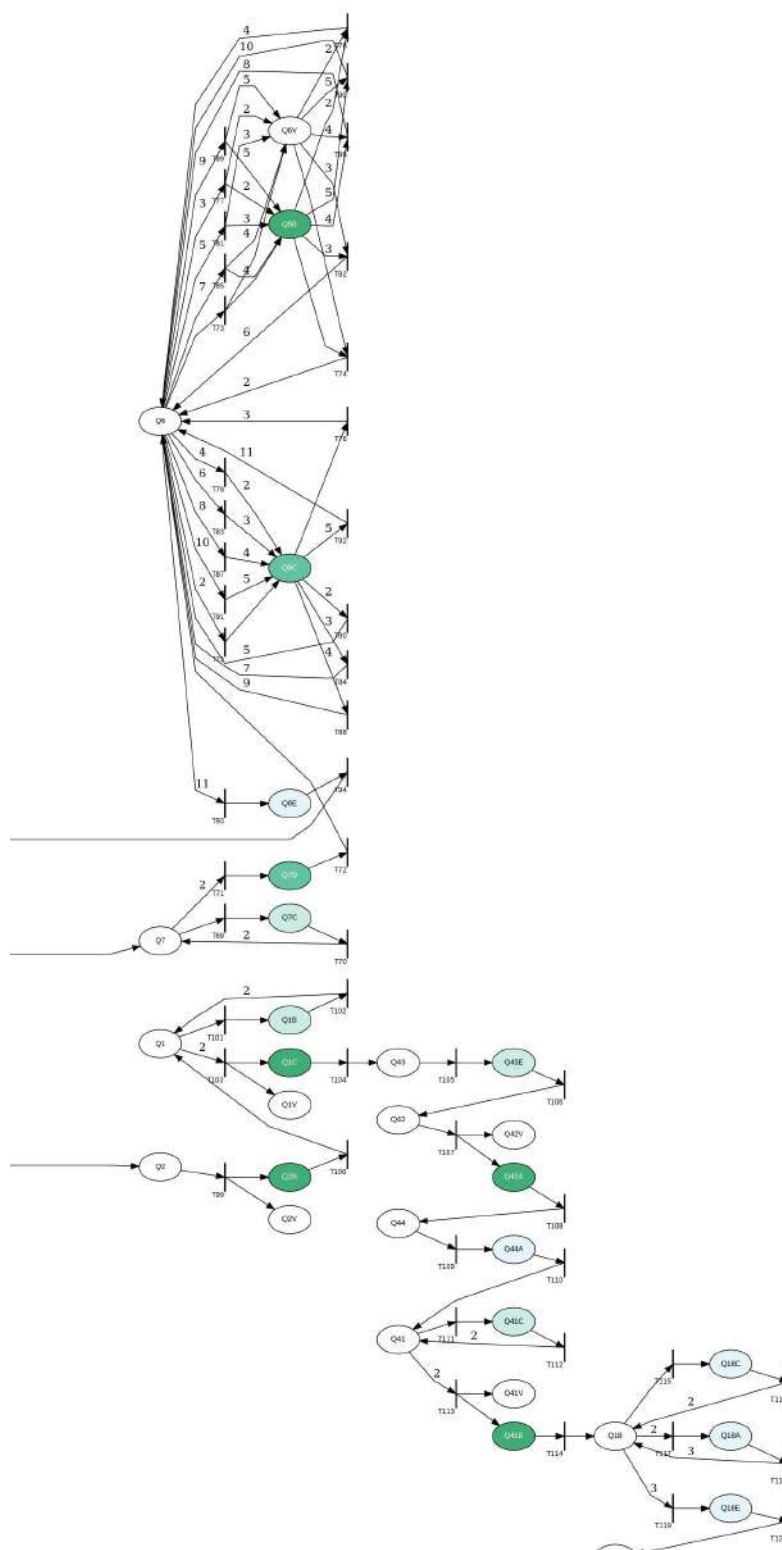


Figura 6.23: Dúvida em relação a questão Q3 da Figura 6.22.

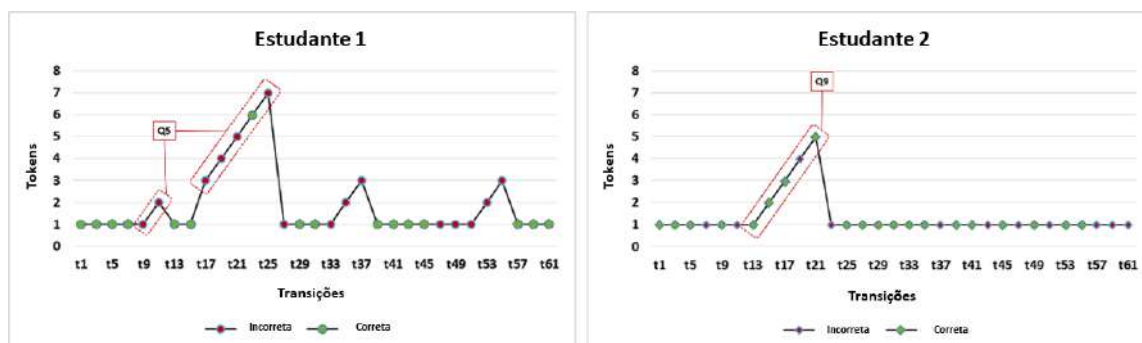


Figura 6.24: Nível de dúvida pelo número de disparos de transições.

No caso do estudante 2, o índice de dúvida está abaixo do aluno 1. Seu maior índice de dúvida estava na questão 9, quase o mesmo nível que o aluno 1, mas em uma questão diferente. Semelhante ao estudante 1, ele/ela permaneceu certo em suas outras respostas. Em sua dúvida, diferente do aluno 1, ele/ela manteve a resposta correta, mudando apenas uma vez para incorreta, mas depois voltou para a correta.

Em suma, a partir da análise dessas informações contidas na estrutura da rede de Petri, o professor pode identificar e orientar os alunos em seus pontos de dificuldade. Apontando suas falhas e mostrando como podem melhorar seu desempenho. Já os estudantes tem a oportunidade de observar seus próprio erros e recebem, por meio da avaliação automática, um retorno que realmente possa mensurar seu aprendizado, deixando de considerar somente o certo e o errado.

6.5 Resumo

Este capítulo explicou como os resultados foram obtidos. Passando desde a fase de planejamento dos estudos de caso, até suas aplicações. Os resultados foram relacionados as propostas e definições descritas no Capítulo 5.

O planejamento descrito abrange os objetivos e as limitações dos estudos de caso. Sobre as limitações, a principal provém da complexidade e dimensão dos modelos, que variam de aluno para aluno, já que no mínimo uma rede terá 130 lugares (considerando que o aluno respondeu a todas as questões do experimento e errou todas) e suas respectivas transições.

Os resultados experimentais são descritos em função das métricas (descritas na Seção 5.2) de Nota Tradicional e Ponderada, Nível de Desvio, Nível de Dúvida e Nível de Confusão. Além disso, são apresentadas duas formas de analisar os dados obtidos com a avaliação, sendo uma observando apenas as métricas e outra a partir da execução das marcações na estrutura da rede de Petri.

Por fim, são apresentados os gráficos de tais análises juntamente com a apresentação de algumas redes de Petri modeladas pela avaliação automática de alguns estudantes, sendo que mais modelos podem ser encontrados no link: https://drive.google.com/open?id=1ithpr_dMO6kfq4I6HJdPU61VMqkNwbpT.

Capítulo 7

Conclusão

Um dos principais desafios da área de automatização da avaliação e geração de *feedback* é propor metodologias que sirvam de apoio pedagógico para alunos e professores além de propor métodos que sejam aplicáveis a diversas áreas do conhecimento, que contribuam para o aprendizado dos estudantes e facilitem o trabalho dos professores, de modo que estes possam se dedicar a tarefas mais produtivas do que classificar, individualmente, o desempenho de seus alunos.

Neste trabalho foi apresentada uma proposta de método formal, utilizando um modelo conhecido como redes de Petri, que busca servir como meio de automatização do processo de avaliação e classificação do desempenho dos estudantes, bem como de proporcionar a geração automática de *feedbacks*, de modo que seja possível que este identifique suas falhas e seus pontos de maior dificuldade. Devido a problemas de desenvolvimento, não foi possível estabelecer uma automatização a nível de programação da geração de *feedbacks*, mas é apresentado uma breve análise de como ele pode ser gerado a partir da avaliação automática.

Objetivando analisar o estado da arte em relação aos métodos, metodologias e técnicas presentes na literatura referente a avaliação e geração de *feedback* automático, foi apresentada uma revisão sistemática da literatura. Esta, identificou onze métodos mais utilizados nessa área, sendo os três primeiros: métricas de similaridade, aprendizado de máquina e grafos. Nesta revisão, redes de Petri foi encontrada em menor proporção, demonstrando evidências de sua aplicabilidade e carência em pesquisa. Esta revisão foi submetida e aceita para publicação no XXIX Simpósio Brasileiro de Informática na Educação (SBIE) (consultar Batista *et al.* (2018)).

O método utilizado foi explicado em função de suas três etapas: pré-processamento e métricas, avaliação automática e geração de *feedback*. As métricas utilizadas são baseadas na proposta de Leitão (2017). O pré-processamento demonstra a transformação das informações contidas nos *logs* de respostas dos estudantes em componentes de uma rede de Petri. A etapa de avaliação automática consiste na apresentação das definições de construção das matrizes dos modelos de rede de Petri, para avaliação, e sua forma de execução para classificar e gerar a nota do desempenho do aluno. A geração de *feedback* demonstrou como informações relevantes podem ser extraídas da estrutura e da execução da rede de Petri.

Para comprovação do método, foi realizado um estudo de caso, utilizando a plataforma educacional disponibilizada por Leitão (2017) em que um questionário de múltipla-escolha foi respondido por duas turmas de 30 alunos cada no primeiro experimento, e três turmas, divididas entre dois dias, responderam ao questionário com 44 perguntas. Este experimento foram descritos desde seu planejamento, mostrando as limitações identificadas para eles, até seus resultados

experimentais, onde foi possível identificar relações que podem surgir da análise entre as métricas utilizadas e dos modelos gerados.

Com base no experimento realizado, o método proposto mostra ser promissor em suas análises e na maneira como busca fornecer *feedback* para os estudantes. Seu formalismo e sistematização garantem uma análise com forte base matemática e sua independência de bases de dados proporciona a adequação do método a questionários que gerem esse tipo de informação como saída, sendo necessário apenas o *log* de execução da plataforma educacional.

7.1 Trabalhos Futuros

Esta proposta foi composta pela implementação dos componentes de pré-processamento, avaliação automática e análises de *feedback* que podem ser gerados a partir dos modelos de avaliação, juntamente com aplicações de estudos de caso. Como citado em capítulos anteriores, existem vertentes que ainda precisam ser exploradas, de modo a melhorar a proposta do método. Dentre elas:

- **Avaliação Automática**

1. Incrementar o componente de avaliação automática: essa tarefa consiste em modificar o componente de avaliação automática, de modo que seja possível o professor selecionar mais de uma métrica para avaliação. Por exemplo: o professor poderá deixar que 50% da nota seja provida pela métrica de ‘Nota Tradicional’ e o restante seja dividido entre as métricas de ‘nível de dúvida’ (levando em consideração a quantidade de vezes que o estudante mudou de resposta para uma questão) e ‘nível de desvio’ (considerando o quão próximo da resposta correta o aluno conseguiu chegar). Estas métricas foram descritas na Seção 5.2;
2. Adaptar o modelo de forma a servir como suporte para a metodologia de avaliação de Teoria de Resposta ao Item;
3. Aplicação de novos casos de teste: reaplicar o método para comprovar a eficiência de suas alterações e inovações. Serão necessários novos estudos de caso, semelhantes aos que foram demonstrados nesta proposta, utilizando um conjunto novo de informações, que serão obtidas por meio da plataforma educacional de Leitão (2017);
4. Estudar a possibilidade de incrementar o método formal descrito nesse trabalho com aplicações de métodos de aprendizado de máquina e redes neurais, tal como evidenciado suas aplicações nessa área pela Seção 3.

- **Feedback Automático**

1. Utilizar o formalismo desenvolvido neste trabalho e propor um formalismo secundário para a geração automática de *feedback*;
2. O *feedback* pode ser baseado em algum modo visual, tal como a técnica de ‘Heatmaps’ descrita na Seção 3;
3. Utilizar as estruturas das redes de Petri geradas pelo método de avaliação automática para propor novos meios de *feedback* formal para professores e estudantes.

Apêndice A

Redes de Petri geradas

Este apêndice apresenta algumas das redes de Petri geradas pela modelagem do comportamento dos alunos ao responderem ao questionário avaliativo utilizado no estudo de caso descrito na Capítulo 6. Nas imagens é possível notar como os modelos de rede de Petri variam de aluno para aluno. É possível notar, por meio das colorações, os focos de dúvida dos estudantes. E em relação as respostas, os níveis de desvio estão destacados por suas colorações.

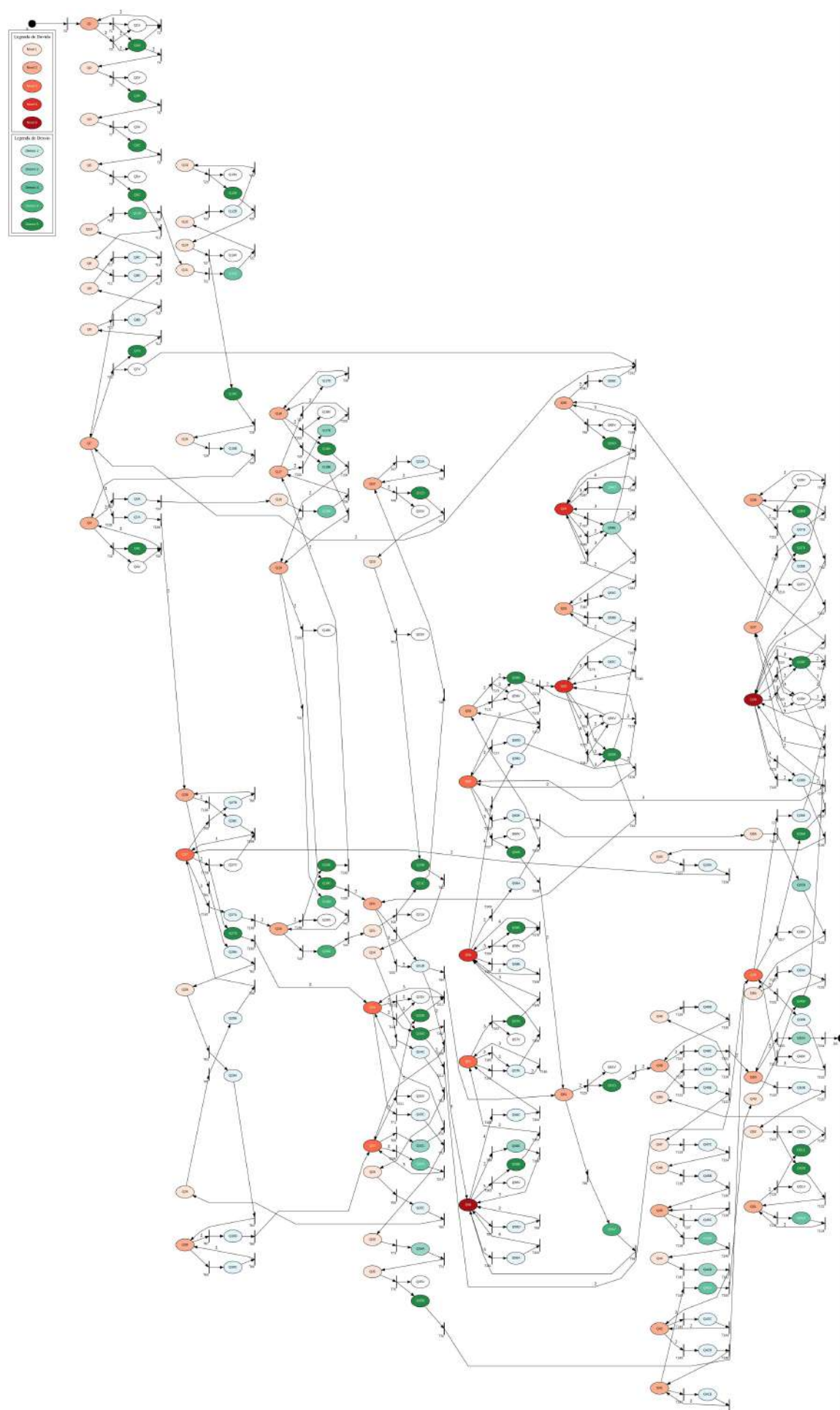


Figura A.1: Exemplo 1 do modelo de rede de Petri gerada para um estudante.

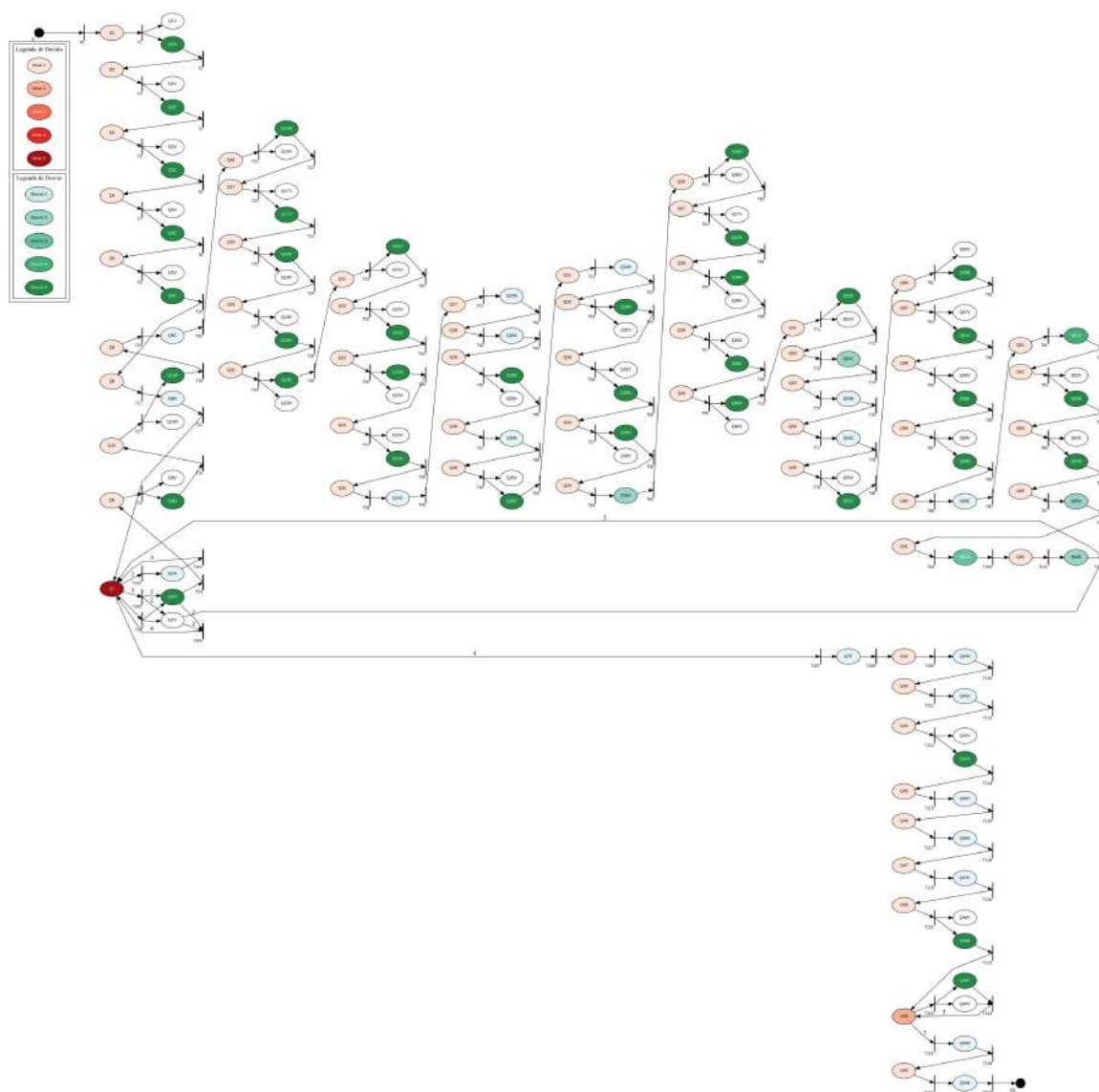


Figura A.2: Exemplo 2 do modelo de rede de Petri gerada para um estudante.

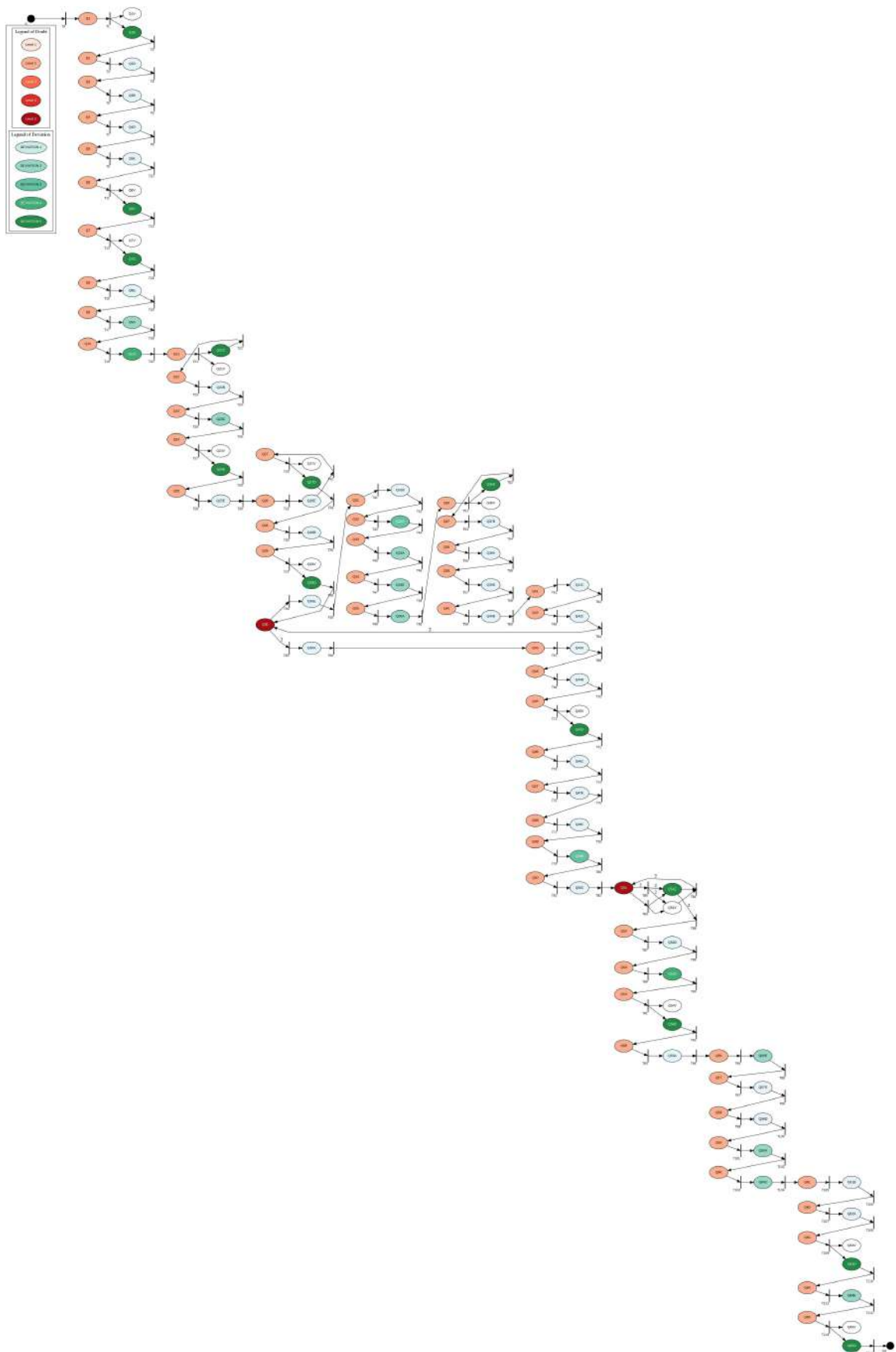
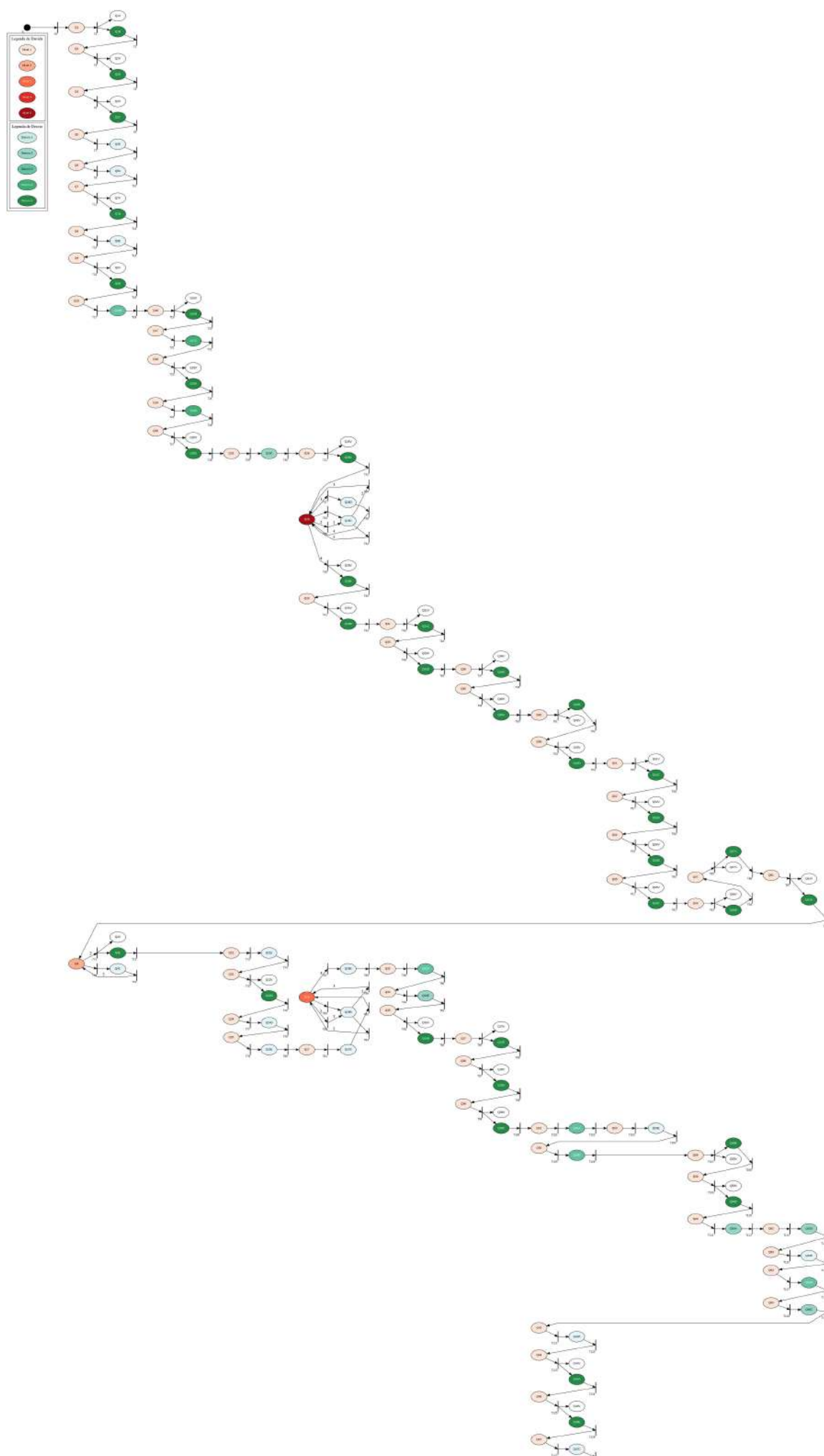
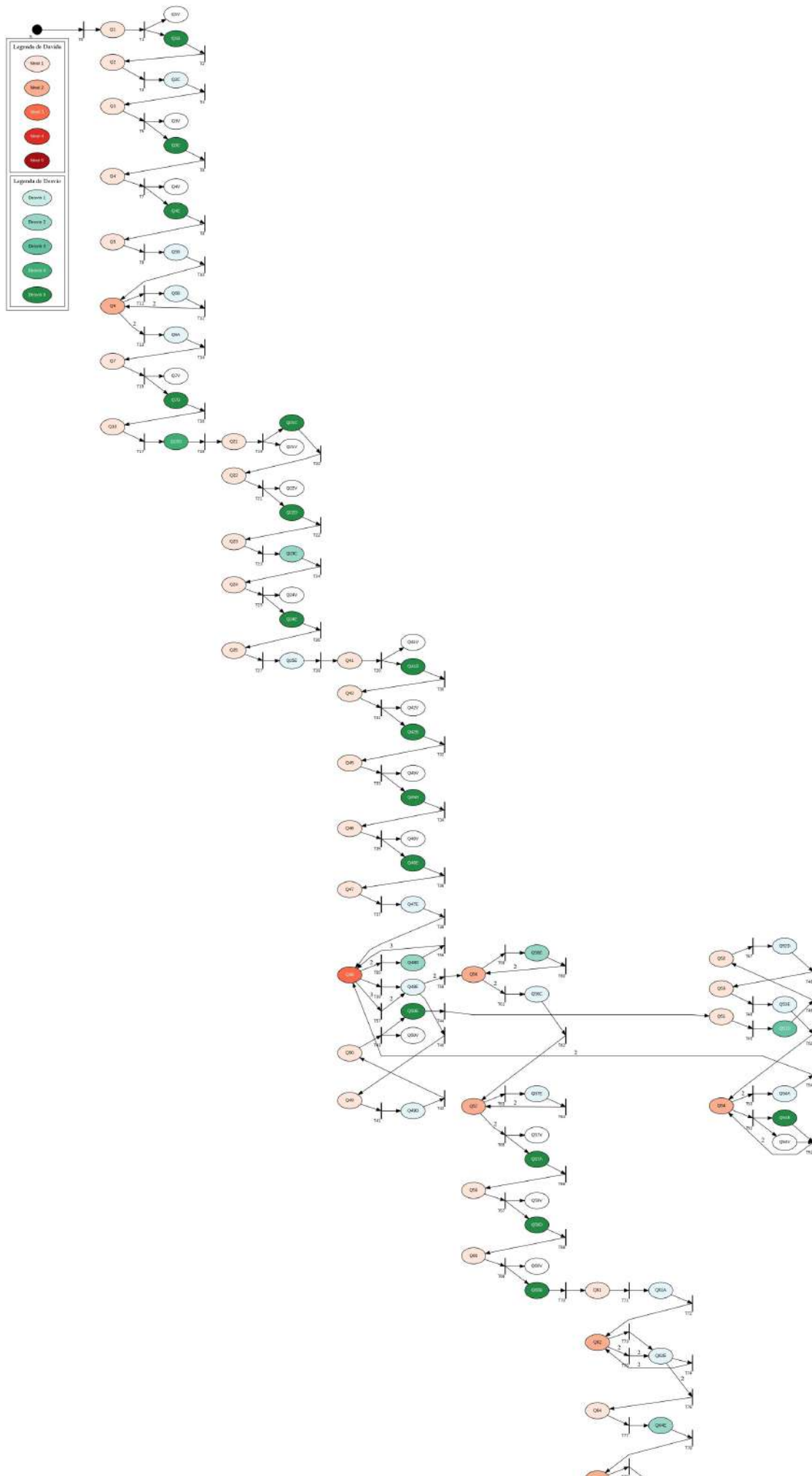


Figura A.3: Exemplo 3 do modelo de rede de Petri gerada para um estudante.





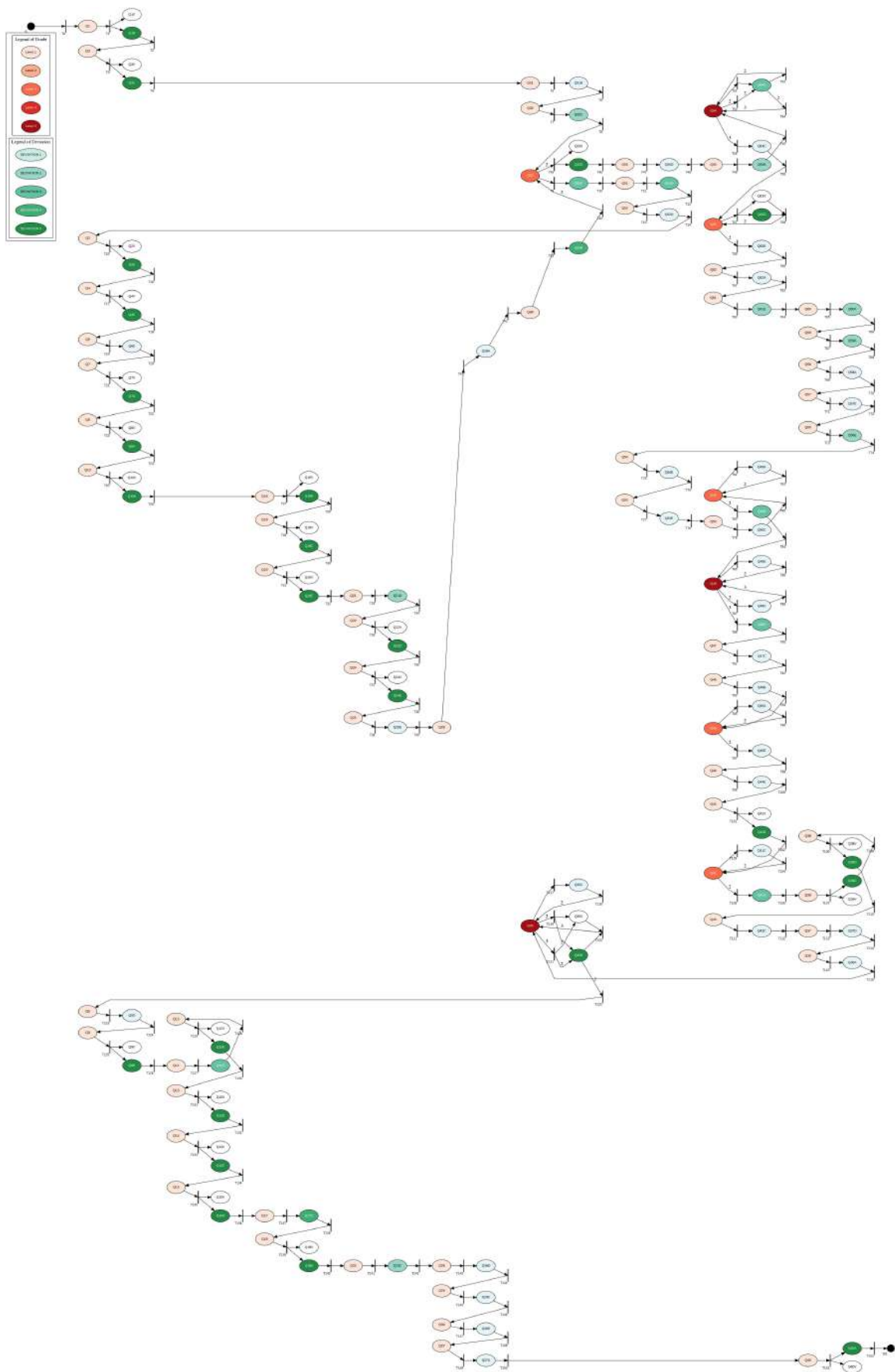


Figura A.6: Exemplo 6 do modelo de rede de Petri gerada para um estudante.

Referências Bibliográficas

- Abramson D. e Pike L. When formal systems kill: Computer ethics and formal methods. *APA Newsletter on Philosophy and Computers*, 11, 2011.
- AlShamsi F. e Elnagar A. Egrader: The programming solutions' grader in introductory Java courses. In *CSEDU 2011 - Proceedings of the 3rd International Conference on Computer Supported Education*, volume 2, páginas 36–45, 2011.
- Alur R., D'Antoni L., Gulwani S., Kini D. e Viswanathan M. Automated grading of DFA constructions. *IJCAI International Joint Conference on Artificial Intelligence*, páginas 1976–1982, 2013.
- Aziz M., Chi H., Tibrewal A., Grossman M. e Sarkar V. Auto-grading for parallel programs. In *Proceedings of the Workshop on Education for High-Performance Computing*, página 3. ACM, 2015.
- Baier C., Katoen J.-P. e Larsen K. G. *Principles of model checking*. MIT press, 2008.
- Bajaj N. e Seshia S. Autograding and Problem Generation for Hybrid Automata Constructions - CybOrg. *Formal Methods for Engineering Education*, 2014.
- Barreto R. d. S. *A Time Petri Net-Based Methodology for Embedded Hard Real-Time Software Synthesis*. PhD thesis, UNIVERSIDADE FEDERAL DE PERNAMBUCO, 2005.
- Basili V. R., Caldiera G. e Rombach H. D. Experience factory. *Encyclopedia of software engineering*, 1994.
- Basitere M. e Ivala E. The Effects of Wiley PLUS Web-based Homework system on student Performance in the Chemical Engineering Extended Curriculum Program: Introductory Physics Course. *ICEL 2015 - 10th International Conference on e-Learning*, páginas 31–40, 2015.
- Batista R. L., Salgado N. e Barreto R. Avaliação e feedback automático em educação apoiada por tecnologia: Um mapeamento sistemático da literatura. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 29, página 178, 2018.
- BenSassi M. e Laroussi M. Formal evaluation methodology of educational scenario in distance learning. In *International Conference on Web-Based Learning*, páginas 61–70. Springer, 2013.
- Bolón-Canedo V., Díez J., Luaces O., Bahamonde A. e Alonso-Betanzos A. Paving the way for providing teaching feedback in automatic evaluation of open response assignments. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, páginas 3447–3453. IEEE, 2017.
- Cardoso A. L. e Burham T. F. Construção do Conhecimento em Sala de Aula Mediada por um Ambiente Virtual de Aprendizagem. In *Simpósio Brasileiro de Informática na Educação*. SBC, 2007.

- Cassandras C. G. e Lafortune S. *Introduction to discrete event systems*. Springer Science & Business Media, 2009.
- Correia H., Leal J. P. e Paiva J. C. Enhancing feedback to students in automated diagram assessment. In *OASICS-OpenAccess Series in Informatics*, volume 56. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
- Cunha E., Custodio M., Rocha H. e Barreto R. Formal Verification of UML Sequence Diagrams in the Embedded Systems Context. *2011 Brazilian Symposium on Computing System Engineering*, páginas 39–45, 2011.
- Custódio M. M. VERIFICAÇÃO DE MODELOS UML DE SOFTWARE EMBARCADO COM MODEL CHECKING. Master's thesis, UNIVERSIDADE FEDERAL DO AMAZONAS, 2009.
- Danilov N., Shulga T., Frolova N., Melnikova N., Vagarina N. e Pchelintseva E. Software usability evaluation based on the user pinpoint activity heat map. In *Software Engineering Perspectives and Application in Intelligent Systems*, páginas 217–225. Springer, 2016.
- de Souza M. C., de Souza M. d. F. C., Gomes D. G., Barroso G. C., de Souza C. T., Filho J. A. d. C., Pequeno M. C. e Andrade R. M. C. Locpn: Redes De Petri Coloridas Na Produção De Objetos De Aprendizagem. *Brazilian Journal of Computers in Education*, 15, 2007.
- Edmison B., Edwards S. H. e Pérez-Quñones M. A. Using spectrum-based fault location and heatmaps to express debugging suggestions to student programmers. In *ACM International Conference Proceeding Series*, páginas 48–54. Association for Computing Machinery, 2017.
- Fast E., Lee C., Aiken A., Bernstein M. S., Koller D. e Smith E. Crowd-scale interactive formal reasoning and analytics. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, páginas 363–372. ACM, 2013.
- Gamalel-Din S. A. An intelligent grading system for e-learning environments. In *Proceedings of the 1st International Workshop on Interaction Design in Educational Environments - Volume 1: IDEE, (ICEIS 2012)*, páginas 3–13. INSTICC, SciTePress, 2012. ISBN 978-989-8565-17-4.
- Gisev N., Bell J. S. e Chen T. F. Interrater agreement and interrater reliability: key concepts, approaches, and applications. *Research in Social and Administrative Pharmacy*, 9(3):330–338, 2013.
- Godoy E. P., Lopes W. C., Sousa R. V., Porto A. J. V. e Inamasu R. Y. Modelagem e simulação de redes de comunicação baseadas no protocolo can - Controller area network. *Controle y Automacao*, 21:425–438, 2010.
- Gorgônio K. C. Adaptação de modelos em redes de petri coloridas. Master's thesis, UNIVERSIDADE FEDERAL DA PARAÍBA, 2001.
- Hsiao I.-H. e Govindarajan S. K. P. Semantic feedback for paper-based programming exams. In *Proceedings - IEEE 16th International Conference on Advanced Learning Technologies, ICALT 2016*, páginas 65–69. Institute of Electrical and Electronics Engineers Inc., 2016.
- Inventado P. S., Scupelli P., Heffernan C. e Heffernan N. Feedback design patterns for math online learning systems. In *Proceedings of the 22nd European Conference on Pattern Languages of Programs*, página 31. ACM, 2017.
- Juniwal G. *CPSGrader: Auto-Grading and Feedback Generation for Cyber-Physical Systems Education*. PhD thesis, University of California at Berkeley, 2014.

- Király S., Nehéz K. e Hornyák O. Some aspects of grading java code submissions in moocs. *Research in Learning Technology*, 25, 2017.
- Kitchenham B. Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33: 1–26, 2004.
- Kitchenham B., Brereton O. P., Budgen D., Turner M., Bailey J. e Linkman S. Systematic literature reviews in software engineering—a systematic literature review. *Information and software technology*, 51:7–15, 2009.
- Kitchenham B. e Brereton P. A systematic review of systematic review process research in software engineering. *Information and software technology*, 55:2049–2075, 2013.
- Kyrilov A. e Noelle D. C. Using case-based reasoning to improve the quality of feedback provided by automated grading systems. In *Proceedings of the International Conference e-Learning 2014 - Part of the Multi Conference on Computer Science and Information Systems, MCCSIS 2014*, páginas 384–388. IADIS, 2014.
- Lan A. S., Vats D., Waters A. E. e Baraniuk R. G. Mathematical language processing: Automatic grading and feedback for open response mathematical questions. In *L@S 2015 - 2nd ACM Conference on Learning at Scale*, páginas 167–176. Association for Computing Machinery, Inc, 2015.
- Landis J. R. e Koch G. G. The measurement of observer agreement for categorical data. *biometrics*, páginas 159–174, 1977.
- Leitão G. D. S. Uma Plataforma de Suporte ao Docente no Contexto da Educação Digital por Uma Plataforma de Suporte ao Docente no Contexto da Educação Digital. Master's thesis, UNIVERSIDADE FEDERAL DO AMAZONAS, 2017.
- Lima Filho D. L. e Queluz G. L. A tecnologia ea educação tecnológica: elementos para uma sistematização conceitual. *Educação & Tecnologia*, 10:19–28, 2005.
- Monin J.-F. e Hinchey M. *Understanding Formal Methods*. Springer-Verlag, 2003.
- Murata T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4): 541–580, 1989.
- Nakagawa E. Y., Scannavino K. R. F., Fabbri S. C. P. F. e Ferrari F. C. *Revisão Sistemática da Literatura em Engenharia de Software: Teoria e Prática*. Elsevier Brasil, 2017.
- Pape S., Flake J., Beckmann A. e Jürjens J. STAGE: A software tool for automatic grading of testing exercises: Case study paper. In *Proceedings - International Conference on Software Engineering*, páginas 491–500. IEEE Computer Society, 2016.
- Petri C. A. Communication with automata. 1966.
- Posada J. G., Baranauskas M. C. e Buchdid S. B. A informática na educação : o que revelam os trabalhos publicados no Brasil Informatics in education : what the works published in Brazil reveal. *Revista Brasileira de Informática na Educação*, 24, 2016.
- Pressman R. e Maxim B. *Engenharia de Software-8ª Edição*. McGraw Hill Brasil, 2016.
- Raman K. e Joachims T. Bayesian ordinal peer grading. In *L@S 2015 - 2nd ACM Conference on Learning at Scale*, páginas 149–156. Association for Computing Machinery, Inc, 2015.

- Reischmann T. e Kuchen H. Towards an e-assessment tool for advanced software engineering skills. In *ACM International Conference Proceeding Series*, páginas 81–90. Association for Computing Machinery, 2016.
- Rezende F. A. Características do ambiente virtual construcionista de ensino e aprendizagem na formação de professores universitários. Master's thesis, UNIVERSIDADE ESTADUAL DE CAMPINAS, 2004.
- Seshia S. A. Formal methods for engineering education. Technical report, Technical Report UCB/EECS-2015-170, EECS Department, UC Berkeley, 2015.
- Shehab A., Elhoseny M. e Hassanien A. E. A Hybrid Scheme for Automated Essay Grading Based on LVQ and NLP Techniques. *Computer Engineering Conference (ICENCO)*, 12th International, páginas 65–70, 2017.
- Shen V. R. L., Yang C.-Y., Wang Y.-Y. e Lin Y.-H. Application of High-level Fuzzy Petri Nets to Educational Grading System. *Expert Syst. Appl.*, 39:12935–12946, 2012.
- Silva T. T., Valente J. A. e Dias P. Diferentes abordagens da Educação a Distância mediada por computador e via Internet. *Revista Scitis*, 2014.
- Smith N., Thomas P. e Waugh K. Automatic Grading of Free-Form Diagrams with Label Hypernymy. In *2013 Learning and Teaching in Computing and Engineering*, páginas 136–142. IEEE, 2013.
- Striewe M. e Goedicke M. A Review of Static Analysis Approaches for Programming Exercises. *Communications in Computer and Information Science*, 439:100–113, 2014.
- Thaler T., Houy C., Fettke P. e Loos P. Automated Assessment of Process Modeling Exams: Basic Ideas and Prototypical Implementation. *Modellierung 2016 - Workshopband. Workshop zur Modellierung in der Hochschullehre (MoHoL-2016)*, 255:63–70, 2016.
- Umbreja K., Kukk V. e Jaanus M. New concepts of automatic answer evaluation in competence based learning. In *IEEE Engineering Education Conference*, páginas 922–925. IEEE, 2014.
- Valente J. A. O computador na sociedade do conhecimento. *Oea_Nied/Unicamp*, 1999.
- Vujosevic-Janicic M., Nikolić M., Tošić D. e Kuncak V. Software verification and graph similarity for automated evaluation of students' assignments. *Information and Software Technology*, 55: 1004–1016, 2013.
- Waters A. E., Tinapple D. e Baraniuk R. G. BayesRank: A bayesian approach to ranked peer grading. In *L@S 2015 - 2nd ACM Conference on Learning at Scale*, páginas 177–183. Association for Computing Machinery, Inc, 2015.
- Wazlawick R. *Metodologia de pesquisa para ciência da computação*, volume 2. Elsevier Brasil, 2017.
- Woodcock J., Larsen P. G., Bicarregui J. e Fitzgerald J. Formal methods: Practice and experience. *ACM computing surveys (CSUR)*, 41, 2009.