

Desenvolvimento e Avaliação de uma Ferramenta Interativa Baseada em Exemplos para o Aprendizado de Modelagem de Sistemas Usando Redes de Petri

Title: Development and Evaluation of an Example-Based Interactive Tool for Learning System Modeling Using Petri Nets

John Wesley Soares de Lima

UFRPE

ORCID: [0000-0003-1606-6517](https://orcid.org/0000-0003-1606-6517)

john.lima@ufrpe.br

Taciana Pontual Falcão

UFRPE

ORCID: [0000-0003-2775-4913](https://orcid.org/0000-0003-2775-4913)

taciana.pontual@ufrpe.br

Ermeson Andrade

UFRPE

ORCID: [0000-0002-9614-4492](https://orcid.org/0000-0002-9614-4492)

ermeson.andrade@ufrpe.br

Resumo

Os sistemas computacionais têm se tornado cada vez mais complexos e isso tem demandado o desenvolvimento de técnicas que garantam certas características, como alta disponibilidade e baixo custo. Assim, o interesse pela utilização de modelos computacionais tem aumentado nos últimos anos, visto que eles podem ser usados em situações em que é muito caro e até mesmo impossível testar ou medir as diversas características dos sistemas computacionais (ex.: disponibilidade ou confiança). No entanto, um grande desafio no uso de tais modelos é o seu entendimento, visto que não são intuitivos e requerem um considerável esforço de aprendizado da notação utilizada. Muitas vezes, abordagem de ensino expositivo em sala de aula, nos cursos da área de computação, é insuficiente para garantir a aprendizagem de modelagem computacional. Ferramentas educativas interativas baseadas em exemplos podem auxiliar a aprendizagem por meio de exercício e prática. Este trabalho aborda o desenvolvimento e avaliação da TryRdP, uma ferramenta web gratuita, interativa e baseada em exemplos para auxiliar na aprendizagem de modelagem computacional usando Redes de Petri (RdP), um conteúdo que integra currículos de cursos de ciência da computação, engenharia da computação e afins. As RdP são formalismos matemáticos largamente usados para modelar, analisar e projetar sistemas diversos, entre eles, sistemas computacionais. Para avaliar a eficácia da ferramenta, foram realizados experimentos com alunos e professores de cursos superiores de computação, investigando a usabilidade e a utilidade da ferramenta no apoio ao aprendizado de modelagem computacional. De forma geral, os resultados indicam que a ferramenta atende aos principais requisitos de usabilidade e foi bem avaliada em termos de sua aplicação didático-pedagógica.

Palavras-Chave: Aprendizagem, Modelagem, Rede de Petri, Ferramenta.

Abstract

Computational systems have become increasingly complex and this has required the development of techniques to guarantee certain characteristics such as high availability and low cost. Thus, the interest in the use of computational models has increased in recent years, given that they can be used in situations where it is very expensive or even impossible to test or measure the various characteristics of computer systems (e.g. availability or reliability). However, a great challenge in the use of such models is their understanding, since they are not intuitive and require a considerable effort to learn the notation used. Often, the lecture approach in the classroom, in computer science majors, is insufficient to guarantee the learning of computational modeling. Interactive educational tools based on examples can assist learning through exercise and practice. This work addresses the development and evaluation of TryRdP, a free, interactive and example-based web tool to assist the learning of computational modeling using Petri Nets, a content that integrates curricula in computer science, computer engineering and others courses. Petri Nets are mathematical formalisms widely used to model, analyze and design diverse systems, including computer systems. In order to evaluate the effectiveness of the tool, experiments were carried out with students and professors from higher education computer science majors, investigating usability

Cite as: Lima, J. W. S., Pontual Falcão, T., & Andrade, E. (2021). Desenvolvimento e Avaliação de uma Ferramenta Interativa Baseada em Exemplos para o Aprendizado de Modelagem de Sistemas Usando Redes de Petri. *Revista Brasileira de Informática na Educação*, 29, 1232-1261. DOI: 10.5753/rbie.2021.2072

and utility of the tool in supporting the learning of computational modeling. Overall, the results indicate that the tool meets the main requirements of usability and was well evaluated regarding its didactic-pedagogical application.

Keywords: Learning, Modeling, Petri net, Tool.

1 Introdução

O desenvolvimento tecnológico está associado com mudanças sociais e econômicas, além de possibilitar uma maior competição global entre países mediante a criação de novos produtos e serviços (OECD/Eurostat, 2005; Kormann & Suberg, 2021). Nesse sentido, cientistas e engenheiros têm juntado esforços na pesquisa e desenvolvimento de novos produtos e serviços, difundindo a inovação na economia global (Dietz *et al.*, 2000). Entre as diversas áreas de tecnologia existentes, a área da Tecnologia da Informação (TI) vem se destacando pelo seu rápido crescimento (U.S. Bureau of Labor Statistics, 2013, 2020). Segundo as projeções estatísticas mostradas pelo U.S. Bureau of Labor Statistics (2020), a quantidade de empregos na área de TI crescerá 11% entre os anos de 2019 e 2029 e esse crescimento é maior do que a média das outras áreas de ocupação. Vale ressaltar que a projeção mostrada não levou em consideração o impacto causado na economia global pela pandemia devida ao vírus SARS-CoV-2, que iniciou no final do ano de 2019 na cidade chinesa de Wuhan (Mo *et al.*, 2020; Mei *et al.*, 2020). No entanto, segundo a *Harvey Nash / KPMG CIO Survey 2020*, logo no início da pandemia as empresas gastaram 15 bilhões de dólares a mais por semana com TI para permitir trabalho doméstico seguro, e isso foi considerado como “um dos maiores surtos de investimento em tecnologia da história” (KPMG, 2020). Além disso, 70% dos CIOs (Chief Information Officer) acreditam que essa mudança positiva para com a área de TI será permanente (Tech Monitor, 2020).

Em paralelo a esse rápido crescimento, sistemas computacionais têm se tornado cada vez mais complexos. Sistemas como Ciberfísicos (*Cyber-Physical System* - CPS) e Internet das Coisas (*Internet of Things* - IoT) vêm sendo largamente estudados e prometem impactar a área de TI (Rossiter, 2018; Rho, *et al.*, 2016). Tais sistemas estão associados, por exemplo, à chamada “Indústria 4.0”. O termo “Indústria 4.0” surgiu na Alemanha em 2011 e está relacionado ao que há de mais moderno na área de TI aplicado à indústria (Falani *et al.*, 2020). Um sistema industrial 4.0 é complexo e tem por base quatro pilares: sistema ciberfísico, internet das coisas, computação em nuvem e *big data* (Zhou *et al.*, 2015). Além dos sistemas citados, existem também aqueles voltados para saúde e que são bem estudados atualmente. Os termos “Saúde Eletrônica” (*e-health*) e “Saúde Móvel” (*m-health*) são largamente utilizados e se referem, respectivamente, ao uso da TI (sistemas distribuídos, IoT, computação em nuvem e etc.) aplicado à saúde, e ao uso de aplicativos móveis no auxílio ao diagnóstico e na prática da medicina em geral (Cavalcanti *et al.*, 2020; Marcolino *et al.*, 2018; Farahani *et al.*, 2017).

Sistemas computacionais complexos, como os citados acima, requerem um alto nível de qualidade, desempenho, disponibilidade e devem ser capazes de contornar falhas (tolerância a falhas). Assim, a busca por tais sistemas tem motivado o desenvolvimento de métodos de modelagem. Em muitas áreas da computação como, por exemplo, sistemas embarcados e distribuídos, os ambientes computacionais não são analisados diretamente, mas sim, por meio de modelos (Trivedi *et al.*, 2012). Esses modelos podem ser usados para situações em que é muito caro e até mesmo impossível testar ou medir as diversas características dos sistemas computacionais (ex.: disponibilidade ou confiança) (Andrade *et al.*, 2017). Segundo Lima *et al.* (2021a, p. 2), “a vantagem de se usar modelos é que eles nos permitem abstrair um problema complexo de modo a torná-lo mais simples. [...] De maneira mais específica, a modelagem de sistemas é um estudo multidisciplinar que visa representar os sistemas através de modelos, de modo que os mesmos possam ser analisados sem a implementação real do sistema.” Essa

representação é obtida usando-se métodos de abstração por meio do processo de modelagem. Redes de Petri (Murata, 1989), Diagrama de Confiabilidade (Hirel *et al.*, 2000a), e Árvore de Falhas (Bennetts, 1975) são exemplos de métodos de modelagem formais que têm sido largamente usados na modelagem e análise dos mais variados tipos de sistemas computacionais. Tais métodos são apoiados por fundamentos matemáticos sólidos, que suportam sua semântica precisa, estimulam a avaliação quantitativa e fornecem suporte para verificações de propriedades qualitativas.

Entre os vários métodos de modelagem formal existentes, as RdP destacam-se como uma ferramenta gráfica e matemática eficiente na modelagem e análise de sistemas complexos (Murata, 1989). As RdP têm a capacidade de modelar com facilidade sistemas nos quais há ocorrência de paralelismo, sequenciamento, escolha não-determinística, sincronismo, compartilhamento de recursos, concorrência e exclusão mútua. Mais especificamente, adotamos neste trabalho uma extensão das redes de Petri denominada de Redes de Petri Estocásticas Generalizadas (*Generalized Stochastic Petri Nets – GSPN*) (Chiola *et al.*, 1993; Marsan *et al.*, 1983). As GSPN podem ser adotadas para análise de desempenho e dependabilidade de sistemas dinâmicos. Esse formalismo matemático também permite que métricas sejam aferidas tanto via simulação, quanto pela análise do espaço de estados. Adicionalmente, as GSPN proveem diversos recursos para modelagem, tais como funções de guarda, pesos, prioridade, entre outros.

As redes de Petri são um assunto abordado em disciplinas nos cursos de graduação e pós-graduação em computação e engenharia. No entanto, o processo de compreensão das redes de Petri, independentemente de suas extensões, nem sempre é fácil, visto que tal formalismo é baseado em representações matemáticas que podem não ser intuitivas. O modelo tradicional de aulas, baseado majoritariamente em exposições teóricas e exercícios, nem sempre é adequado quando se trata de assuntos de natureza abstrata (ex: rede de Petri), o que pode tornar o processo de aprendizado difícil e desmotivador (Costa *et al.*, 2017). Nesse sentido, para auxiliar na compreensão das redes de Petri, é possível recorrer a ferramentas educativas que estão se tornando cada vez mais interativas e customizadas, objetivando apoiar o ensino de conteúdos de natureza abstrata (Cerqueira & Silva, 2009). Com a utilização de ferramentas desse tipo, o aluno tem a possibilidade de explorar visualizações e interagir com modelos, beneficiando-se do feedback proporcionado pela ferramenta, e assim engajando-se em um processo de aprendizagem autônoma e personalizada.

Embora existam várias ferramentas para modelagem e análise de redes de Petri (Silva *et al.*, 2015), não identificamos ferramentas que auxiliem o processo de ensino e aprendizado de modelagem computacional usando tal formalismo. A literatura existente foca principalmente no mapeamento de linguagens de alto nível em redes de Petri com a finalidade de facilitar o seu uso. Este artigo apresenta TryRdP - uma ferramenta baseada na Web para auxiliar no aprendizado de modelagem de sistemas por meio de redes de Petri. A ferramenta foi desenvolvida de modo que ela possa ser usada tanto de modo autônomo quanto em um curso formal mediado por um professor. Este artigo é uma extensão de um trabalho previamente publicado no Simpósio Brasileiro de Educação em Computação (EduComp) 2021 (Lima *et al.*, 2021b), em que foram apresentados a ferramenta e o processo de desenvolvimento. O presente artigo complementa o trabalho anterior, de modo a apresentar a avaliação da ferramenta por professores e estudantes da área de computação. Essa avaliação teve a participação de 8 professores e 15 estudantes voluntários da área de TI, onde o objetivo da mesma foi investigar a usabilidade e a utilidade da ferramenta no apoio ao aprendizado de modelagem computacional.

O presente trabalho está organizado da seguinte forma: a Seção 2 apresenta conceitos teóricos fundamentais sobre redes de Petri. A seção 3 aborda ferramentas existentes relacionadas ao tema. A Seção 4 apresenta como a ferramenta TryRdP foi desenvolvida. A Seção 5 detalha o funcionamento da ferramenta. A seção 6 detalha a avaliação da ferramenta com alunos e

professores voluntários, e os resultados. Por fim, a Seção 7 conclui este trabalho e descreve trabalhos futuros.

2 Redes de Petri

As redes de Petri originaram-se da tese de doutorado de Carl Adam Petri em 1962 na Universidade de Bonn (Murata, 1984). Elas são um formalismo matemático usado para descrever sistemas que possuem características como concorrência, conflito, sincronização e exclusão mútua. Uma rede de Petri é graficamente desenhada como um multigrafo bipartido direcionado e que possui apenas dois tipos de nós: lugares e transições (representados por círculos e retângulos, respectivamente). Semanticamente, os lugares representam os recursos, estados ou variáveis do sistema, enquanto as transições representam elementos ativos da rede, isto é, as ações realizadas pelo sistema. Um arco em uma rede de Petri interliga um lugar a uma transição ou uma transição a um lugar, mas nunca ocorre interligação entre elementos semelhantes, ou seja, lugar com lugar ou transição com transição (Arteiro *et al.*, 2007).

Além dos elementos básicos de uma RdP citados acima, existem os tokens (também chamados de marcações). Os tokens são representados graficamente por círculos pequenos preenchidos, que residem em lugares e são utilizados para especificar o estado da RdP. Além disso, eles trafegam pela rede por meio dos arcos, se distribuindo na rede. A distribuição do tokens nos lugares da rede determina o estado em que o sistema se encontra em determinado momento (Murata, 1989).

A Figura 1 ilustra um exemplo de uma GSPN que é composta pelos lugares $P1$ e $P2$, por uma transição imediata $T1$, e por duas transições temporizadas $T2$ e $T3$. No lugar $P1$, existem quatro tokens associados. A diferença entre as transições temporizada $T2$ e imediata $T1$, é que a transição $T2$ demanda um tempo (ou *delay*) para ser executada, já a transição imediata $T1$ não demanda tempo (isto é, ocorre de modo imediato), e ocorre de modo prioritário em relação à $T2$ (Chiola *et al.*, 1993; Marsan *et al.*, 1984). Adicionalmente, o tempo associado às transições temporizadas (ex.: $T2$ e $T3$) é distribuído exponencialmente, que é a principal característica de uma GSPN (Murata, 1989).

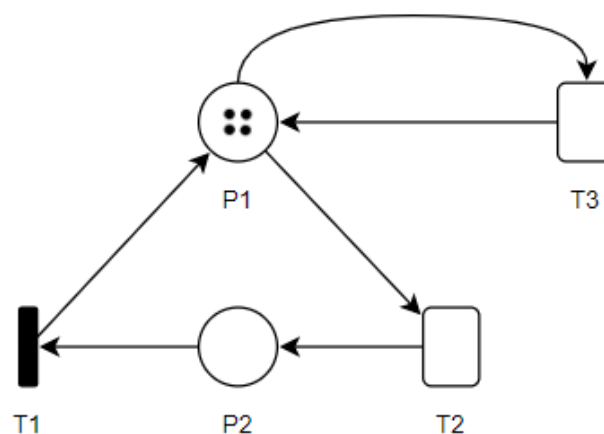


Figura 1: exemplo de uma GSPN.

Para exemplificar, o modelo GSPN da Figura 2 representa os três períodos de um dia. O dia inicia no período da manhã, depois vai para o período da tarde e, por fim, termina no período da noite. No entanto, após o período da noite, se inicia um novo dia com o período da manhã. No modelo GSPN, os elementos do sistema são representados da seguinte maneira: os períodos do dia são representados pelos lugares *Manhã*, *Tarde* e *Noite*; o token que aparece no lugar *Manhã* indica que o dia inicia no período da manhã; as transições *Período1*, *Período2* e *Período3*

representam o tempo de duração de cada um dos três períodos do dia, respectivamente, manhã, tarde e noite; por fim, os arcos representam a ordem de início de cada um dos três períodos.

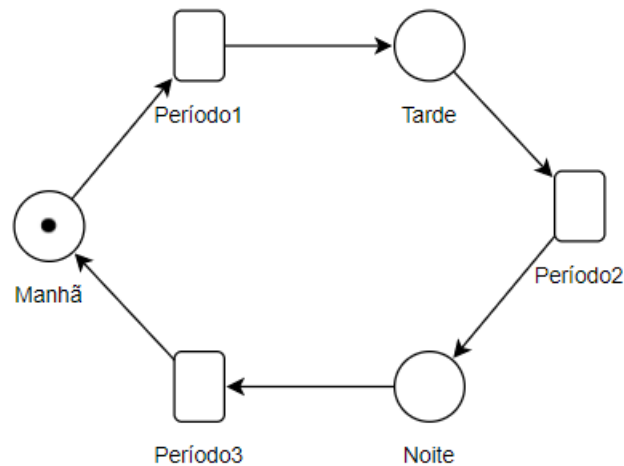


Figura 2: modelo GSPN que representa três períodos de um dia.

Como outro exemplo, o modelo GSPN da Figura 3 representa uma fila com um servidor. Esse modelo pode representar, por exemplo, uma fila de um banco com um caixa para atender clientes, ou um sistema computacional com um servidor para atender às requisições. Nesse modelo, as transições *chegada* e *servidor* representam, respectivamente, o tempo entre chegadas de clientes no sistema, e o tempo em que o cliente finaliza seu atendimento no servidor. Os lugares *fila* e *clienteatendimento* representam, respectivamente, a fila de espera de clientes, e o cliente sendo atendido pelo servidor. O lugar *limite* indica que o servidor pode atender apenas um cliente por vez. A transição imediata *T11* representa a ação do cliente de iniciar um atendimento, desde que o servidor esteja disponível.

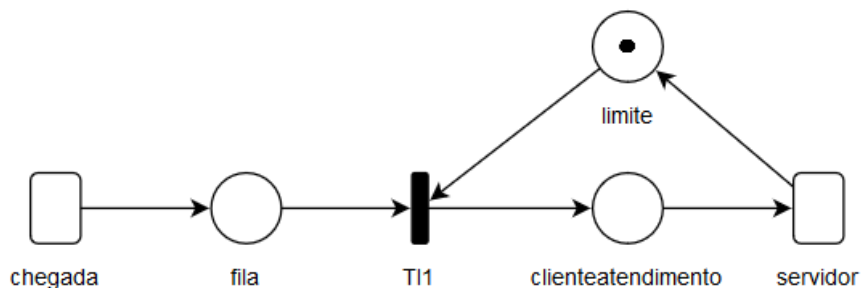


Figura 3: modelo GSPN de uma fila com um servidor.

Por fim, vale explicar sobre as métricas nas RdP. As métricas são notações usadas para representar cálculos matemáticos ou condições dos estados nas RdP (Silva, *et al.*, 2015). Os cálculos, representados pelas métricas, são realizados por ferramentas de modelagem e simulação. Por exemplo, em relação ao modelo da Figura 3, suponha que tenhamos a seguinte métrica: $(E\{\#fila\})$. Essa métrica representa o cálculo da esperança matemática da quantidade de tokens no lugar *fila*. Em outras palavras, essa métrica representa a quantidade média de clientes à espera por um atendimento na fila. Nesse mesmo modelo, também poderíamos calcular, por exemplo, a probabilidade de haver mais de um cliente na fila com a métrica: $(P\{\#fila > 1\})$.

3 Trabalhos Relacionados

Embora existam vários softwares educativos para auxiliar o aprendizado, não foram encontrados softwares educativos com objetivo de auxiliar o aprendizado de modelagem de sistemas com redes de Petri. As ferramentas existentes focam principalmente no mapeamento de linguagens de alto nível em redes de Petri com a finalidade de facilitar o seu uso para modelagem de sistemas. Nesta seção, descrevemos algumas dessas ferramentas. Vale ressaltar que a busca por tais ferramentas se deu através da *search engine Google Scholar*¹. Durante as buscas foram utilizadas palavras chaves que envolviam os temas de informática na educação e de modelagem de sistemas usando redes de Petri, por exemplo, “redes de Petri”, “ferramenta de apoio ao aprendizado” e “modelagem de sistemas”.

A ferramenta ArgoPerformance (Distefano *et al.*, 2005) é um *plugin* projetado para ser usado em uma outra ferramenta chamada ArgoUML CASE, cujo objetivo final é mapear um modelo UML (*Unified Modeling Language*) em uma rede de Petri para avaliar o desempenho de processo de desenvolvimento de software. Já a ferramenta ArgoSPE (Gómez-Martínez & Merseguer, 2006), permite o mapeamento automático de diagramas da UML anotados em uma Rede de Petri Estocástica (SPN) para avaliação de desempenho de sistemas computacionais. A ferramenta ADAPT (Rugina *et al.*, 2008) permite o mapeamento da linguagem AADL (*Architecture Analysis and Design Language*) em Rede de Petri GSPN com o objetivo de fazer avaliação de confiabilidade em processos de engenharia de sistemas. A ferramenta Calau (Andrade *et al.*, 2012) auxilia projetistas no mapeamento do diagrama de estados da SysML (*Systems Modeling Language*), anotado de acordo com o *profile* MARTE, em uma rede de Petri, com o intuito de realizar análises de caminhos críticos para o tempo de execução e o consumo de energia. O propósito dessas ferramentas é facilitar o mapeamento de representações em alto nível em modelos de RdP sem que, necessariamente, o usuário saiba modelar um sistema diretamente com o formalismo de rede de Petri. Por exemplo, se um usuário conhece a AADL, ele poderia usar a ferramenta ADAPT para mapear uma rede de Petri.

Além dessas ferramentas citadas, existem outras usadas exclusivamente para a modelagem e a análise das redes de Petri, tais como Mercury (Silva *et al.*, 2015), TimeNet (Zimmermann, 2017), Oris (Bucci *et al.*, 2010) e SPNP (Hirel *et al.*, 2000b). Vale ressaltar que tais ferramentas são bem conhecidas e são comumente usadas por pesquisadores da área de modelagem de sistemas computacionais. Embora essas ferramentas possam ser usadas para os propósitos de modelagem e análise, as mesmas não dão suporte à aprendizagem das redes de Petri. Ou seja, elas requerem que os usuários já saibam e tenham experiência em modelagem de sistemas. Assim, o uso de tais ferramentas com o propósito de modelagem e análise é um desafio para um usuário iniciante ou com pouco conhecimento em modelagem, visto que um pequeno erro na sintaxe do formalismo ou algum erro de digitação quando, por exemplo, o usuário digita as métricas, compromete toda a análise ou simulação de um modelo. Adicionalmente, quando o usuário comete algum erro na construção das redes de Petri, no geral, nenhuma mensagem de erro com feedback é apresentada, e quando é apresentado algo, a mensagem contém informações vagas sem precisão sobre qual erro o usuário cometeu. O usuário, muitas vezes, só vai saber se houve ou não um erro quando ele/ela simula o modelo e a ferramenta não retorna nenhum valor.

Percebe-se que as ferramentas encontradas que envolvem redes de Petri não possuem objetivos de ensino ou aprendizagem de modelagem de sistemas, de modo que não se caracterizam como softwares educativos (Oliveira *et al.*, 2001). A ferramenta educativa que

¹ <https://scholar.google.com/>

estamos propondo é voltada para a aprendizagem sobre redes de Petri do nível básico ao avançado, apresentando como características de software educativo (Oliveira *et al.*, 2001):

- Fundamentação pedagógica: a ferramenta segue uma abordagem instrucionista, combinando os formatos de tutorial e exercício e prática (Valente, 2000).
- Finalidade didática: tem o objetivo de levar o aluno a construir conhecimento relacionado à modelagem de sistemas usando redes de Petri.
- Interação aluno-software educativo-professor: a ferramenta foi desenvolvida de modo que ela possa ser usada tanto de modo individual quanto em um curso formal mediado por um professor.

4 Método

A ferramenta TryRdP foi desenvolvida de acordo com o fluxograma da Figura 4, com o objetivo de auxiliar o processo de ensino e aprendizado de modelagem computacional usando redes de Petri. Primeiramente, foram buscadas e analisadas ferramentas existentes para modelagem de sistemas computacionais (apresentadas na seção de trabalhos relacionados). Em seguida, os requisitos da ferramenta foram definidos. Assim, os requisitos básicos definidos foram: a ferramenta deve ser gratuita e baseada na *web*; a ferramenta deve conter exercícios de modelagem e de quebra cabeça; deve permitir o uso autônomo e interativo; apresentar *feedbacks* para os estudantes de modo a auxiliá-los na compreensão do conteúdo; apresentar uma sequência lógica/didática na apresentação dos conteúdos; e, por fim, a ferramenta deve conter elementos básicos de “gamificação” como pontuação e barra de progresso. Após isso, a arquitetura foi projetada de modo a incluir os requisitos básicos definidos. Com relação a arquitetura, a mesma é detalhada na Subseção 5.1. Vale ressaltar que a equipe que participou da definição dos requisitos e da arquitetura, bem como do desenvolvimento da ferramenta era composta por um aluno de pós-graduação em computação, uma especialista em interface de usuário e um especialista em redes de Petri.

Após a etapa da definição dos requisitos e da arquitetura deu-se início ao desenvolvimento da ferramenta. A TryRdP foi desenvolvida de modo iterativo e incremental, com implementação e testes ocorrendo de forma paralela. A TryRdP foi desenvolvida usando *PHP*, *HTML*, *CSS*, *Bootstrap framework*, *javaScript* e seis bibliotecas baseadas no *javaScript* (*jQuery*, *html2pdf*, *filesaver*, *VEX*, *CursorJS* e *MxGraph*). Como não foram encontradas ferramentas com objetivo similar, a TryRdP foi projetada essencialmente com base na fundamentação teórica sobre as RdP e as diretrizes didáticas seguidas por professores e profissionais da área. Também vale mencionar que os exemplos contidos em alguns livros como Maciel *et al.* (1996) e Trivedi (2008) serviram de inspiração para a construção do conteúdo didático da ferramenta. Adicionalmente, destacamos que a ferramenta segue o paradigma de software educacional baseado em exercício e prática. Após o desenvolvimento da ferramenta, foi feita a avaliação da TryRdP. Como essa parte é uma das principais contribuições deste trabalho, detalhamos a mesma na Subseção 4.1. A partir dos *feedbacks* dos professores e estudantes obtidos da avaliação, foram implementadas e definidas as melhorias na ferramenta. No Seção 6 são apresentados detalhes das melhorias propostas e implementadas.

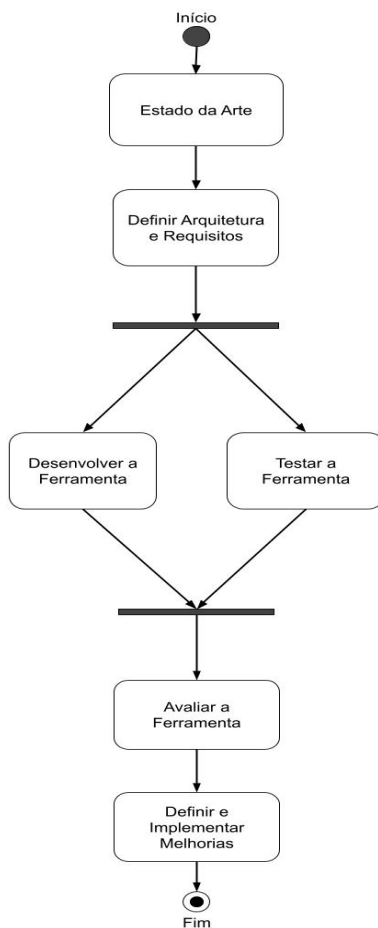


Figura 4: metodologia proposta.

4.1 Avaliação da Ferramenta

A avaliação da ferramenta foi feita com 8 professores especialistas em rede de Petri e 15 estudantes da área de TI, sendo 11 estudantes da graduação e 4 estudantes da pós-graduação. Os participantes foram recrutados por e-mail, e participaram de maneira voluntária. Aqueles que demonstraram interesse em participar receberam esclarecimentos completos sobre as etapas da pesquisa, assim como a garantia de anonimato, e tinham o direito de desistir da participação a qualquer momento sem nenhum tipo de consequência. No caso dos estudantes, não havia nenhuma consequência em seu curso de graduação ou pós-graduação, decorrente de sua participação na pesquisa e a nota obtida nos testes.

Vale ressaltar que a avaliação de um software educativo deve contemplar aspectos de usabilidade e critérios pedagógicos, como abordagens complementares (Junior & Aguiar, 2018). Assim, foram usados critérios de interface inspirados nas heurísticas de Nielsen (1994), que são um conjunto tradicional de heurísticas da área de Interação Humano-Computador, e questões didático-pedagógicas a partir da prática docente dos pesquisadores. Além disso, para avaliação de aprendizagem a partir do uso da ferramenta, foi usado o método de pré e pós-teste – esses instrumentos não foram padronizados, e sim elaborados pelos próprios pesquisadores (Raabe & Bombasar, 2020), a partir de seu conhecimento na área de estudo (redes de Petri) e docência. A significância estatística da diferença entre os resultados dos testes foi avaliada usando-se o teste não paramétrico de Wilcoxon (Triola, 2017).

A avaliação com os professores foi feita com 8 doutores especialistas na área de sistemas computacionais e modelagem de sistemas com redes de Petri, os quais participaram de maneira

voluntária. Tais professores exercem atividades de docência e pesquisa em universidades e institutos federais. *E-mails* foram enviados aos professores mostrando do que se tratava a pesquisa e perguntando se eles poderiam participar da avaliação da ferramenta. A avaliação com professores objetivou avaliar a usabilidade da ferramenta e sua aplicação didático-pedagógica. Os professores receberam um link para usar a ferramenta livremente por um período de 10 dias. Em seguida, foi disponibilizado um formulário² desenvolvido usando o *Google Forms*³, onde os professores classificaram as suas opiniões sobre a ferramenta de acordo com 13 tópicos que cobrem questões de interface e questões didático-pedagógicas (ver Quadro 1). A avaliação de cada tópico se deu por meio de escala Likert (Joshi, 2015) de cinco níveis: “péssimo”, “ruim”, “regular”, “bom” e “ótimo”. Além disso, os professores puderam escrever um *feedback* com pontos positivos ou sugerindo melhorias na ferramenta. Os 10 primeiros tópicos do Quadro 1 foram apresentados tanto aos professores quanto aos estudantes. No entanto, os 3 últimos tópicos, que abordam temas voltados ao conteúdo da ferramenta e a utilidade para o ensino, foram apresentados apenas aos professores.

Já a avaliação com os estudantes objetivou avaliar a usabilidade e o aprendizado decorrente do uso da ferramenta. Houve a participação de 15 voluntários, sendo 11 de graduação e 4 de pós-graduação em áreas relacionadas a tecnologia da informação. Os estudantes foram recrutados de dois modos: i) enviando *e-mails* para grupos de graduação explicando sobre a pesquisa e chamando por voluntários; ii) solicitando os professores que aceitaram participar da pesquisa que convidassem seus orientandos de graduação ou pós-graduação.

Houve duas formas de avaliação com os estudantes: i) uma avaliação de aprendizagem sobre o conteúdo apresentado na ferramenta, por meio de um teste prévio e um teste posterior ao uso da ferramenta; e ii) uma avaliação sobre a usabilidade da ferramenta similar ao que foi feito com os professores. Assim, a avaliação ocorreu em 4 etapas seguindo a seguinte ordem: 1º) um teste prévio com conteúdo sobre modelagem usando as redes de Petri foi aplicado antes dos estudantes usarem a ferramenta TryRdP; 2º) após o teste, os estudantes foram apresentados à TryRdP e puderam usá-la livremente por um período de 10 dias; 3º) em seguida, um outro teste sobre redes de Petri foi aplicado; 4º) por fim, os estudantes foram convidados a darem sugestões e opiniões sobre a ferramenta, onde responderam um formulário de avaliação contendo 10 tópicos (Quadro 1). Tanto os testes quanto o formulário foram implementados com o uso do *Google forms*, e enviados por *e-mail* aos estudantes. A partir dos *feedbacks* dos professores e estudantes, foram implementadas e definidas as melhorias na ferramenta.

² Testes e formulários para professores e estudantes disponível em:

https://www.dropbox.com/sh/12gmppwx5rvzi6e/AAB94_TmRtk3ZZjuBU-q_zMFa?dl=0

³ <https://www.google.com/forms/about/>

Quadro 1: tópicos usados na avaliação de [professores](#) e [estudantes](#).

10 tópicos de avaliação apresentados aos professores e estudantes (inspirados nas heurísticas de Nielsen (1994))	
Tópico 1	Facilidade de uso da ferramenta.
Tópico 2	Visual da ferramenta.
Tópico 3	Tempo de resposta da ferramenta (demora ou rapidez na apresentação dos resultados).
Tópico 4	Liberdade (ou flexibilidade) na navegação.
Tópico 5	Visualização do progresso (o usuário consegue perceber o quanto falta para concluir o curso? - pontuação e progresso).
Tópico 6	Clareza dos textos explicativos.
Tópico 7	Clareza das mensagens de erro ou alerta.
Tópico 8	Satisfação de uso.
Tópico 9	Eficácia para a aprendizagem.
Tópico 10	Feedback a erros nos exercícios.
3 tópicos de avaliação apresentados apenas aos professores (didático-pedagógicos)	
Tópico 11	Completude do conteúdo.
Tópico 12	Utilidade para uso no ensino formal (sala de aula).
Tópico 13	Utilidade para uso autônomo pelos estudantes.

Aqui, vale mostrar alguns detalhes sobre a avaliação da aprendizagem. Primeiramente, as notas do primeiro e segundo teste (com escala de 0 a 100) foram comparadas mediante o uso dos parâmetros de estatística descritiva: média e mediana. Neste caso, como há dois grupos de estudantes com nível de formação distinta (graduação e pós-graduação), a comparação foi feita de três modos: com todos os 15 estudantes, apenas com os 11 estudantes de graduação e apenas com os 4 estudantes de pós-graduação. Vale ressaltar que 4 estudantes da graduação e 4 estudantes da pós-graduação tiveram algum contato com o assunto de redes de Petri antes de serem convidados para participarem da avaliação. Desse modo, a comparação também foi feita com o grupo de 7 estudantes que não tiveram contato com o assunto de redes de Petri antes de serem convidados para a avaliação. Por fim, foi feito um estudo de estatística inferencial (teste não paramétrico de Wilcoxon) a fim de identificar se o uso da ferramenta contribuiu para o aumento das notas dos estudantes.

A respeito dos testes, eles continham 10 questões de múltipla escolha e abordavam os assuntos apresentados na ferramenta. No entanto, essas questões são diferentes dos exercícios mostrados na ferramenta. As questões foram elaboradas de modo que elas envolvessem mais de um assunto apresentado na ferramenta. Ou seja, para que o estudante respondesse uma questão ele deveria dominar mais de um assunto. Acreditamos que o modo como foram feitas as questões (diferentes dos exercícios da ferramenta e exigindo o entendimento de mais de um assunto) não gera algum viés causado pelos próprios testes. Na seção 6 são apresentados os resultados das avaliações.

5 A Ferramenta TryRdP

A ferramenta TryRdP foi desenvolvida de modo a ter conteúdos didáticos relacionados à modelagem usando RdP, com exemplos de aplicações reais e exercícios a serem realizados em um editor de gráfico. A contextualização provida pelos exemplos de situações reais permite uma aprendizagem mais significativa, facilitando a compreensão dos estudantes ao situar os

conteúdos em situações familiares. Assim, primeiramente iremos apresentar a arquitetura adotada. Em seguida, iremos detalhar o funcionamento da ferramenta.

5.1 Arquitetura da Ferramenta TryRdP

A Figura 5 mostra o fluxograma da arquitetura adotada. A partir da interface gráfica, o fluxograma apresenta a execução de cada módulo de acordo com o sentido das setas. Segue a descrição de cada módulo:

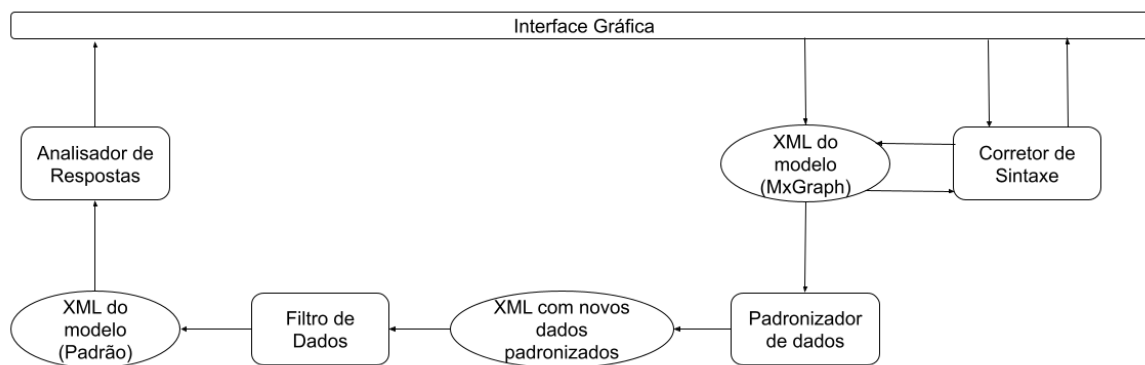


Figura 5: arquitetura da TryRdP.

- Módulo 1 - Interface Gráfica:** nesse módulo, o usuário da ferramenta entra em contato com a interface gráfica que será apresentada na próxima subseção. Esse módulo foi desenvolvido com o uso da biblioteca gráfica *MxGraph*. A biblioteca *MxGraph* possibilita criar um arquivo em XML e o salva na memória temporária do computador cliente. Esse arquivo é usado para armazenar dados dos objetos gráficos introduzidos pelo usuário em tempo real. A biblioteca possui um sistema que armazena dados gerais sobre os objetos gráficos e suas relações, por exemplo, dados sobre a posição (em *pixel*) dos elementos no plano da interface gráfica e o seu *id* (um número único que identifica cada objeto). Em termos de RdP, esse sistema pode ser programado para armazenar dados sobre a rede e sobre os elementos da rede, ou seja, ele armazena quais os elementos da rede foram usados (transições, lugares ou arcos) e as propriedades desses elementos. Esse arquivo XML será utilizado nos Módulos 2 e 3 para obtenção de informações a respeito da RdP.
- Módulo 2 - Corretor de Sintaxe:** aqui vale destacar que os Módulos 1, 2 e 3 são executados de modo conjunto e, também, são executados em tempo real. O Módulo 2 foi desenvolvido para ser um corretor da sintaxe da RdP. Desse modo, conforme o usuário vai montando a rede na interface gráfica, o Módulo 2 vai sendo executado e faz o trabalho de checar se a sintaxe da RdP está sendo corretamente colocada. Esse módulo também é responsável pelas mensagens indicando os erros de sintaxe para que os usuários corrijam. O Módulo 2 também é importante pelo fato de que é necessário que a RdP esteja com a sintaxe correta antes da padronização dos dados que é executada pelo Módulo 3.
- Módulo 3 - Padronizador de Dados:** esse módulo é o responsável por obter o arquivo XML gerado pelo *MxGraph* e criar um padrão de armazenamento de dados. Embora o arquivo XML gerado pelo *MxGraph* traga as relações e propriedades dos elementos, ele não é amigável para os propósitos do uso da RdP devido à

complexidade desse formalismo. Isso ocorre, pois, esse arquivo XML não armazena todos os detalhes existentes nas relações dos elementos de uma RdP. O arquivo XML gerado pelo *MxGraph* armazena dados sobre os objetos tomando por base a origem e a chegada de um determinado arco. As RdP são complexas e as relações binárias do XML gerado pelo *MxGraph* não expressam de maneira acurada esse formalismo. Desse modo, o Módulo 3 traz uma forma mais amigável de armazenamento dos dados dos elementos e isso inclui adicionar novas *tags* com novos dados ao arquivo XML gerado pelo *MxGraph*. Essas *tags* referenciam dados que trazem informações como, por exemplo, os pesos ou outras propriedades das transições imediatas conectadas a um lugar ou a diversos lugares e suas relações. A padronização de dados é importante para que seja feita uma análise das repostas (Módulo 5) de um modo mais simples.

- **Módulo 4 - Filtro de Dados:** esse módulo é executado logo após o usuário enviar a sua resposta para a análise. Esse módulo cria um novo arquivo XML contendo apenas uma parte das *tags* adicionadas no Módulo 3. O Módulo 4 apenas guarda as informações relevantes para a RdP e exclui as outras. Informações como posição (em *pixel*) dos elementos no plano da interface gráfica são excluídas. Além disso, uma parte dos dados padronizados, que foram adicionados no módulo 3, também podem ser excluída. Esse novo arquivo XML será usado para a análise das repostas (Módulo 5).
- **Módulo 5 - Analisador de Respostas:** esse módulo é o responsável por verificar se a resposta enviada pelo usuário está correta ou não e, conseqüentemente, atribuir pontos caso o usuário acerte a resposta. A partir do arquivo XML gerado no Módulo 4, são feitas a análise e comparação dos dados do arquivo XML contendo a resposta do usuário com o arquivo XML da resposta correta. Arquivos XML das repostas corretas dos exercícios ficam armazenados no servidor e são utilizados assim que o Módulo 5 é executado.

A seguir iremos detalhar a interface gráfica da TryRdP, bem como o seu funcionamento básico, visto que são partes fundamentais da ferramenta.

5.2 Funcionamento Básico da TryRdP

A TryRdP está dividida em três módulos obrigatoriamente sequenciais: introdução, modelagem básica e modelagem avançada (ver Figura 6). Em cada módulo, são apresentados conteúdos teóricos sobre o formalismo e, posteriormente, exercícios correspondentes. Nos exercícios, a ferramenta TryRdP verifica se o modelo criado pelo estudante está correto, fornecendo feedback imediato. Cada exercício possui uma pontuação de acerto, de modo que o estudante precisa atingir uma quantidade mínima de pontos para poder passar para outros módulos da ferramenta, que abordam tópicos mais avançados. O aluno pode tentar fazer o exercício quantas vezes quiser, ou ver a resposta, caso não consiga responder à questão. Porém, caso o estudante opte por ver a resposta da questão, ele não receberá a pontuação correspondente.

TOTAL DE PONTOS: 3
PROGRESSO GLOBAL: 7.5%

RESETAR PONTOS E VOLTAR AO INICIO

O que é o TryRdP?

CONTEÚDO

- 1-Introdução
- 2-Modelagem Básica
- 3-Modelagem Avançada

O que é o TryRdP?

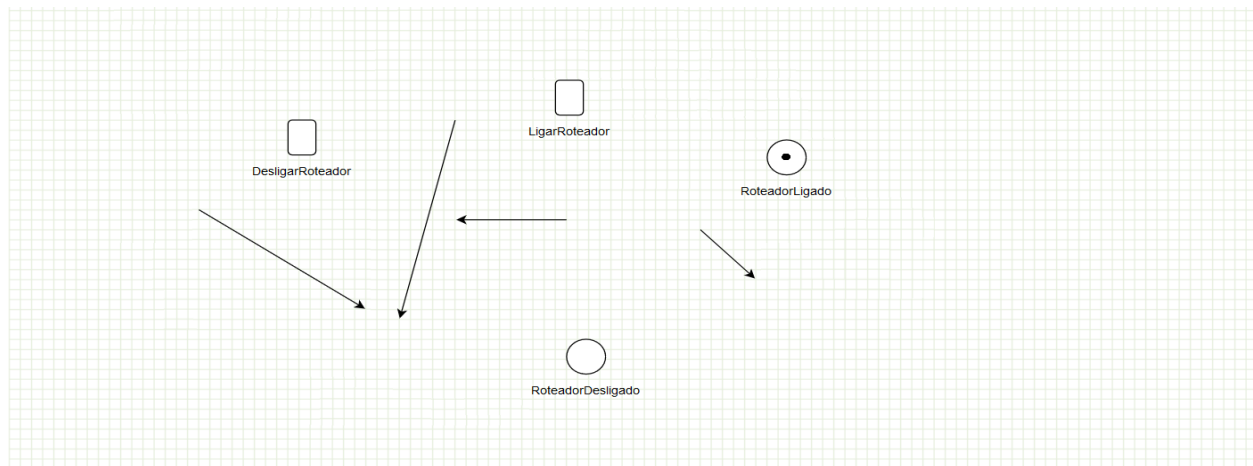
TryRdP é uma ferramenta web gratuita desenvolvida na [redacted]. É uma ferramenta interativa baseada em exemplos, que foi desenvolvida com o objetivo de ajudar os estudantes a compreenderem o funcionamento de um formalismo matemático chamado de rede de Petri. A ferramenta possui conteúdos didáticos e exercícios, permitindo que os alunos aprendam através de exemplos práticos com aplicações reais. O nome TryRdP é um nome composto, onde "Try" vem do inglês e tem aqui o sentido de "pôr em prática", enquanto RdP vem de rede de Petri.

O conteúdo está dividido em três tópicos: Introdução, Modelagem Básica e Modelagem Avançada. Inicialmente são introduzidos os conceitos gerais de uma rede de Petri. Em seguida, são apresentados tipos de redes de Petri específicas como a rede de Petri estocástica. Cada tópico contém explicações dos tópicos abordados, bem como exemplos práticos usando o formalismo das RdP. Os exercícios podem ser de dois tipos: exercícios de "quebra-cabeça" e exercícios de modelagem. Os tipos de exercícios variam de acordo com o assunto. No entanto, o usuário apenas poderá passar de um tópico para outro (ex: Introdução para Modelagem Básica) após responder os exercícios ao final de cada tópico. Além disso, o usuário terá que obter uma pontuação total mínima em cada tópico para poder passar para o próximo tópico. Ao final do último tópico, se o usuário obtiver a pontuação mínima, ele/ela receberá um certificado de conclusão. Veja as regras de pontuação apresentadas abaixo. Observação: o progresso ficará salvo indefinidamente na memória local do navegador até você clicar em "RESETAR PONTOS E VOLTAR AO INICIO" para apagar os dados.

Figura 6: menu com os três módulos (barra vertical à esquerda).

Existem na ferramenta dois tipos de exercícios: quebra cabeça e modelagem. Nos exercícios de quebra cabeça, o aluno deve conectar os arcos as transições e lugares (arrastando os elementos com o mouse), de modo a representar o modelo do sistema proposto. Os exercícios de quebra-cabeça são mais simples, pois fornecem todos os elementos necessários para o aluno montar a rede de Petri que modela o sistema apresentado – basta descobrir como conectá-los. Assim, a ferramenta é baseada no modelo de aprendizagem por prática de exercícios por tentativa e erro, mas com espaço para exploração de ideias até chegar à solução ao longo desse processo, e de forma alinhada à concepção de que o erro faz parte do processo de aprendizagem.

A Figura 7 mostra um exemplo de exercício de quebra cabeça, com o seguinte enunciado: *“Monte uma rede de Petri que representa o comportamento de ligar e desligar de um roteador. Os lugares abaixo representam estados do roteador que podem ser ligado ou desligado. Inicialmente, o roteador está ligado, de modo que existe um token no lugar ‘RoteadorLigado’ indicando tal condição inicial. As transições representam as ações de desligar ou ligar o roteador. Monte o modelo de modo que, após o roteador ser desligado, ele possa ser ligado novamente (e vice-versa), formando uma rede cíclica com arcos apontando para um único sentido”*. Como se pode observar pelos botões na parte inferior da Figura 7, o aluno poderá responder à questão ou ver a resposta. Caso o aluno não acerte a resposta, ao clicar no botão “Enviar Resposta e Avançar”, uma mensagem de erro irá aparecer na tela (ver Figura 8). O aluno poderá corrigir o modelo e submeter novamente a resposta da questão. Vale ressaltar que, como nos exercícios de quebra cabeça os elementos já são dados, os feedbacks dos erros são mais simples do que nos exercícios de modelagem. Nos exercícios de modelagem os feedbacks são mais detalhados e específicos.



Voltar

Ver Resposta

Enviar Resposta e Avançar

Figura 7: exemplo de exercício de quebra cabeça.

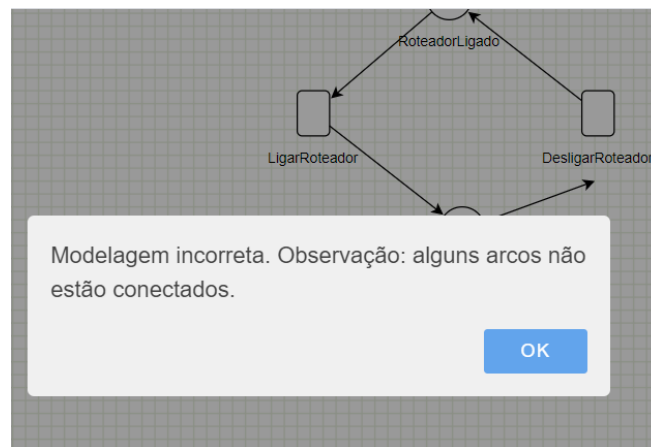


Figura 8: mensagem indicando o erro cometido.

Os exercícios de modelagem são mais complexos, pois o estudante deve selecionar os elementos para modelar o sistema (usando os botões do lado esquerdo da Figura 9), ou seja, esses elementos não são apresentados ao estudante como no quebra-cabeça. Ao clique no botão, o respectivo elemento é exibido no editor gráfico, para manipulação por parte do estudante. Na Figura 9, os botões de cima para baixo representam, respectivamente, transição temporizada, transição imediata, lugar, arco, arco inibidor e métrica. O aluno pode remover ou adicionar os elementos das RdP ao editor gráfico quantas vezes quiser. Além disso, diferentemente dos exercícios de quebra cabeça, nos exercícios de modelagem os elementos possuem propriedades, e de acordo com o enunciado do exercício, o aluno deverá adicionar valores às propriedades de alguns dos elementos. A Figura 10 mostra as propriedades de uma transição temporizada.

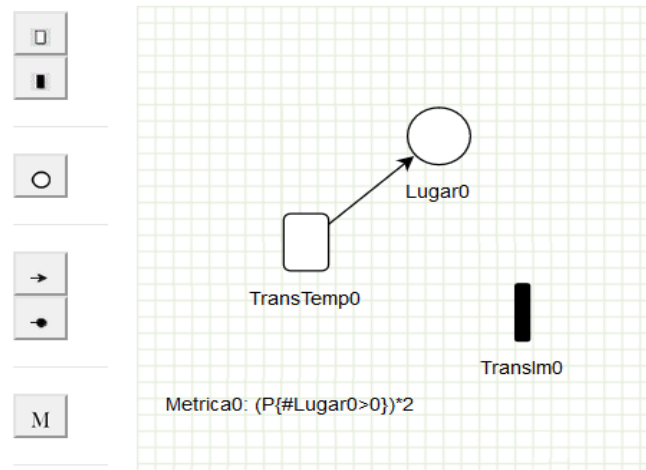


Figura 9: exemplo de exercício de modelagem.

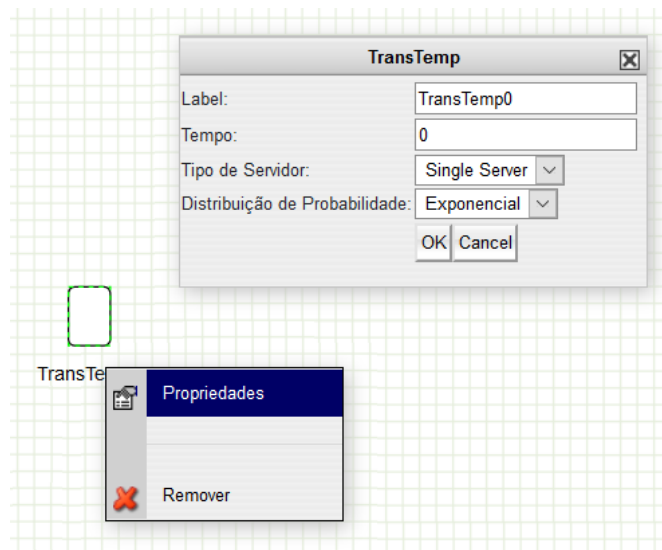


Figura 10: propriedades de uma transição temporizada em um exercício de modelagem.

Os exercícios de modelagem também possuem recursos que corrigem certos erros em tempo real, isto é, enquanto o aluno está fazendo o exercício. Por exemplo, a Figura 11 mostra um erro que foi a tentativa de conectar duas transições. Nas RdP, uma transição só pode se conectar com um lugar e vice-versa. Além desses erros, existem os erros relacionados à digitação incorreta das propriedades dos elementos e das métricas do modelo. A Figura 12 mostra um erro em que na métrica ($E\{\#Lugar1\}$), foi digitado o nome de um lugar que não existe. Note que nesse caso a métrica correta seria ($E\{\#Lugar0\}$), visto que o nome correto do lugar é *Lugar0*. Vale ser ressaltado que essas correções servem para auxiliar os estudantes quanto ao uso correto das RdP, permitindo que eles foquem no modelo conceitual, e facilite o uso futuro de ferramentas computacionais de análises e simulação das RdP, como o Mercury (Silva *et al.*, 2015) ou o TimeNet (Zimmermann, 2017).

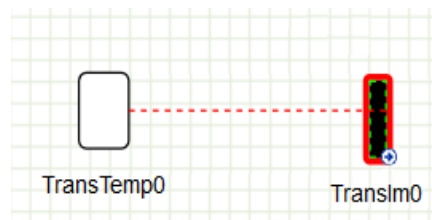


Figura 11: alerta de erro relacionado ao formalismo das RdP.

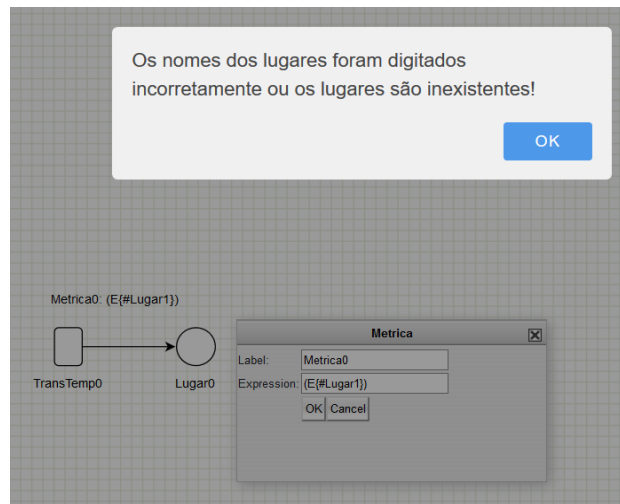


Figura 12: alerta de erro de digitação.

O nível e o tipo dos exercícios variam de acordo com o progresso do estudante. Nos dois primeiros módulos (introdução e modelagem básica), os exercícios são do tipo quebra-cabeça e modelagem. No último módulo (modelagem avançada), os exercícios são apenas de modelagem. No entanto, apenas para os exercícios de modelagem, os elementos que compõem a rede de Petri e as propriedades de certos elementos variam conforme o progresso do estudante. Por exemplo, na Figura 13, são mostradas as propriedades de uma transição antes de introduzirmos o conceito de uma transição temporizada. Esse conceito só é apresentado no módulo 2 da ferramenta. Após apresentar esse novo conceito, novos elementos são adicionados à transição (agora chamada de transição temporizada), conforme apresentado na Figura 10. Do mesmo modo, elementos das redes de Petri são mostrados conforme o progresso do aluno. Por exemplo, na Figura 14 é mostrado apenas três elementos que compõem uma rede de Petri no módulo 1: transição, lugar e arco. Já no módulo 3, que já aborda arcos inibidores, transições imediatas e métricas, esses novos elementos são mostrados (ver Figura 9).

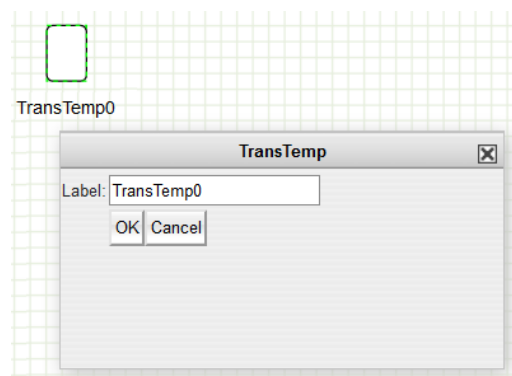


Figura 13: propriedades da transição apresentada na seção 1.

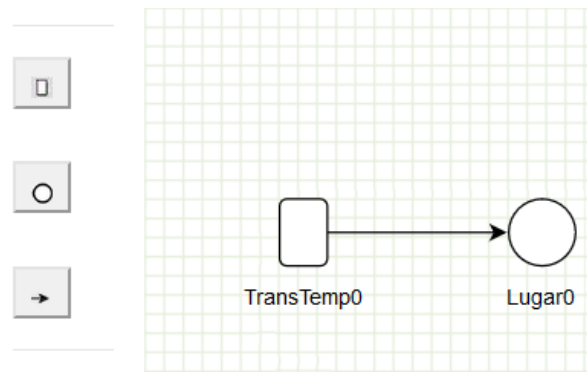


Figura 14: elementos nos exercícios de modelagem na seção 1.

Para exemplificar um exercício de modelagem de nível avançado (ver Figura 15), considere o seguinte enunciado: “Suponha que dois processadores desejam acessar uma memória compartilhada. Ambos têm o mesmo comportamento. Eles trabalham localmente por algum tempo, depois solicitam o acesso à memória compartilhada, acessam a memória e finalmente liberam a memória. O processador que estiver acessando a memória não pode fazer uma nova solicitação de acesso sem que antes tenha terminado o seu acesso e liberado a memória compartilhada. A memória compartilhada é um recurso exclusivo. Isto é, enquanto um processador acessa a memória compartilhada, o outro tem que esperar até que o primeiro tenha finalizado seu acesso. Assumimos que, se a memória não está em uso, um processador pode acessar imediatamente. Adicionalmente, assumimos que, em média, o processador 1 solicita acesso a memória a cada 25 nanosegundos, enquanto o processador 2 solicita acesso a cada 20 nanosegundos. Também, em média, o processador 1 gasta 5 nanosegundos acessando a memória. Já o processador 2 gasta, em média, 5.5 nanosegundos. Construa uma rede de Petri que representa esse sistema. Além disso, escreva três métricas que calculam: i) a taxa média (em porcentagem) de utilização geral da memória compartilhada; ii) a taxa média (em porcentagem) de utilização da memória pelo processador 1; iii) e a taxa média (solicitação/nanosegundo) que o processador 2 solicita acesso a memória.” A resposta desse exercício é mostrada na Figura 15 e com as seguintes métricas seguindo o enunciado, respectivamente: i) $Metrica1: (P\{\#memoria=0\}) * 100$; ii) $Metrica2: (P\{\#acesso1 > 0\}) * 100$; iii) $Metrica3: (P\{\#acesso2 > 0\}) / 5.5$. Observe que o exercício de nível mais avançado requer que o estudante já tenha domínio dos conceitos fundamentais sobre rede de Petri, diferentemente do exercício mostrado na Figura 7 da seção de introdução da ferramenta TryRdP.

Além disso, conforme já mencionado, quando um aluno responde e acerta uma questão ele recebe uma pontuação e, dependendo do total de pontos, o aluno pode seguir para o próximo módulo. No entanto, se o aluno não conseguir o número mínimo de pontos para seguir para o próximo módulo, ele receberá uma mensagem indicando que ele pode fazer os exercícios que ele não concluiu e obter a pontuação mínima para passar de módulo. Ao final do último módulo, se o aluno conseguir atingir a pontuação mínima, ele/ela obterá um certificado de participação.

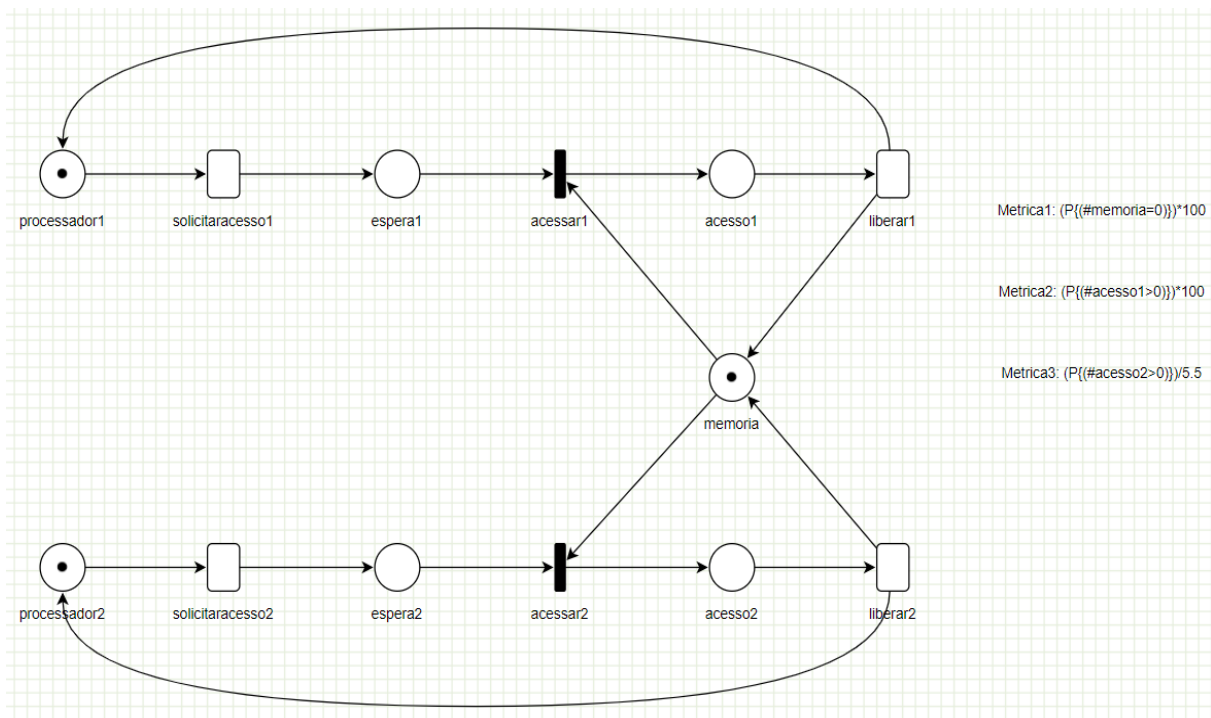


Figura 15: exemplo de exercício de modelagem de nível avançado.

6 Resultados da Avaliação da Ferramenta

Abaixo segue as descrições dos resultados das avaliações com professores e estudantes, conforme detalhadas na Subseção 4.1.

6.1 Resultados da Avaliação com Professores

A Figura 16 mostra o resultado das avaliações dos professores sobre cada um dos 13 tópicos (Quadro 1). O gráfico é dividido em três partes: esquerda, meio e direita. A parte da esquerda mostra a proporção em porcentagem da quantidade de avaliações do tipo “ruim” (cor amarelo) ou “péssimo” (cor amarelo escuro). A parte do meio mostra a proporção da quantidade da avaliação “regular” (cinza). Já a parte da direita mostra a proporção da quantidade de avaliações do tipo “bom” (cor verde claro) ou “ótimo” (cor verde escuro). Além disso, as avaliações dos tópicos são mostradas em ordem decrescente: as que têm a maior soma de avaliações como “bom” e “ótimo” aparecem na parte superior e as que têm soma de avaliações mais negativas aparecem na parte inferior. Como pode ser observado, as avaliações foram muito boas e em nenhum dos tópicos houve as avaliações “péssimo” ou “ruim”.

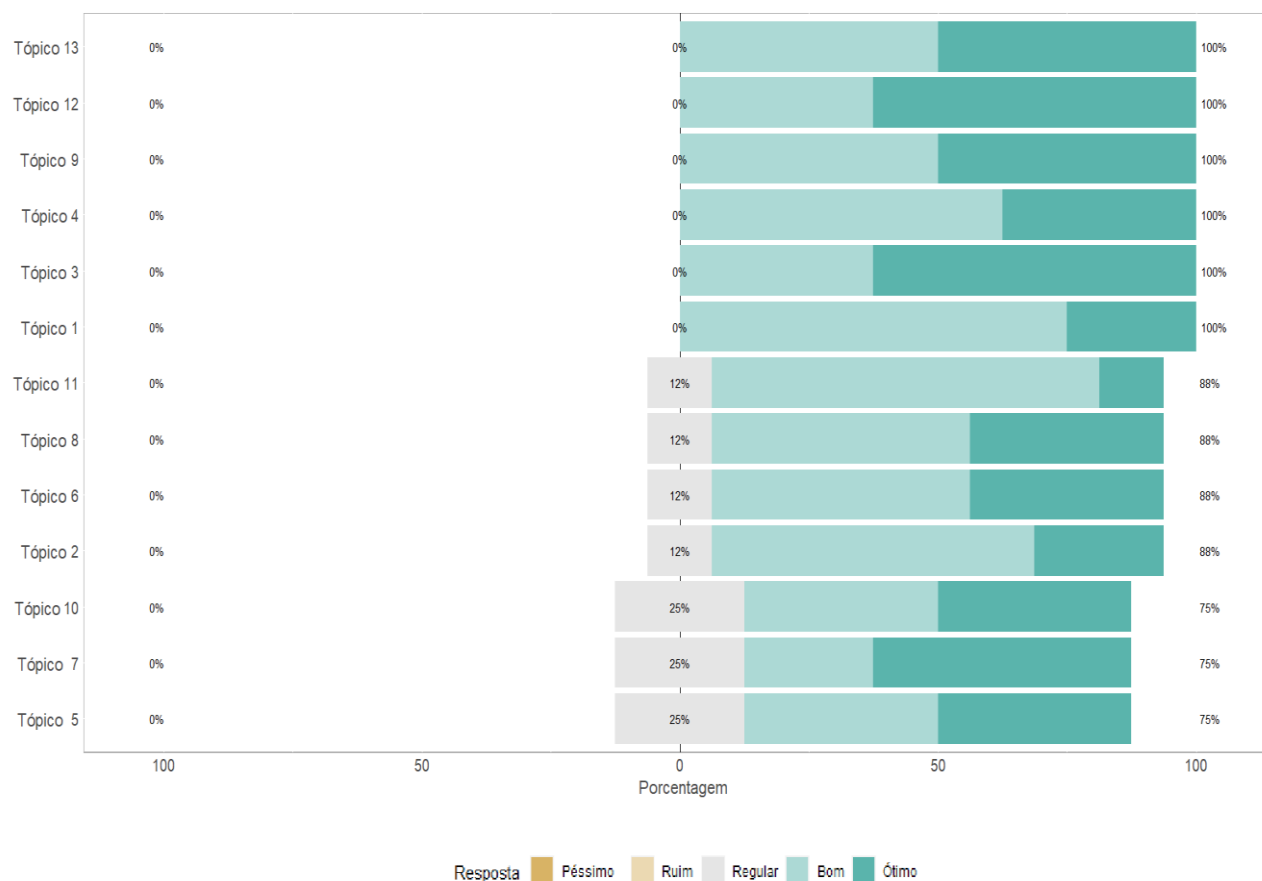


Figura 16: Resultados da avaliação feita pelos professores (ver descrição dos tópicos no [Quadro 1](#)).

Como se vê na Figura 16, os tópicos 1, 3, 4, 9, 12, e 13 foram os mais bem avaliados. Em especial, vale destacar o tópico 9, que trata da eficácia para a aprendizagem, aliado ao 12 e 13, que tratam da utilidade da ferramenta tanto para uso autônomo quanto mediado, indicando uma ótima percepção dos professores quanto à sua utilidade na aprendizagem. Nos campos abertos, os professores afirmaram que a ferramenta é um excelente instrumento de apoio ao ensino de modelagem de sistemas usando rede de Petri. Em particular, destacaram o tipo de exercício de quebra cabeça, que segundo eles, ajuda o estudante iniciante a compreender o processo de modelagem.

Por outro lado, os tópicos 5, 7 e 10 foram os menos bem avaliados, indicando que a percepção do progresso e o feedback provido pela ferramenta, embora considerados bons, poderiam melhorar. Esse é um aspecto importante visto que o feedback é essencial para a aprendizagem, principalmente nas situações de uso autônomo, em que a ferramenta será a fonte principal para compreensão do erro, e não há mediação de um professor para fornecer explicações. A seguir, são mostradas opiniões e sugestões de melhorias dadas pelos professores nos campos abertos, no formulário de avaliação. Nos quadros 2, 3, 4, 5 e 6, as sugestões e opiniões são apresentadas agrupadas por tema (foram omitidas sugestões de melhorias linguísticas, correção de erros ortográficos e elogios). Para cada opinião ou sugestão, apresentamos um comentário discutindo a questão, e indicando se foram feitos ajustes ou registradas como melhorias futuras.

Quadro 2: sugestões/opiniões dos professores quanto à facilidade e satisfação de uso da ferramenta.

Sugestões/opiniões dos professores quanto à facilidade e satisfação de uso da ferramenta	
Opinião	<p>Alguns professores consideraram um pouco trabalhosa a ação de conectar os arcos com o movimento de arrastar (usando mouse ou <i>trackpad</i>), nos exercícios de quebra cabeça.</p> <p>Comentário dos autores: a forma de interação da versão atual está limitada pelas tecnologias usadas.</p> <p>Melhorias futuras: outras tecnologias devem ser estudadas para melhorar a usabilidade na ação de conectar os elementos.</p>
Sugestão	<p>Adicionar o botão <i>tokengame</i> nos exercícios de modelagem.</p> <p>Comentário dos autores: o <i>tokengame</i> é um recurso que gera uma animação no modelo de RdP construída pelo usuário, sendo uma visualização bastante interessante, e comum em ferramentas de simulação e análise (ex.: Mercury). O <i>tokengame</i> não foi implementado dada a sua complexidade e limitação de tempo desta pesquisa.</p> <p>Melhorias futuras: outras tecnologias devem ser estudadas para permitir a implementação do <i>tokengame</i>.</p>

Quadro 3: sugestões/opiniões dos professores quanto à visualização do progresso.

Sugestões/opiniões dos professores quanto à visualização do progresso	
Sugestão	<p>Alguns professores indicaram que a visualização do progresso não estava muito evidente.</p> <p>Alteração realizada: na nova versão da interface, essa questão foi melhorada, com uma barra colorida.</p>

Quadro 4: sugestões/opiniões dos professores quanto à aprendizagem do conteúdo e nível de dificuldade.

Sugestões/opiniões dos professores quanto à aprendizagem do conteúdo e nível de dificuldade	
Opinião	<p>O exercício 11 foi considerado um pouco difícil para quem está iniciando os estudos em rede de Petri.</p> <p>Comentário dos autores: estruturalmente o modelo desse exercício é simples. Ele contém apenas duas transições temporizadas, duas transições imediatas e três lugares. No entanto, o raciocínio que o usuário deve ter para se chegar no modelo requer um pouco de trabalho. Porém, o modelo desse exercício foi escolhido e mantido por ser usado na modelagem de sistemas computacionais. Em caso de dificuldades, o estudante poderá ver a resposta, não ficando preso por conta do exercício. A ferramenta foi construída se baseando na concepção de que o erro faz parte do processo de aprendizagem.</p>

Quadro 5: sugestões/opiniões dos professores quanto à completude do conteúdo.

Sugestões/opiniões dos professores quanto à completude do conteúdo	
Sugestão	<p>Alguns professores sugeriram como trabalho futuro a tradução para o inglês e espanhol da ferramenta.</p> <p>Comentário dos autores: de fato, almejamos no futuro traduzir os textos da ferramenta para outros idiomas.</p> <p>Melhorias futuras: tradução da ferramenta para inglês e espanhol.</p>
Sugestão	<p>Alguns professores sugeriram adicionar referências de artigos em formato de <i>links</i> para que o estudante, caso ele deseje, busque informações mais aprofundadas.</p> <p>Alteração realizada: foram adicionadas referências conforme a sugestão.</p>
Sugestão	<p>Incluir vídeos com um professor explicando alguns conceitos, além do texto.</p> <p>Comentário dos autores: pretendemos fazer isso no futuro, com o auxílio de outros professores.</p> <p>Melhorias futuras: incluir vídeos explicativos curtos para diversificar o formato dos conteúdos.</p>
Sugestão	<p>Abordar outros tipos de rede de Petri.</p> <p>Comentário dos autores: o assunto sobre as redes de Petri é bastante extenso. Por limitação de tempo, o escopo dessa pesquisa limitou-se a GSPN.</p> <p>Melhorias futuras: estender o conteúdo coberto, incluindo outros tipos de redes de Petri.</p>

Quadro 6: sugestões/opiniões dos professores quanto à clareza das mensagens de erro e feedbacks.

Sugestões/opiniões dos professores quanto à clareza das mensagens de erro e feedbacks	
Opinião	<p>No exercício 4, quando a resposta é submetida de modo incorreto, a mensagem de erro que é mostrada é genérica.</p> <p>Comentário dos autores: o exercício 4 é do tipo quebra cabeça, em que os elementos são mostrados e devem ser organizados pelo estudante. Como os exercícios de quebra cabeça são mais simples que os de modelagem, as mensagens de erro são mais genéricas, para evitar que a resposta fique evidente na mensagem.</p> <p>Melhorias futuras: considerando-se a importância das mensagens de feedback em ambientes educacionais, deve ser feito um estudo focado em como melhorar as mensagens de forma a apoiar o aprendizado de maneira mais eficaz.</p>
Opinião	<p>Alguns professores notaram que em alguns dos exercícios mais complexos de modelagem, a precisão das mensagens de erros diminui.</p> <p>Comentário dos autores: nos exercícios de modelagem, as mensagens de erros são mais precisas que nos exercícios de quebra cabeça. No entanto, na parte de modelagem avançada, quando é abordado o assunto de métricas e função de guarda, a precisão tende a diminuir. Isso ocorre, pois, a ferramenta trabalha não apenas com o modelo gráfico, mas também, com reconhecimento de <i>strings</i> e caracteres escritos nas métricas, funções de guarda e <i>labels</i> dos elementos da rede. Para realizar tal reconhecimento, a ferramenta usa estruturas complexas de “<i>if-else</i>” e “<i>switch-case</i>”. Tais estruturas apresentam limitações.</p> <p>Melhorias futuras: técnicas mais avançadas de inteligência artificial podem ser utilizadas para que aumentar a precisão das mensagens.</p>

6.2 Resultados da Avaliação com Estudantes

A seguir serão apresentados os resultados da avaliação de aprendizagem com os estudantes. As Tabelas 1, 2, 3 e 4 mostram os parâmetros estatísticos descritivos e o intervalo (mínimo e máximo) das notas dos testes realizados antes e depois do uso da ferramenta, respectivamente, de todos os 15 estudantes, os 11 estudantes da graduação, os 7 estudantes da graduação que não tiveram contato com o assunto de redes de Petri antes de serem convidados e os 4 estudantes da pós-graduação. Como mostrado nas tabelas, os estudantes obtiveram progresso nas notas. Na comparação realizada com todos os 15 estudantes (ver Tabela 1), a média das notas teve um aumento de quase 160% enquanto a mediana teve um aumento de 700%. Comparando os grupos de estudantes da pós-graduação e da graduação (ver Tabelas 2 e 4) é possível observar que ambos obtiveram progresso em suas notas. No entanto, os alunos de graduação obtiveram um progresso superior, tendo a média aumentada em mais de 270% e sua mediana aumentada em 800%. Já os estudantes da pós-graduação, aumentaram a sua média em mais de 40% e a sua mediana em mais de 54%. Analisando apenas o grupo dos estudantes que não tiveram contato com o assunto de redes de Petri antes de serem convidados (ver Tabela 3), observamos que o progresso deles foi muito superior ao progresso dos outros grupos. Nesse caso, a média aumento em 4759% evoluindo de 1,42 pontos para 69 pontos.

A fim de verificar se de fato existe uma diferença estatística significativa entre as notas dos alunos antes de depois do uso da ferramenta, iremos realizar um teste de hipótese. Vale mencionar que o teste foi realizado com o uso da linguagem R (Ihaka & Gentleman, 1996).

Primeiramente, realizou-se um teste de normalidade dos dois grupos de notas dos testes prévio e posterior dos 15 estudantes. A Tabela 5 mostra os resultados do teste de normalidade de *Shapiro-Wilk* com significância de 0,05. Segundo esse teste, se o resultado do cálculo de *p-value* for maior que 0,05, considera-se que o grupo de dados segue uma distribuição normal. Conforme mostrado na Tabela 5, os dois grupos de notas não seguem uma distribuição normal, pois a condição *p-value* > 0,05 não é satisfeita. Dessa forma, como a premissa de normalidade não é atendida, devemos realizar um teste não paramétrico a fim de verificar se o uso da ferramenta contribui no melhoramento das notas. Como os dois grupos de notas são dependentes, ou seja, foram extraídas de uma pesquisa com a mesma amostra de estudantes, consideramos realizar o teste não paramétrico de *Wilcoxon* para dados emparelhados com significância de 0,05 (Triola, 2017).

O teste de *Wilcoxon* compara dois grupos de dados emparelhados e considera duas hipóteses: hipótese nula – a mediana dos dois grupos são iguais; hipótese alternativa – a mediana dos dois grupos são diferentes. Se *p-value* > 0,05, então aceitamos a hipótese nula. Caso contrário, ou seja, *p-value* ≤ 0,05, aceitamos a hipótese alternativa. Do ponto de vista prático, se *p-value* ≤ 0,05, afirmamos que as medianas das notas dos alunos são diferentes. Ou seja, a ferramenta contribuiu para o aumento das notas. Desse modo, calculamos o *p-value* e verificamos que ele é igual a 0,001. Assim, a partir do teste de *Wilcoxon*, concluímos que a ferramenta contribuiu para o aprendizado dos estudantes.

Tabela 1: Comparação das notas dos 15 estudantes (graduação e pós-graduação) antes e depois de usarem a ferramenta.

Estatísticas das notas dos 15 estudantes (graduação e pós-graduação) antes e depois de usarem a ferramenta			
	Média	Mediana	Intervalo
Antes	28	10	0 - 90
Depois	72	80	30- 100

Tabela 2: Comparação das notas dos 11 estudantes de graduação antes e depois de usarem a ferramenta.

Estatísticas das notas dos 11 estudantes de graduação antes e depois de usarem a ferramenta			
	Média	Mediana	Intervalo
Antes	19	0	0 - 70
Depois	71	80	50 - 90

Tabela 3: Comparação das notas dos 7 estudantes da graduação que não tiveram contato com o assunto de redes de Petri antes de serem convidados.

	Estatísticas das notas dos 11 estudantes de graduação antes e depois de usarem a ferramenta		
	Média	Mediana	Intervalo
Antes	1,42	0	0 - 10
Depois	69	80	50 - 90

Tabela 4: Comparação das notas dos 4 estudantes de pós-graduação antes e depois de usarem a ferramenta.

	Estatísticas das notas dos 4 estudantes de pós-graduação antes e depois de usarem a ferramenta		
	Média	Mediana	Intervalo
Antes	52,5	55	10 - 90
Depois	75	85	30 - 100

Tabela 5: Teste de normalidade dos dois grupos de dados.

	Teste de <i>Shapiro-Wilk</i> com significância de 0,05	
	<i>p-value</i>	As notas segue uma distribuição normal, ou seja, <i>p-value</i> > 0,05 ?
Antes	0,001	não
Depois	0,0232	não

Agora serão mostrados os resultados da avaliação de usabilidade com os estudantes. A Figura 17 mostra o resultado das avaliações dos estudantes. As avaliações dos estudantes foram muito boas e em nenhum dos tópicos houve a avaliação como “péssimo”. Os tópicos 1, 2, 3, 7, 8 e 9 foram os mais bem avaliados. A boa avaliação dos tópicos 1, 2 e 3 indica uma boa usabilidade da ferramenta e a satisfação com a interface. Vale destacar também o tópico 9, que trata da eficácia para o aprendizado, que foi muito bem avaliado tanto pelos professores quanto pelos estudantes. Nos comentários, os estudantes elogiaram a ferramenta e afirmaram que conseguiram obter um maior domínio no assunto de modelagem usando as redes de Petri. Outro tópico também interessante de destacar é o tópico 7, que trata da clareza das mensagens de erro ou alerta. Apesar desse tópico ter sido bem avaliado pelos estudantes, ele foi um dos tópicos menos bem avaliados pelos professores.

Por outro lado, os tópicos 4, 5, 6 e 10 foram os menos bem avaliados pelos estudantes. Os tópicos 5 e 10 merecem destaque pois ambos foram os menos bem avaliados tanto pelos professores quanto pelos estudantes, reforçando que a visualização do progresso e o feedback

dos erros podem melhorar. As sugestões e opiniões dos estudantes para os tópicos 5 e 10 foram semelhantes às dos professores (mostradas nos Quadros 3 e 6). Os Quadros 7 e 8 mostram outras sugestões escritas pelos estudantes sobre a ferramenta agrupadas por tema, com os comentários dos autores deste trabalho para cada uma.

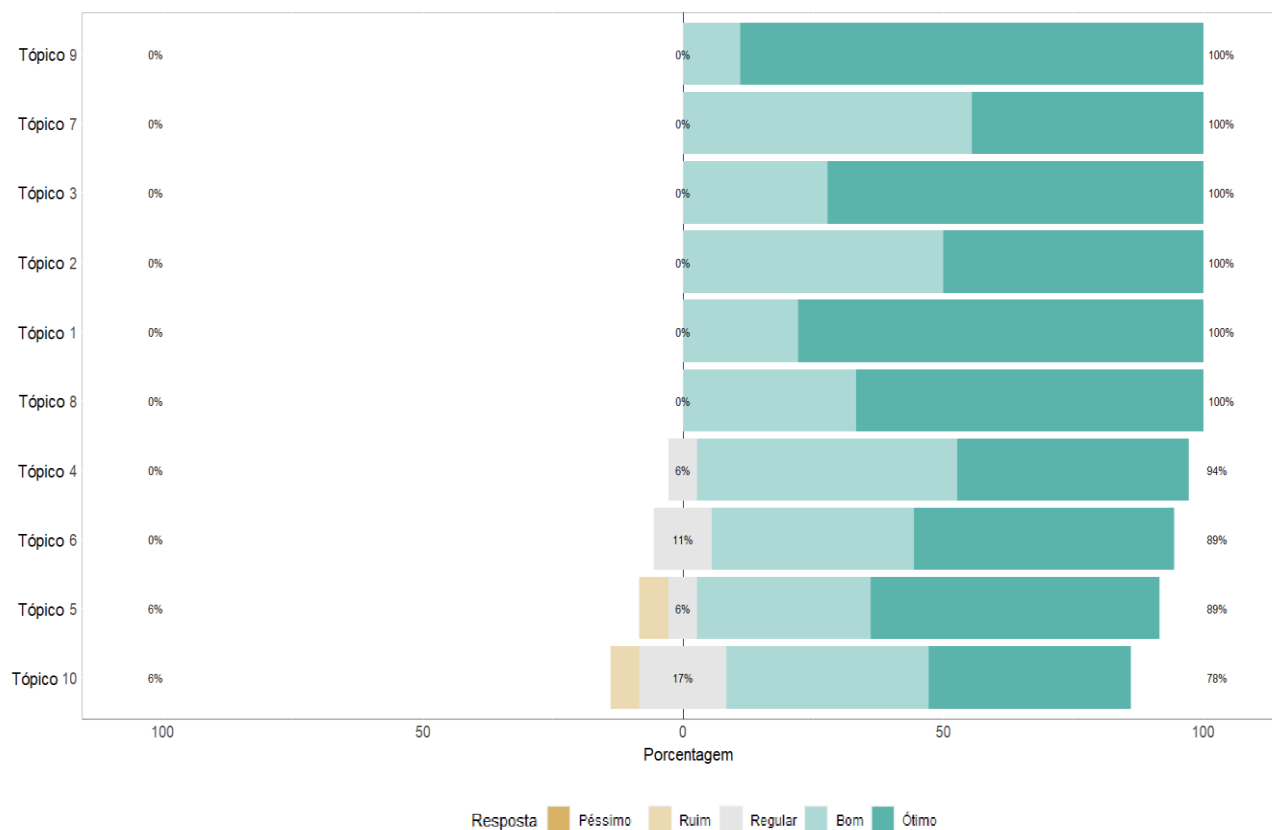


Figura 17: Resultados das avaliações feitas pelos estudantes (ver descrição dos tópicos no [Quadro 1](#)).

Quadro 7: sugestão dos estudantes quanto à liberdade (ou flexibilidade) na navegação.

Sugestão dos estudantes quanto à liberdade (ou flexibilidade) na navegação	
Sugestão	<p>Foi sugerido que o usuário possa navegar por todos os módulos mesmo sem ter concluído o anterior de modo a dar mais liberdade na navegação.</p> <p>Comentário dos autores: os módulos seguem uma sequência didática e lógica. Além disso, para se entender um módulo posterior, é necessário entender ou ter estudado o(s) módulo(s) anterior(es). Assim, por uma questão de aprendizagem, é preferível que a ferramenta não fique totalmente aberta.</p>

Quadro 8: sugestões dos estudantes quanto à facilidade de uso da ferramenta.

Sugestões dos estudantes quanto à facilidade de uso da ferramenta	
Sugestão	<p>Foi sugerido que ferramenta tenha um <i>layout</i> responsivo para o acesso a partir de dispositivos móveis como celular ou <i>tablets</i>.</p> <p>Comentário dos autores: a ferramenta foi desenvolvida usando <i>Bootstrap framework</i> e, portanto, possui elementos de responsividade. Embora seja possível construir modelos usando celular ou <i>tablet</i>, a interação com a interface é mais difícil. Nesse caso, em um trabalho futuro, pretendemos estender a ferramenta TryRdP com tecnologias específicas para esses dispositivos.</p>
Sugestão	<p>Considerar a questão de acessibilidade para usuários com certas deficiências.</p> <p>Comentário dos autores: para a ferramenta TryRdP, a questão de acessibilidade deve levar em conta não apenas os textos escritos, mas também, a construção e o controle dos modelos das redes de Petri. Isso torna o trabalho mais complexo e, nesse caso, um estudo futuro mais específico deve ser realizado.</p>

7 Conclusões

Artefatos digitais podem funcionar como potencializadores no aprendizado, em especial, para ensino de conteúdos de natureza abstrata, como no caso deste trabalho, modelagem usando redes de Petri. Não foram encontradas ferramentas que auxiliem no processo de ensino e aprendizagem de modelagem por meio das redes de Petri, e a ferramenta TryRdP contribui para suprir essa lacuna. A ferramenta tem um viés prático de modo a orientar o estudante a montar as redes de Petri que representem um determinado sistema. Com isso, os estudantes podem entender previamente “por quê”, “para que” e “como” usar as redes de Petri, antes de entrarem em contato com os detalhes matemáticos mais complexos do formalismo.

A ferramenta TryRdP foi desenvolvida para que o estudante a use individualmente e de maneira autônoma. Embora seja voltada, a princípio, para estudantes de graduação ou pós-graduação em computação, pode ser de interesse também a estudantes de qualquer área das ciências exatas que estejam motivados a aprender o assunto. Porém, além do uso autônomo que a ferramenta oferece, graças à sua estrutura que contempla uma sequência didática com conteúdos teóricos e exercícios para prática, a TryRdP também pode ser usada como uma ferramenta de apoio no ensino formal, em disciplinas de graduação e pós-graduação, como complemento às aulas, para reforçar o conhecimento adquirido, e em abordagens de sala de aula invertida. O professor pode acompanhar a resolução dos exercícios da TryRdP, e esclarecer dúvidas que o feedback da ferramenta não tenha conseguido suprir.

O ambiente da ferramenta permite que o estudante explore o assunto de modelagem de modo que ao abrir um determinado tópico (como modelagem de disponibilidade de um sistema computacional), o estudante terá acesso a uma página com explicações e exemplos sobre esse tópico. Em seguida, com base nos conceitos apresentados, o aluno é encorajado a reutilizar os modelos de RdP para a resolução de exercícios. O feedback imediato fornecido pela ferramenta durante a resolução de exercícios é essencial para que o estudante seja guiado e apoiado ao longo do aprendizado. A ferramenta mostra feedbacks em tempo real enquanto o aluno está fazendo os exercícios, de modo que caso ele/ela cometa algum erro relacionado à sintaxe do formalismo ou algum erro de digitação que comprometa a sintaxe das métricas uma mensagem

será apresentada. Além disso, caso o usuário tenha cometido algum erro, após a realização de um exercício ele/ela recebe uma mensagem com feedback sugerindo o possível erro para que o usuário possa corrigir. Entretanto, é importante destacar que, embora bem avaliadas, as mensagens de feedback poderiam ser melhoradas, de acordo com a avaliação de professores e estudantes.

Os resultados das avaliações com professores e alunos mostraram que a ferramenta tem grande potencial enquanto um instrumento de apoio ao ensino formal sobre modelagem de sistemas usando as redes de Petri, assim como, a ferramenta é, também, um bom instrumento para o estudo individual desse assunto. Apesar das avaliações positivas, o tamanho da amostra obtida foi pequeno contando apenas com 8 professores e 15 estudantes. Isso pode ser considerado uma limitação do experimento, visto que, se fosse utilizada uma amostra maior, o resultado das avaliações teria maior confiabilidade.

Possíveis trabalhos futuros, coletados a partir das sugestões dos participantes da pesquisa, incluem: tradução do texto da ferramenta para o inglês e o espanhol; desenvolvimento de uma versão da ferramenta para dispositivos móveis; investigar tecnologias para implementar o *tokengame*; adicionar vídeos explicativos com especialistas; e por fim, investigar tecnologias para aumentar a acessibilidade para pessoas com deficiência. Além disso, uma nova avaliação da ferramenta com uma amostra maior também poderia ser feita.

Agradecimentos

Queremos agradecer a FACEPE (Nº Processo: IBPG-0370-1.03/19) pelo apoio neste e em inúmeros outros trabalhos realizados pela Universidade Federal Rural de Pernambuco.

Referências

- Andrade, E., Nogueira, B., Matos, R., Callou, G., & Maciel, P. (2017). Availability modeling and analysis of a disaster-recovery as-a-service solution. *J Grid Computing*, 17, 603–621. DOI: [10.1007/s10723-018-9446-2](https://doi.org/10.1007/s10723-018-9446-2) [GS Search]
- Andrade, E. C., Alves, M., Nogueira, B., & Maciel, P. (2012). Calau: An environment for modeling and analyzing embedded real-time systems. In: *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference*, pages 3135–3140. DOI: [10.1109/ICSMC.2012.6378273](https://doi.org/10.1109/ICSMC.2012.6378273) [GS Search]
- Arteiro, R. D., Souza, F. N., Rosa, N. S., & Maciel, P. R. M. (2007). Utilizando redes de Petri para modelagem de desempenho de middleware orientado a mensagem. In: *XXVII Congresso da SBC*. [GS Search]
- Bennetts, R. (1975). On the analysis of fault trees. *IEEE Transactions on reliability*, 24(3):175–185. DOI: [10.1109/TR.1975.5215143](https://doi.org/10.1109/TR.1975.5215143) [GS Search]
- Bucci, G., Carnevali, L., Ridi, L., & Vicario, E. (2010). Oris: a tool for modeling, verification and evaluation of real-time systems. *Int J Softw Tools Technol Transfer*, 12, 391–403. DOI: [10.1007/s10009-010-0156-8](https://doi.org/10.1007/s10009-010-0156-8) [GS Search]
- Cavalcanti, T. C., Kim S., Lee K., Lee S-Y., Park M. K., & Hwang J. Y. (2020). Smartphone-based spectral imaging otoscope: System development and preliminary study for evaluation of its potential as a mobile diagnostic tool. *J. Biophotonics*, 13:e201960213. DOI:[10.1002/jbio.201960213](https://doi.org/10.1002/jbio.201960213) [GS Search]

- Cerqueira, R. G., & Silva, V. C. (2009). Aprendendo conceitos de computação gráfica através de um ambiente multimídia e interativo com OpenGL. *Anais do Workshop de Informática na Escola*, pp. 1733-1742. ISSN 2316-6541. [[GS Search](#)]
- Chiola, G., Marsan, M. A., Balbo, G., & Conte, G. (1993). Generalized Stochastic Petri Nets: A Definition at the Net Level and Its Implications. *IEEE Trans. Software Eng.*, 19, 89-107. DOI: [10.1109/32.214828](#) [[GS Search](#)]
- Costa, D., Teixeira, D., Grisotto, R., & Rocha, B. (2017). Lpt: ferramenta educacional para auxiliar o ensino/aprendizagem de traduções de diferentes níveis de linguagens de programação. In: *Anais do Workshop de Informática na Escola*, 23, page 695. DOI: [10.5753/cbie.wie.2017.695](#) [[GS Search](#)]
- Dietz, J., Chompalov, I., Bozeman, B., Lane, E., & Park, J. (2000). Using the Curriculum Vita to Study the Career Paths of Scientists and Engineers: An Exploratory Assessment. *Scientometrics*, 49(3), 419-442. DOI: [10.1023/a:1010537606969](#) [[GS Search](#)]
- Distefano, S., Paci, D., Puliafito, A., Scarpa, M., Papardo, C., & Sperone, S. (2005). Design and implementation of a performance plug-in for the argouml tool. In *IASTED Conf. on Software Engineering*, pages 337–342. [[GS Search](#)]
- Falani, L., Aguiar, C., & Forno, A. (2020). Mapeamento da literatura sobre as tecnologias da indústria 4.0 no segmento têxtil brasileiro. *Brazilian Journal of Development*, 6, 42437-42452. DOI: [10.34117/bjdv6n7-019](#) [[GS Search](#)]
- Farahani, B., Firouzi, F., Chang, V., Badaroglu, M., Constant, N., & Mankodiya, K. (2017). Towards fog-driven IoT eHealth: Promises and challenges of IoT in medicine and healthcare. *Future Generation Computer Systems*, 78. DOI: [10.1016/j.future.2017.04.036](#) [[GS Search](#)]
- Gómez-Martínez, E., & Merseguer, J. (2006). ArgoSPE: Model-based software performance engineering. In: *Petri Nets and Other Models of Concurrency-ICATPN 2006*, pages 401–410. DOI: [10.1007/11767589_23](#) [[GS Search](#)]
- Hirel, C., Sahner, R., Zang, X., & Trivedi, K. (2000a). Reliability and performability modeling using sharpe. In: *International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pages 345–349. Springer. DOI: [10.1007/3-540-46429-8_28](#) [[GS Search](#)]
- Hirel, C., Tuffin, B., & Trivedi, K. (2000b). SPNP: Stochastic petri nets. Version 6.0. *Lecture Notes in Computer Science*, 1786, 354-357. [[GS Search](#)]
- Ihaka, R., & Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. *Journal of Computational and Graphical Statistics*, 5(3), 299-314. DOI: [10.1080/10618600.1996.10474713](#) [[GS Search](#)]
- Joshi, A., Kale, S., Chandel, S., & Pal, D. K. (2015). Likert Scale: Explored and Explained. *Current Journal of Applied Science and Technology*, 7(4), 396-403. DOI: [10.9734/BJAST/2015/14975](#) [[GS Search](#)]
- Junior, O., & Aguiar, Y. P. C. (2018). Taxonomia de critérios para avaliação de software educativo-tacase. In: *Simpósio Brasileiro de Informática na Educação*, 29(1), pp. 298-307. DOI: [10.5753/cbie.sbie.2018.298](#) [[GS Search](#)]
- Kormann, H., & Suberg, B. (2021). Innovation Strategy for Renewal and Growth. In: Kormann H., Suberg B. (eds) *Topics of Family Business Governance*. Management for Professionals. Springer, Cham. DOI: [10.1007/978-3-030-58019-3_15](#) [[GS Search](#)]

- KPMG. (2020). *COVID-19 forces one of the biggest surges in tech investment in history, finds world's largest tech leadership survey*. Retrieved March 20, 2021, from <<https://home.kpmg/xx/en/home/media/press-releases/2020/09/covid-19-forces-one-of-the-biggest-surges-in-technology-investment-in-history-finds-worlds-largest-technology-leadership-survey.html#:~:text=The%20largest%20technology%20leadership%20survey,10%20IT%20leaders%20report%20that>>.
- Lima, J. W. S., Callou, G. R. A., & Andrade, E. C. (2021a). Teoria de Filas e Rede de Petri Estocástica: Um tutorial. *Research, Society and Development*, 10, p. e2810312826. DOI: [10.33448/rsd-v10i3.12826](https://doi.org/10.33448/rsd-v10i3.12826) [GS Search]
- Lima, J. W. S., Falcão, T. P., & Andrade, E. C. (2021b). TryRdP: uma Ferramenta para o Aprendizado de Modelagem de Sistemas usando Redes de Petri. In: *Anais do Simpósio Brasileiro de Educação em Computação*, pp. 362-370. SBC. DOI: [10.5753/educomp.2021.14504](https://doi.org/10.5753/educomp.2021.14504) [GS Search]
- Maciel, P., Lins, R., & Cunha, P. (1996). Uma Introdução às Redes de Petri e Aplicações. Campinas, SP: *Sociedade Brasileira de Computação*, v. 1. 213 p. [GS Search]
- Marcolino, M., Oliveira, J., D'Agostino, M., Ribeiro, A. L., Alkmim, M., & Novillo-Ortiz, D. (2018). The Impact of mHealth Interventions: Systematic Review of Systematic Reviews. *JMIR mHealth and uHealth*, 6, e23. DOI: [10.2196/mhealth.8873](https://doi.org/10.2196/mhealth.8873) [GS Search]
- Marsan, M. A., Conte, G., & Balbo, G. (1984). A Class of Generalized Stochastic Petri Nets for the Performance Evaluation of Multiprocessor Systems. *ACM Trans. Comput. Syst.*, 2, 93-122. DOI: [10.1145/190.191](https://doi.org/10.1145/190.191) [GS Search]
- Mei, H., Dong, X., Wang, Y., Tang, L., & Hu, Y. (2020). Managing patients with cancer during the COVID-19 pandemic: frontline experience from Wuhan. *The Lancet. Oncology*, 21(5), 634–636. DOI: [10.1016/S1470-2045\(20\)30238-2](https://doi.org/10.1016/S1470-2045(20)30238-2) [GS Search]
- Mo, P. et al. (2020). Clinical Characteristics of Refractory COVID-19 Pneumonia in Wuhan, China. *Clinical Infectious Diseases*, ciaa270. DOI: [10.1093/cid/ciaa270](https://doi.org/10.1093/cid/ciaa270) [GS Search]
- Murata, T. (1989). Petri nets: Properties, analysis and applications. In: *Proceedings of the IEEE*, 77(4), 541-580. DOI: [10.1109/5.24143](https://doi.org/10.1109/5.24143) [GS Search]
- Murata, T. (1984). Petri Nets and their Application an Introduction. In: *Chang SK. (eds) Management and Office Information Systems*. Springer, Boston, MA. [GS Search]
- Nielsen, J. (1994). Enhancing the explanatory power of usability heuristics. In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, 152-158. DOI: [10.1007/978-1-4613-2677-9_20](https://doi.org/10.1007/978-1-4613-2677-9_20) [GS Search]
- OECD/Eurostat (2005), Oslo Manual, Guidelines for Collecting and Interpreting Innovation Data, Paris: OECD. DOI: [10.1787/19900414](https://doi.org/10.1787/19900414) [GS Search]
- Oliveira, C. C., Costa, J. W., & Moreira, M. (2001). *Ambientes Informatizados de Aprendizagem: Produção e Avaliação de software Educativo*. Campinas, SP: Papirus. [GS Search]
- Raabe, A. L. A., & Bombasar, J. R. (2020). Mensuração e testes em Informática na Educação. In: *JQUES, Patricia Augustin; SIQUEIRA; Sean; BITTENCOURT, Ig; PIMENTEL, Mariano. (Org.) Metodologia de Pesquisa Científica em Informática na Educação: Abordagem Quantitativa. Porto Alegre: SBC. (Série Metodologia de Pesquisa em Informática na Educação, v. 2) Disponível em: <<https://metodologia.ceie-br.org/livro-2>>.* [GS Search]

- Rho, S., Vasilakos, A., & Chen, W. (2016). Cyber physical systems technologies and applications. *Future Generation Computer Systems*, 56. DOI: [10.1016/j.future.2015.10.019](https://doi.org/10.1016/j.future.2015.10.019) [GS Search]
- Rossiter, D. (2018). *Past, present & future of information technology in pedometrics*. Geoderma. DOI: [10.1016/j.geoderma.2018.03.009](https://doi.org/10.1016/j.geoderma.2018.03.009) [GS Search]
- Rugina, A. E., Kanoun, K., & Kaâniche, M. (2008). The adapt tool: From aadl architectural models to stochastic petri nets through model transformation. In: *Dependable Computing Conference, 2008. EDCC 2008*. Seventh European, pages 85–90. IEEE. DOI: [10.1109/EDCC-7.2008.14](https://doi.org/10.1109/EDCC-7.2008.14) [GS Search]
- Silva, B., Matos, R., Callou, G., Figueiredo, J., Oliveira, D., Ferreira, J., Dantas, J., Junior, A., Alves, V., & Maciel, P. (2015). Mercury: An integrated environment for performance and dependability evaluation of general systems. In: *Proceedings of Industrial Track at 45th Dependable Systems and Networks Conference (DSN)*. [GS Search]
- Tech Monitor. (2020). *CIOs Wielding More Influence in the Boardroom as IT Spending Surges by \$15 Billion a Week*. Retrieved March 20, 2021, from <https://techmonitor.ai/techonology/data/cio-survey-2020-kpmg>.
- Triola, M. F. (2017). *Introdução à Estatística*. 12a Edição. Editora LTC. [GS Search]
- Trivedi, K. S. (2008). *Probability & Statistics with Reliability, Queuing and Computer Science Applications*. John Wiley & Sons. [GS Search]
- Trivedi, K., Andrade, E., & Machida, F. (2012). Combining performance and availability analysis in practice. *Advances in Computers*, 84. Elsevier. [GS Search]
- U.S. Bureau of Labor Statistics. (2013). *Economic News Releases: Employment Projections*. Retrieved March 20, 2021, from <https://www.bls.gov/opub/btn/volume-2/careers-in-growing-field-of-information-technology-services.htm#ednref4>.
- U.S. Bureau of Labor Statistics. (2020). *Economic News Releases: Employment Projections*. Retrieved March 20, 2021, from <https://www.bls.gov/bls/newsrels.htm#OEP>.
- Valente, J. A. (2000). Informática na educação: instrucionismo x construcionismo. *Revista Educação Pública*. ISSN: 1984-6290. [GS Search]
- Zhou, K., Liu, T., & Zhou, L. (2015). Industry 4.0: Towards future industrial opportunities and challenges. In: *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, pp. 2147-2152. [GS Search]
- Zimmermann, A. (2017). Modelling and Performance Evaluation with TimeNET 4.4. In: Bertrand N., Bortolussi L. (eds) Quantitative Evaluation of Systems. QEST 2017. *Lecture Notes in Computer Science, vol 10503*. Springer, Cham. [GS Search]