UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JÚLIO RODRIGUES DE MENDONÇA NETO

Modelos Probabilísticos e Método de Tomada de Decisão para Avaliação e Seleção de Estratégias de Recuperação de Desastres

Recife
2020

JÚLIO RODRIGUES DE MENDONÇA NETO

Modelos Probabilísticos e Método de Tomada de Decisão para Avaliação e Seleção de Estratégias de Recuperação de Desastres

Tese de Doutorado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de Concentração: Engenharia de Software e Linguagens de Programação
Orientador: *Prof. Dr.* Ricardo Massa Ferreira Lima
Coorientador: *Prof. Dr.* Ermeson Carneiro de Andrade

Recife
2020

Catalogação na fonte
Bibliotecária Mariana de Souza Alves CRB4-2105

M539m Mendonça Neto, Júlio Rodrigues de
Modelos probabilísticos e método de tomada de decisão para avaliação e seleção de estratégias de recuperação de desastres / Júlio Rodrigues de Mendonça Neto. – 2020.
177f., il., fig., tab.

Orientador: Ricardo Massa Ferreira Lima.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn, Ciência da computação, Recife, 2020.
Inclui referências e apêndices.

1. Engenharia de Software e Linguagens de Programação. 2. Recuperação de desastres. 3. Tolerância a falhas. 4. Modelos. I. Lima, Ricardo Massa Ferreira. (orientador) II. Título.

005.1 CDD (22. ed.) UFPE-CCEN 2021-04

JÚLIO RODRIGUES DE MENDONÇA NETO

"**Modelos Probabilísticos e Método de Tomada de Decisão para Avaliação e Seleção de Estratégias de Recuperação de Desastres**"

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 29/10/2020.

Orientador: *Prof. Dr.* Ricardo Massa Ferreira Lima

BANCA EXAMINADORA

Prof. Dr. Paulo Romero Martins Maciel
Centro de Informática / UFPE

Prof. Dr. Nelson Souto Rosa
Centro de Informática / UFPE

Prof. Dr. Bruno Costa e Silva Nogueira
Instituto de Computação / UFAL

Prof. Dr. Gustavo Rau de Almeida Callou
Departamento de Computação/ UFRPE

Prof. Dr. Jamilson Ramalho Dantas
Colegiado em Ciência da Computação / UNIVASF

À Diana, Júlio, Felipe e Marco por todo apoio e amor durante minha vida.

AGRADECIMENTOS

Ao meu orientador, Prof. Ricardo Massa, por aceitar o desafio de me orientar, pelo ótimo acolhimento, paciência, amizade e principalmente pela confiança em mim e no meu trabalho. Agradeço ao meu co-orientador, Prof. Ermeson Andrade, pelo imensurável apoio nessa caminhada, por sempre estar disponível para ajudar, aconselhar, pela paciência e amizade. Agradeço também ao Prof. Dongseong Kim pela valiosa oportunidade de realizar o doutorado sanduíche em seu laboratório.

Agradeço a todos os membros da banca, os professores Nelson Rosa, Jamilson Dantas, Gustavo Callou e Bruno Nogueira, pelos valiosos comentários e revisões feitas nesta tese e, de forma especial, agradeço ao professor Paulo Maciel pelas revisões, comentários, discussões e mentorias recebidas durante os anos de mestrado e doutorado no CIn-UFPE.

Aos meus pais, Diana e Júlio, pelo apoio, incentivo, por acreditarem em mim, por me dar coragem, força e carinho para seguir meu caminho. Mesmo diante de tantas dificuldades, vocês sempre fazem de tudo para apoiar seus filhos e nos ajudar a vencer nossas batalhas. Sem vocês eu nada seria. Não poderia ter melhores pais. Eu amo vocês. Também sou grato à meus avós, Júlio Rodrigues (vovô Júlio, *in memoriam*), Josefa Barbosa (vovó Zezita, *in memoriam*), Antônio Pereira (vovô Antônio, *in memoriam*) e Mercês Lira (vovó Mercês, *in memoriam*) que mesmo não estando presentes fisicamente durante toda essa jornada, me ajudaram a caminhá-la, cuidando de mim e me ajudando a me tornar quem sou.

Agradeço à minha companheira, Helena, por todo amor, carinho, incentivo, apoio e motivação durante essa jornada. À minha toda minha família, por todo apoio, encorajamento, suporte e amizade. De forma especial, aos meus irmãos, Marco e Felipe. Aos meus tios, Fátima, Milton, Rosa, Josué, Ivanice (*in memoriam*), João, Floriza e Lindoaldo. À todas minhas primas e primos que estiveram junto comigo nessa jornada: Arthur, Gabriel, Thiago, Vinícius, Inês, Izabel, Betânia, Lindicélia, Lindivânia e Lindifávia. A minha madrinha, Josete e ao meu padrinho Vavá (*in memoriam*), que sempre me apoiaram. Vocês, e todos os outros não citados aqui (são muitos), dão mais sentido a vida.

Aos amigos que fiz no Cin-UFPE e UFRPE, em especial a Carlos Melo, Érico Augusto, Eric Borba, Ewerton Queiroz, Marco Aurélio, Paulo Pereira, Ronierison Maciel e Wilson Medeiros por todo o convívio, amizade, discussões, apoio, e ajuda.

Aos amigos Agay Borges, Camila Sabino, Dreyton Henrique, Fernanda Franklin, Giully Tavares, Jair Galvão, Íris Nascimento, Júlia Melo, Marina Ângela, Marília Ângela, Matheus Soares, Michael Machado, Niellys Costa, Thainá Nogueira, Túlio Quirino, Pedro Melo, Rafaela Teotônio, Vick Brito, Rubens Marinho, Yago Nobre pelas discussões e bons momentos de descontração.

A todos os amigos e colegas que me apoiaram durante minha estadia na Nova Zelândia: Ashu Dutta, Bilal Ishfaq, Dilli Sharma, Haipeng Li, Matthew Ruffell, Jihye Yun, Sanjay

Basukala e Simon Enochson.

Ao CIn-UFPE, FACEPE, e CAPES pela infraestrutura e pelos recursos recebidos.

RESUMO

Em empresas de todos os tipos e tamanhos, os sistemas de Tecnologia da Informação e Comunicação (TIC) têm se tornado vitais para delas. A ocorrência de desastres ou falhas inesperadas podem afetar a infraestrutura computacional sob a qual os sistemas de TIC operam, causando a indisponibilidade desses sistemas e/ou perda de dados. A indisponibilidade de tais sistemas ainda pode gerar insatisfação dos clientes e, consequentemente, ocasionar a diminuição de receita. Nos últimos anos, estratégias de *Disaster Recovery* (DR) têm sido adotadas para auxiliar na redução da indisponibilidade dos sistemas de TIC e na mitigação da perda de dados. Contudo, não existe uma estratégia única que atenda aos requisitos de todas as organizações (ex.: baixo custo e alta disponibilidade). A avaliação eficiente e precisa das estratégias de DR é fundamental para garantir que uma delas seja mais adequada à realidade de uma empresa. Os modelos probabilísticos têm sido usados para representar e analisar infraestruturas computacionais e sistemas de TIC, por serem eficientes para realização de análises quantitativas. Adicionalmente, métodos de decisão multicritério têm sido empregados para auxiliar no processo de tomada de decisão, especialmente quando são considerados critérios conflitantes. Dessa forma, esta tese propõe o uso integrado de modelos probabilísticos, modelos de custo e métodos de decisão multicritério para analisar e selecionar estratégias de DR que sejam mais adequadas aos requisitos ou restrições das empresas. Uma metodologia é definida para guiar o processo de avaliação de tais estratégias com os modelos e método propostos. Embora existam vários tipos de modelos probabilísticos, este trabalho, mais especificamente, adota as redes de Petri devido à sua capacidade de modelar sistemas computacionais e permitir a obtenção de métricas importantes relativas à DR, como disponibilidade, indisponibilidade, *Recovery Time Objective* (RTO) e *Recovery Point Objective* (RPO). Os modelos de custos são propostos pois pequenas e médias empresas muitas vezes possuem um orçamento limitado para investimento em DR, sendo este um fator relevante para a escolha de uma estratégia. Apesar desses modelos ajudarem as empresas ou os projetistas na avaliação das estratégias de DR, a quantidade de opções a serem analisadas pode ser imensa. Assim, também é proposto a utilização de um método de decisão multicritério, baseado na *Technique for Order of Preference by Similarity to Ideal Solution* (TOPSIS), para auxiliar no processo de seleção de estratégias de DR mais adequadas aos requisitos definidos pelas empresas ou projetistas. Por fim, estudos de caso demonstram a viabilidade e a utilidade da abordagem proposta em orientar empresas ou projetistas no processo de avaliação e seleção de estratégias de DR que mais se adequam às necessidades delas.

Palavras-chaves: Recuperação de desastres. Tolerância a falhas. Modelos. Redes de Petri. Dependabilidade. Tomada de decisão.

ABSTRACT

In companies of all types and sizes, Information and Communication Technology (ICT) systems have become vital to them. The occurrence of disasters or unexpected failures can affect the computational infrastructure under which ICT systems operate, causing these systems' unavailability and/or data loss. Such systems' unavailability can still generate customer dissatisfaction and, consequently, cause a decrease in revenue. In recent years, Disaster Recovery (DR) strategies have been adopted to reduce ICT systems' unavailability and mitigate data loss. However, not a single strategy meets all organizations' requirements (e.g., low cost and high availability). An efficient and accurate assessment of DR strategies is essential to ensure that one of them is more appropriate for a company's reality. Probabilistic models have been used to represent and analyze computational infrastructures and ICT systems, as they are efficient for performing quantitative analyzes. Additionally, multicriteria decision-making methods have been used to assist in the decision-making process, especially when conflicting criteria are considered. Thus, this thesis proposes the integrated use of probabilistic models, cost models, and multicriteria decision-making methods to analyze and select DR strategies that are more proper to the companies' requirements or restrictions. A metodology is defined to guide the evaluation process of these strategies using the proposed models and methods. Although there are several probabilistic models, this work, more specifically, adopts Petri nets due to its capacity to model computational systems and allow obtaining important metrics related to DR, such as availability, unavailability, Recovery Time Objective (RTO), and Recovery Point Objective (RPO). The cost models are proposed as small and medium-sized companies often have a limited budget for DR, which is a relevant factor for a strategy choice. Therefore, this work also proposes cost models that allow the evaluation of DR infrastructures considering the acquisition, maintenance, and operation costs. In this analysis, the costs related to the loss of productivity and sales caused by the evaluated infrastructures' unavailability are also considered. Although these models help companies or designers to evaluate DR strategies, the number of options to be analyzed can be immense. Thus, this work also proposes a multicriteria decision-making method, based on the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS), to assist in selecting DR strategies most appropriate to the requirements defined by companies or designers. Finally, case studies demonstrate the feasibility and usefulness of the proposed approach in guiding companies or designers in evaluating and selecting DR strategies that best suit their needs.

Keywords: Disaster recovery. Fault tolerance. Models. Petri nets. Dependability. Decision-making.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AHP	*Analytic Hierarchy Process*
API	*Application Programming Interface*
BD	Banco de Dados
CA	Custo de Aquisição
CM	Custo de Manutenção
CO	Custo de Operação
COP	Custo de Oportunidade
CTMC	*Continuous Time Markov Chain*
DC	*Data Center*
DEMATEL	*Decision-Making Trial and Evaluation Laboratory*
DNP	*DEMATEL-based Network Process*
DR	*Disaster Recovery*
DSPN	*Deterministic and Stochastic Petri Net*
GB	*Gigabyte*
GDE3	*Generalized Differential Evolution*
GRASP	*Greedy Randomized Adaptive Search Procedure*
HD	*Hard Disk*
IaaS	*Infraestructure-as-a-Service*
IC	Intervalo de Confiança
ICT	Information and Communication Technology
IoT	*Internet of Things*
IP	*Internet Protocol*
LEP	*Lost Employee Productivity*
LSR	*Lost Sales Revenue*
MARTE	*Profile for Modeling and Analysis of Real-time and Embedded systems*
MB	*Megabyte*
Mbps	*Megabits* por segundo
MCC	*Mobile Cloud Computing*
MCDM	*Multi-criteria decision-making*
MTTF	*Mean Time to Fail*

MTTR	*Mean Time to Repair*
NIST	*National Institute of Standards and Technology*
NSGA-II	*Non-dominated Sorting Genetic Algorithm*
PaaS	*Plataform-as-a-Service*
PoI	*Privately Owned Infrastructure*
RAM	*Random-access Memory*
RBD	*Reliability Block Diagram*
RPO	*Recovery Point Objective*
RTO	*Recovery Time Objective*
SaaS	*Software-as-a-Service*
SDN	*Software-Defined Network*
SGBD	Sistema Gerenciador de Banco de Dados
SLA	*Service-level Agreement*
SO	Sistema Operacional
SPN	*Stochastic Petri Nets*
SQL	*Structured Query Language*
SRN	*Stochastic Reward Nets*
SSD	*Solid-State Drive*
SVC	*SAN Volume Controller*
SysML	*System Modelling Language*
TIC	Tecnologia da Informação e Comunicação
TOPSIS	*Technique for Order of Preference by Similarity to Ideal Solution*
VIKOR	*VIšekriterijumsko KOmpromisno Rangiranje*
VM	*Virtual Machine*
VMM	*Virtual Machine Monitor*

SUMÁRIO

1 INTRODUÇÃO

Este capítulo apresenta o contexto no qual o trabalho está inserido e descreve a motivação e a justificativa para sua realização. São destacados aspectos relacionados, principalmente, à importância dos sistemas de Tecnologia da Informação e Comunicação (TIC) no mundo dos negócios atuais e o planejamento e avaliação de estratégias de *Disaster Recovery* (DR) (em português, Recuperação de Desastres) para tais sistemas. Além disso, são apresentados os objetivos desta tese e, por fim, é detalhada sua organização.

1.1 CONTEXTO

Nos ambientes de negócios modernos, os sistemas de TIC se tornaram vitais para o funcionamento de grandes ou pequenas empresas. Geralmente, esses sistemas precisam dar suporte às operações das empresas 24 horas por dia, 7 dias por semana, 365 dias ao ano. Quer seja para apoiar as atividades da empresa ou manter os negócios operacionais, os sistemas de TIC precisam de uma infraestrutura computacional para operar. Desta forma, tanto a infraestrutura computacional como os sistemas necessitam estar funcionando para auxiliar as operações das empresas. A ocorrência de falhas na infraestrutura computacional ou nos sistemas pode resultar em interrupções, inviabilizando a correta operação dos sistemas, e pode ter sérias consequências. Por exemplo, a perda ou indisponibilidade de dados devido a tais interrupções não previstas pode gerar a insatisfação de clientes ou perda de vendas, ocasionando diminuição de receita. Uma pesquisa realizada pela ITIC apontou que 86% das empresas poderiam perder mais de U$ 300.000,00 para cada hora de indisponibilidade dos sistemas de TIC (INFORMATION TECHNOLOGY INTELLIGENCE CONSULTING CORP, 2019).

Nesse contexto, com o objetivo de auxiliar a mitigar a indisponibilidade de sistemas e a perda de dados, surge a DR. A DR pode ser definida como um conjunto de processos e métodos usados para garantir que sistemas possam ser recuperados após a ocorrência de falhas inesperadas ou extraordinárias (BAUER; ADAMS; EUSTACE, 2011; REESE, 2009). Nos últimos anos, diferentes estratégias de DR têm sido adotadas para auxiliar na redução da indisponibilidade dos sistemas de TIC e na mitigação da perda de dados (LOWE; DAVIS; GREEN, 2016). Especialmente com a popularização da computação em nuvem, a adoção de diferentes estratégias de DR tem crescido (UNITRENDS, 2018). As razões para isso consistem principalmente no fato de estratégias de DR baseadas em computação em nuvem apresentarem benefícios, como por exemplo, a facilidade da contratação e um menor investimento inicial, se comparadas a estratégias tradicionais como *backup* e restauração de dados através de fitas (ANDRADE et al., 2017).

Diferentes estratégias (ex.: *backup*, replicação de dados ou replicação de Bancos de

Dados (BDs)) podem ser utilizadas para prover a capacidade de recuperação de desastres aos sistemas de TIC (ALSHAMMARI; ALWAN; ALSHAIKHLI, 2016). Porém, não existe uma estratégia única que consiga atender aos requisitos de todas as organizações. Estratégias projetadas para oferecer alta disponibilidade dos serviços podem cumprir com essa função, mas isso pode resultar em um alto custo e uma infraestrutura computacional subutilizada, visto que os desastres são eventos raros. Por outro lado, estratégias projetadas para suportar a capacidade operacional regular de um sistema (ex.: atender até 500 clientes conectados por minuto) podem não ser eficientes para lidar com a demanda em caso de falhas inesperadas que causem interrupções de parte ou de todo o sistema.

Quando se trata de pequenas e médias empresas, existem diferentes critérios e limitações financeiras a serem consideradas para o investimento em TICs. A escolha de estratégias que atendam os diferentes requisitos desse tipo de empresas é fundamental (ALMEIDA et al., 2015). Assim, a escolha de uma estratégia de DR não pode considerar apenas um único critério (ex.: maximizar a disponibilidade). Isto é, é necessário considerar múltiplos critérios como disponibilidade, custo, e a eficácia da estratégia. Dessa forma, a escolha de uma estratégia de DR se torna tarefa complexa, devido à variedade de estratégias existentes e suas configurações possíveis.

Ao mesmo tempo, os modelos analíticos vêm sendo utilizados para análise de infraestruturas e sistemas computacionais (NGUYEN; KIM; PARK, 2016; SILVA, 2016; LIU et al., 2018; Bai et al., 2020). Cadeias de Markov (TRIVEDI, 1982) e redes de Petri (PETRI, 1962) são exemplos de formalismos que têm sido empregados para a modelagem, a análise e a simulação de diferentes tipos de sistemas, incluindo infraestruturas computacionais e estratégias de DR. Esses formalismos possuem fundamentação matemática sólida e, consequentemente, são eficientes para realizar análises quantitativas, além de verificações de propriedades. Isso tudo de forma rápida, com baixo custo financeiro e um bom nível de confiabilidade dos resultados se comparados à implementação de ambientes de testes reais.

1.2 MOTIVAÇÃO E JUSTIFICATIVA

Os desastres podem pôr em risco o funcionamento de sistemas e infraestruturas computacionais e, consequentemente, os negócios. Até mesmo grandes empresas como o Google, a Amazon *web services*, o *WhatsApp*, a *America Airlines*, e o banco HSBC experimentaram ocorrências de falhas inesperadas que causaram a interrupção de parte ou de todos os seus serviços (CHARETTE, 2018; MEDINA, 2020). Um relatório apresentado pela *LogicMonitor* (LOGICMONITOR, 2019) apontou que 93% das empresas pesquisadas sofreram com algum tipo de falha nos últimos 3 anos. Vale destacar que empresas que participaram dessa pesquisa estão localizadas em diferentes países e são de diversas áreas como tecnologia, transporte, educação, finanças, e saúde. O mesmo relatório ainda aponta que empresas que sofrem com essas falhas inesperadas podem ter um custo operacional 16 vezes maior que aquelas mais bem preparadas para tais falhas. Isso mostra que a proteção dos sistemas

de TIC contra falhas inesperadas são de extrema importância para evitar altos custos operacionais.

As empresas precisam estar preparadas para as ocorrências de falhas inesperadas ou até mesmo catastróficas que podem causar interrupções dos sistemas de TIC. Apesar de ser praticamente impossível eliminar todos os riscos das ocorrências dessas falhas, a adoção de estratégias de DR pode auxiliar a mitigar os danos causados pelas mesmas nos sistemas computacionais (LOGICMONITOR, 2019). A crescente adoção de estratégias de DR por parte das organizações nos últimos anos demostra a preocupação e necessidade existentes relacionadas à proteção de dados e operacionalidade dos sistemas de TIC (UNITRENDS, 2018). Além disso, a adoção de estratégias de DR ainda pode trazer benefícios como redução dos custos, preservação da reputação das empresas, e redução no tempo de recuperação dos sistemas de TIC em caso de falhas.

A modelagem através das redes de Petri tem sido utilizada para a avaliação de diversos tipos de sistemas. Dentre esses tipos de sistemas, é possível destacar as infraestruturas computacionais e as estratégias de DR (NGUYEN; KIM; PARK, 2016; ANDRADE et al., 2017; LIU et al., 2018; OLIVEIRA et al., 2019). Uma extensão das redes de Petri, é chamada *Deterministic and Stochastic Petri Net* (DSPN). As DSPNs podem ser adotadas para análise de desempenho, confiança e disponibilidade de sistemas dinâmicos e eventos discretos. Tal formalismo provê diversos recursos para modelagem, tais como funções de guarda, pesos, prioridade, taxas determinísticas e exponenciais, dependência de marcação, entre outros (GERMAN, 2000). Assim, a modelagem através das DSPNs é um recurso poderoso para avaliar estratégias de DR. Vale ser ressaltado que a obtenção de métricas relativas à estratégias de DR pode ser bastante útil para verificar o comportamento, o funcionamento e a viabilidade da adoção de estratégias de DR antes de sua implantação.

Uma vez que empresas possuem diferentes características e requisitos, a escolha de uma estratégia de DR deve considerar essas diferenças. Visto que esses requisitos podem ser representados através de métricas quantitativas, métodos de decisão multicritério podem auxiliar na escolha das melhores alternativas de DR para uma empresa. Métodos de decisão baseados em redes neurais, algoritmos genéticos, *Analytic Hierarchy Process* (AHP), e *Technique for Order of Preference by Similarity to Ideal Solution* (TOPSIS) têm sido adotados para auxiliar no processo decisório envolvendo múltiplos critérios (JAIN; DUBES, 1988; ZOPOUNIDIS; PARDALOS, 2010). Assim, uma abordagem integrada baseada em modelos analíticos e métodos de decisão multicritério para escolher as estratégias de recuperação de desastres é fundamental.

1.3 PROBLEMA

Os sistemas de TIC e as informações (dados) gerenciadas por eles são essenciais para o funcionamento e a operacionalidade das empresas. Porém, a ocorrência de eventos de desastres pode impossibilitar o funcionamento normal desses sistemas, sejam eventos

oriundos de causa natural (ex.: tempestades ou enchentes), sejam causados pela ação humana (ex.: ataques terroristas, imprudência ou ataque de *hackers*) ou ainda uma combinação dos dois (LOGICMONITOR, 2019). Portanto, as empresas precisam estar preparadas para mitigar os danos ou as perdas que podem acontecer devido às interrupções ocasionadas por tais eventos.

Embora as estratégias de DR possam mitigar as perdas de dados e manter os sistemas de TIC operacionais, existe uma grande variedade de estratégias de DR que pode ser empregada em infraestruturas computacionais (MENDONçA et al., 2019a). Cada uma dessas estratégias possui características próprias, e que podem variar de acordo com o ambiente em que possam ser adotadas (ANDRADE et al., 2017). Sendo assim, é necessário que as empresas consigam selecionar as estratégias que melhor se adequem tanto à realidade delas quanto ao orçamento.

As estratégias de DR precisam observar as necessidades específicas de cada organização. Essas necessidades podem contemplar critérios técnicos (ex.: indisponibilidade ou número de transações perdidas) e critérios não-técnicos, como custo. Adicionalmente, na maioria das vezes, os critérios e requisitos das empresas são conflitantes (ex.: alta disponibilidade e baixo custo). Alguns trabalhos (SILVA, 2016; NGUYEN; KIM; PARK, 2016; YANG et al., 2017; ANDRADE; NOGUEIRA, 2019) têm sido realizados para auxiliar no planejamento da implantação de estratégias de DR através da modelagem de sistemas e infraestruturas. No entanto, esses trabalhos apresentam limitações por não considerarem diferentes características das estratégias de DR, além dos critérios e dos requisitos de cada empresa (MENDONçA et al., 2019a). Considerando esses desafios, o problema principal abordado nesta tese é definido na seguinte pergunta: *como auxiliar empresas, projetistas ou pessoas interessadas na adoção de estratégias de DR a selecionar a(s) melhor(es) estratégia(s) considerando diferentes requisitos e critérios?*

Dessa forma, esta tese de doutorado propõe a análise e a seleção de estratégias de DR que sejam mais adequadas aos requisitos e critérios das empresas. Para alcançar tais objetivos é proposto o uso integrado de modelos probabilísticos, modelos de custo e métodos de tomada de decisão multicritério para analisar e selecionar estratégias de DR de acordo com os requisitos e critérios das empresas. Além disso, é proposta uma metodologia para guiar o processo de análise e seleção de tais estratégias. O uso integrado desses elementos poderá oferecer às empresas, aos projetistas ou às pessoas interessadas informações para o processo de tomada de decisão de escolher ou não uma determinada estratégia de DR.

1.4 OBJETIVOS

Diante das motivações, dos problemas e dos desafios descritos acima, o principal objetivo desta tese é propor um conjunto de modelos para avaliar e selecionar estratégias de DR que podem ser implantadas em infraestruturas computacionais. Nesta tese, são

abordadas as estratégias de *backup*, replicação de BD, e replicação de ambientes. Dessa forma, são propostos modelos probabilísticos e modelos de custo para avaliar as estratégias quantitativamente considerando importantes métricas de DR, como disponibilidade, indisponibilidade, *Recovery Time Objective* (RTO), *Recovery Point Objective* (RPO), e custo. Também é proposta a aplicação de um método de decisão multicritério para auxiliar na seleção e ranqueamento das estratégias de DR que são avaliadas através dos modelos probabilísticos. Esse método de decisão permite selecionar as estratégias de DR que mais se adequam aos requisitos específicos de cada empresa (ex.: maximizar a disponibilidade e minimizar o RTO e o custo). Assim, o uso integrado dos modelos e método de decisão permite a identificação da(s) estratégia(s) de DR que melhor se adeque às necessidades das empresas.
Mais especificamente, os objetivos desta tese são:

- propor modelos para análise de diferentes estratégias de DR. Tais modelos permitirão a realização de análises quantitativas para obtenção de importantes métricas de DR como disponibilidade, RTO, e RPO;
- avaliar os custos de utilização, de operação e de manutenção de diferentes estratégias de DR;
- aplicar um método de tomada de decisão que permita selecionar a(s) estratégia(s) de DR mais adequadas considerando critérios múltiplos. A adoção de tal método visa a facilitar o processo de tomada de decisão por parte das empresas, dos projetistas ou das pessoas interessadas na adoção de estratégias de DR.
- definir uma metodologia que guie o uso integrado de modelos probabilísticos, modelos de custo, e métodos de tomada de decisão multicritério para avaliação de diferentes estratégias de DR;
- analisar a corretude dos modelos propostos através da execução de experimentos em infraestruturas reais.

1.5 ORGANIZAÇÃO DA TESE

Esta tese está organizada como descrito a seguir. O Capítulo 2 introduz os conceitos fundamentais utilizados nesta tese. O Capítulo 3 detalha os trabalhos relacionados. O Capítulo 4 descreve a metodologia proposta para análise e seleção de estratégias de DR. O Capítulo 5 apresenta os modelos desenvolvidos e adotados para analisar diferentes estratégias de DR. O Capítulo 6 apresenta os estudos de caso realizados utilizando a metodologia e modelos propostos. Por fim, o Capítulo 7 conclui o trabalho, expõe as contribuições, discute as limitações e apresenta os trabalhos futuros.

2 FUNDAMENTOS

Este capítulo apresenta os principais conceitos para um melhor entendimento desta pesquisa. Primeiramente, conceitos sobre virtualização e computação em nuvem são introduzidos. Posteriormente, são apresentados alguns fundamentos sobre dependabilidade, disponibilidade de sistemas, modelagem de sistemas e redes de Petri. Também são mostrados conceitos acerca da recuperação de desastres e estratégias associadas. Por último, são abordados os principais conceitos relacionados à análise de agrupamento e métodos de tomada de decisão multicritério.

2.1 VIRTUALIZAÇÃO

Em computação, o termo virtualização geralmente se refere à abstração de um componente físico em um objeto lógico (PORTNOY, 2012). Assim, tecnologias de virtualização proveem a capacidade de emular diferentes recursos de *hardware*. Existem vários possíveis produtos de virtualização, como de servidores, de redes e de armazenamento. Dessa forma, a virtualização possui os subsídios necessários para emular capacidades de armazenamento, processamento e rede, dentro de um dispositivo. Assim, a virtualização é uma das tecnologias mais importantes para habilitar ambientes de computação em nuvem.

Uma das aplicações mais comuns para a virtualização, é a virtualização de servidores. Esta técnica torna possível a hospedagem de vários Sistemas Operacionais (SOs) independentes em uma única máquina física. Usualmente, um SO virtualizado é chamado de SO convidado ou *Virtual Machine* (VM). Existem diferentes abordagens para a virtualização de servidores, como a *Full Virtualization*, *Paravirtualization*, *Operating System virtualization* e *Hardware Virtualization* (GUEDES, 2019). Em todas as abordagens, os SO convidados têm seus recursos (ex.: memória, processador, armazenamento e conexões de rede) gerenciado por um *Hypervisor*, também conhecido como *Virtual Machine Monitor* (VMM) (PORTNOY, 2012). A abordagem mais comumente utilizada é a *Full Virtualization*, onde todos os componentes de *hardware* são virtualizados para cada SO convidado. Na *Paravirtualization*, os recursos de *hardware* são compartilhados entre os SOs convidados, que têm conhecimento que são executo sobre um *Hypervisor*. Já a *Operating System virtualization*, também conhecida como virtualização baseada em *containers*, utiliza recursos do SO para permitir a existência várias instâncias isoladas de espaços de usuário. Apesar das diferentes abordagens, em geral, o uso da virtualização de servidores ajuda a reduzir a subutilização de servidores físicos, além de prover gerenciamento centralizado para um conjunto de servidores.

2.1.1 Computação em Nuvem

O *National Institute of Standards and Technology* (NIST) define a computação em nuvem como (tradução livre):

> *"Um modelo que fornece conveniência e provê acesso sob demanda, através da rede, a um conjunto de recursos computacionais compartilhados e configuráveis (redes, servidores, armazenamento, aplicações e serviços), que podem ser rapidamente disponibilizados com um esforço mínimo de gerenciamento ou interação com o provedor de serviços."(BADGER et al., 2012)*

Este modelo permite que usuários tenham acesso aos recursos computacionais de forma elástica, sob demanda e a um baixo custo, entregues de maneira semelhante a serviços tradicionais como água, gás, eletricidade e telefonia. Isto significa que os usuários podem solicitar estes recursos sempre que necessário, e serão cobrados apenas pela utilização deles durante o período de tempo que estes recursos foram disponibilizados (DINH et al., 2011). A computação em nuvem é um paradigma computacional recente, mas que foi construído com base em tecnologias existentes como virtualização, *grid computing* e *utility computing*. Ela utiliza-se da maioria dos tipos de virtualização. Dentre eles, o mais presente na computação em nuvem é a virtualização de servidores. Nesse paradigma, os recursos computacionais são provisionados para os clientes através da virtualização de uma infraestrutura de *Data Center* (DC), de forma que, em nuvens públicas, cada cliente só paga pela quantidade de recursos virtualizados que consumir (*utility computing*) (OLIVEIRA, 2014).

O NIST (BADGER et al., 2012) define três tipos de modelos de negócio para oferta de serviços da computação em nuvem: *Software-as-a-Service* (SaaS), *Plataform-as-a-Service* (PaaS) e *Infraestructure-as-a-Service* (IaaS).

- **SaaS**. O consumidor utiliza a infraestrutura provida pelo fornecedor apenas para consumir aplicações presentes na nuvem. As aplicações podem ser acessíveis a partir de diversos dispositivos clientes, seja por um *browser* ou um aplicativo, por exemplo. O consumidor não pode administrar nem controlar a infraestrutura da nuvem.

- **PaaS**. O consumidor utiliza a infraestrutura da nuvem através de bibliotecas de linguagens de programação suportadas pelo provedor para controlar e utilizar os recursos da nuvem na aplicação desenvolvida. Contudo, o consumidor não pode administrar e nem controlar a rede, servidores e SOs, mas tem controle sobre os aplicativos implementados e possivelmente sobre as configurações para o ambiente de hospedagem de aplicativos.

- **IaaS**. O consumidor tem acesso ao provisionamento de processamento, armazenamento, redes e outros recursos de computação fundamentais, onde é possível implementar e executar *softwares*, que podem incluir desde SOs a aplicações. Porém,

o consumidor não pode administrar nem controlar a infraestrutura de nuvem subjacente, mas tem um controle limitado de componentes de rede selecionados (por exemplo, *firewalls*).

Por fim, o NIST (BADGER et al., 2012) define quatro tipos de nuvem:

- **Privada.** A infraestrutura de nuvem é fornecida para uso exclusivo de uma única organização (por exemplo, uma empresa). Ela pode ser controlada, gerenciada e operada pela organização ou um terceiro.
- **Comunitária.** A infraestrutura de nuvem é fornecida para uma comunidade específica onde os consumidores têm preocupações comuns (por exemplo, grupos de pesquisa). Ela pode ser controlada, gerenciada e operada por uma ou mais organizações da comunidade ou um terceiro.
- **Pública.** A infraestrutura de nuvem é fornecida para uso aberto ao público em geral. Ela pode ser controlada, gerenciada e operada por organização empresarial, acadêmica ou governamental. O fornecedor da nuvem determina as regras de utilização.
- **Híbrida.** A infraestrutura de nuvem é uma composição de duas ou mais infraestruturas de nuvem distintas (privada, comunitária ou pública), mas possuem tecnologia padronizada que permite a portabilidade de dados (por exemplo, balanceamento de carga entre nuvens).

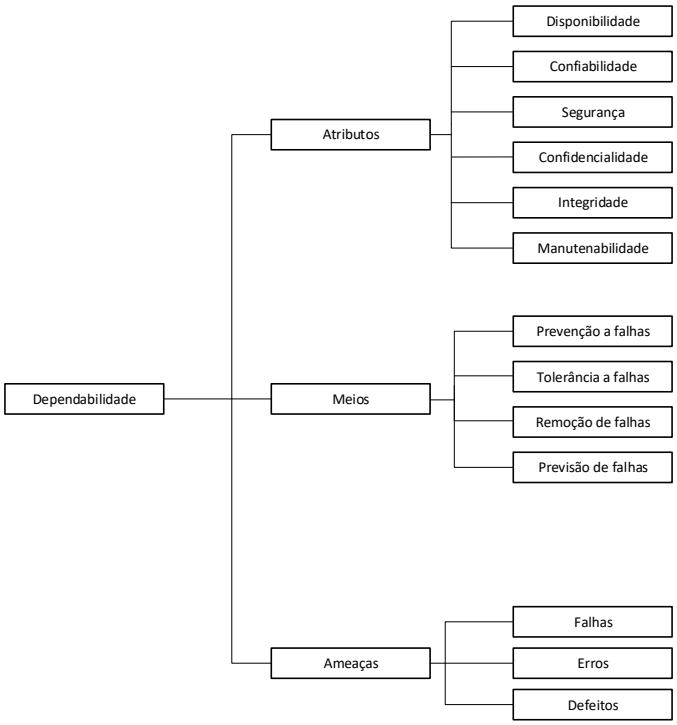
2.2 DEPENDABILIDADE

Devido à grande quantidade de serviços oferecidos através da Internet, a dependabilidade tornou-se um interesse para o desenvolvimento implantação e operação de sistemas (MACIEL et al., 2012). Existem diferentes definições de dependabilidade. Uma das definições trata dependabilidade como a capacidade de oferecer uma funcionalidade específica, que pode ser justificadamente confiável (AVIZIENIS et al., 2004). Uma segunda, afirma que a dependabilidade de um sistema é a execução de ações especificadas ou apresentação de resultados específicos de maneira confiável e em tempo adequado (PARHAMI, 1988).

Avizienis et al. (2004) apresentam de forma sistemática os conceitos de dependabilidade. A Figura 1 exibe esses conceitos que são resumidos a seguir:

- **Atributos** - Possibilitam a obtenção de medidas quantitativas. Geralmente, esses elementos são essenciais para a análise de serviços, sistemas, ou infraestruturas;
- **Meios** - São os meios pelos quais a dependabilidade é atingida;
- **Ameaças** - Ameaças que podem pôr em risco os sistemas.

Figura 1 – Árvore da dependabilidade.



Fonte: Adaptado de (AVIZIENIS et al., 2004)

A dependabilidade compreende os seguintes atributos: disponibilidade (prontidão do sistema para ser usado), confiabilidade (continuidade da prestação do serviço), segurança (ausência de consequências catastróficas), integridade (ausência de alterações no sistema) e manutenção (capacidade de manutenção). Os meios são agrupados em remoção de falhas (reduzir o número ou a severidade das falhas), tolerância a falhas (entregar o serviço correto mesmo na presença de falhas), prevenção de falhas (prevenir a ocorrência ou introdução de falhas) e previsão de falhas (estimar o número presente, a incidência futura e as consequências das falhas). As ameaças são agrupadas em defeito, erros e falhas. A falha (*fault*) de um sistema consiste no comportamento anormal do sistema ou de um subcomponente em relação à sua especificação. Assim, uma falha pode ser causar um erro (*error*). Um erro é a porção do estado do sistema responsável por conduzi-lo à um defeito. Por fim, um defeito (*failure*) é a incapacidade do sistema entregar os serviços especificados (AVIZIENIS et al., 2004).

Este trabalho tem o foco de estudar aspectos de dependabilidade, observando especificamente a disponibilidade em infraestruturas computacionais e estratégias de DR. Assim, na próxima subsecção são discutidos os aspectos relativos ao atributo de disponibilidade.

2.2.1 Disponibilidade

A disponibilidade de um sistema é a probabilidade de que ele esteja operacional durante um determinado período de tempo, mesmo alternando entre estado de defeito e operacional (LAPRIE, 1992). Ela pode ser obtida utilizando-se a relação entre o período de tempo em que o sistema está operacional (*Uptime*) pelo período de tempo de observação do sistema (*uptime + downtime*). O *downtime* representa o período de tempo em que o sistema não está operacional devido à ocorrência de um defeito ou atividade de reparo (XIE; DAI; POH, 2004; KUO; ZUO, 2003). Assim, a Equação 2.1 determina a disponibilidade (A) de um sistema. O valor encontrado por essa equação é representado entre 0 e 1. De forma que, se a disponibilidade de um sistema é igual a *0,99535*, isso indica que o sistema se encontra funcionando durante 99,535% do tempo observado e inativo em 0,465% do tempo observado.

$$A = \frac{uptime}{uptime + downtime} \tag{2.1}$$

A disponibilidade também pode ser expressa em termos de número de noves (N) conforme a Equação 2.2 (XIE; DAI; POH, 2004; KUO; ZUO, 2003). Nesta Equação, o valor "100" representa o nível de disponibilidade máxima que o sistema pode atingir e A representa a disponibilidade do sistema.

$$N = 2 - log(100 - A) \tag{2.2}$$

Outra forma de medir a disponibilidade é através da relação entre o *Mean Time to Fail* (MTTF) e a soma do MTTF com o *Mean Time to Repair* (MTTR) (ver Equação 2.3). Onde o MTTF é o tempo médio para a ocorrência de falhas no sistema, e o MTTR é o tempo médio até o reparo do sistema (XIE; DAI; POH, 2004; KUO; ZUO, 2003). Da mesma forma que na Equação 2.1, o valor encontrado aqui também é representado entre 0 e 1.

$$A = \frac{MTTF}{MTTF + MTTR} \tag{2.3}$$

Formalmente, o MTTF é definido pelo valor médio da função de confiabilidade (XIE; DAI; POH, 2004; KUO; ZUO, 2003):

$$MTTF = \int_0^\infty R(t)dt \tag{2.4}$$

Já o MTTR é formalmente definido por (XIE; DAI; POH, 2004; KUO; ZUO, 2003):

$$MTTR = MTTF \times \frac{UA}{A} \tag{2.5}$$

Onde UA representa a indisponibilidade do sistema e A representa a disponibilidade do sistema. Para se obter a indisponibilidade de um sistema, basta subtrair o valor da disponibilidade encontrada por 1, conforme Equação 2.6.

$$UA = 1 - A \tag{2.6}$$

Além disso, é possível obter a indisponibilidade (D) de um sistema durante um tempo de observação. Para isto, basta realizar a multiplicação da indisponibilidade (UA) pelo tempo de observação do sistema (T), conforme a Equação 2.7.

$$D = UA \times T \tag{2.7}$$

2.3 MODELAGEM DE SISTEMAS

Segundo Johnson e Margalho (2011), "*a modelagem de um sistema é a representação, geralmente simplificada, de um sistema (real) através de relações lógicas e matemáticas com o objetivo de estudar e entender o comportamento desse sistema*". Existem vários tipos de modelos que podem ser utilizados para a avaliação de dependabilidade. *Reliability Block Diagrams* (RBDs) (XIE; DAI; POH, 2004), árvores de falhas (SAHNER; TRIVEDI; PULIAFITO, 1997), *Stochastic Petri Nets* (SPN) (MARSAN, 1990) e *Continuous Time Markov Chain* (CTMC) (TRIVEDI, 1982) têm sido usados para modelar sistemas tolerantes a falhas e avaliar medidas de dependabilidade. Estes tipos de modelo diferem de um para outro não só na facilidade de utilização para uma aplicação em particular, mas em termos de poder de modelagem (MALHOTRA; TRIVEDI, 1994).

Eles podem ser classificados em modelos combinatórios e baseados em estado (MACIEL et al., 2012). Modelos baseados em estado podem também se referir como não combinatórios, e modelos combinatoriais podem ser identificados como modelos não baseados em estados. A seguir, são introduzidos conceitos acerca das redes de Petri e uma extensão, as DSPNs, ambos os modelos baseados em estados, que são usadas nesta tese.

2.3.1 Redes de Petri

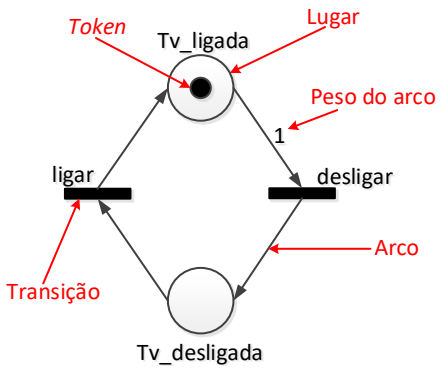
As redes de Petri podem ser definidas como uma técnica de modelagem e análise de sistemas complexos. Ela possui uma representação matemática e mecanismos de análise que permitem a verificação de características de sistemas (PETRI, 1962). Desde sua concepção, elas vem sendo utilizada para representar e analisar diferentes sistemas em várias áreas do conhecimento (ZIMMERMANN; BODE; HOMMEL, 1996; MACIEL; LINS; CUNHA, 1996; CASSANDRAS; LAFORTUNE, 2010; TRIVEDI; GROTTKE; ANDRADE, 2010). O modelo mais básico das redes de Petri é conhecido como *Places/Transition*. Contudo, outras extensões deste modelo surgiram ao longo do tempo, como as redes de Petri temporizadas

(MERLIN; FARBER, 1976), estocásticas (MARSAN, 1990), de alto-nível (JENSEN, 1991) e orientada a objetos (JANOUSEK, 1998).

Um exemplo de uma rede de Petri *Place/Transition* é mostrado na Figura 2. Os seguintes elementos compõem esse tipo de rede de Petri (MURATA, 1989):

- **Token** - Representa o recurso básico de uma rede, necessário para compor uma marcação que aponta o estado em que o sistema se encontra em determinado momento. Graficamente, *tokens* são representados por um círculo preto pequeno. Em alguns casos, também pode ser representado por números.
- **Lugar** - Representam os elementos passivos da rede, tendo como principal função armazenamento de *tokens*, os quais são removidos e adicionados à medida que as transições são disparadas. Graficamente, os lugares são representados por círculos brancos de borda preta.
- **Transição** - Representa as ações realizadas pelo sistema e, portanto, causam a mudança da marcação (estado) da rede. Graficamente, são representadas por barras retangulares.
- **Arco** - Representa o fluxo dos *tokens* pela rede. Um arco é dito *arco de entrada* quando tem origem em um lugar e destino uma transição. Já quando tem uma transição como origem e um lugar como destino é chamado de *arco de saída*. O arco também pode ter um peso associado, indicado por um número próximo ao arco (quando não existir essa notação, o peso é igual a 1), que reflete na quantidade de *tokens* gerados ou consumidos. Caso seja um arco de entrada, irá ser **consumida** a quantidade de *tokens* referente ao peso do arco. Porém, se for um arco de saída, irá ser **gerada** a quantidade de *tokens* referente ao peso do arco.

Figura 2 – Exemplo de uma rede de Petri *Place/Transition*.



Fonte: Criado pelo autor.

Uma rede de Petri não permite a ligação por um arco entre elementos do mesmo tipo, ou seja, um lugar ligado a um lugar ou uma transição ligada a outra transição. Além disso, para ser disparada, uma transição t precisa estar habilitada. Uma transição t está habilitada quando cada um dos lugares conectados a ela por um arco de entrada possuem uma quantidade de *tokens* igual ou superior ao peso de cada arco de entrada correspondente. O disparo da transição causa mudança na marcação da rede de Petri, o que indica uma mudança de estado do sistema ou de um elemento. O disparo faz com que os *tokens* sejam consumidos, e novos *tokens* sejam gerados nos lugares conectados à transição por arcos de saída.

No exemplo da Figura 2, a presença do *token* no lugar *Tv_ligada* indica o estado atual do elemento, neste caso, que a TV está ligada. A transição *desligar* está habilitada, pois o arco de entrada possui peso 1, o mesmo número de *tokens* presente no lugar *Tv_ligada*. Quando é disparada, a transição *desligar* consome o *token* presente no lugar *Tv_ligada* e gera 1 *token* no lugar *Tv_desligada*. A partir deste momento, a transição *ligar* está habilitada, e quando é disparada consome o *token* do lugar *Tv_desligada*, e gera 1 *token* no lugar *Tv_ligada*, pois o peso do arco de saída é igual a 1.

Formalmente, uma rede de Petri é definida como segue (MURATA, 1989):

Definição 2.3.1. Uma rede de Petri é uma quíntupla $R = (P, T, F, W, M_0)$, onde:

- $P = \{p_1, p_2, ..., p_n\}$, é um conjunto finito não-vazio de lugares;
- $T = \{t_1, t_2, ..., t_n\}$, é conjunto finito não-vazio de transições, $P \cap T = \varnothing$;
- $F \subseteq (P \times T) \cup (T \times P)$, é um conjunto de arcos;
- $W : F \longrightarrow IR^+ \cup \{0\}$, é a função de atribuição de peso aos arcos;
- $M_0 : P \longrightarrow IN$, é a função de marcação inicial, onde $P \cap T = \varnothing$ e $P \cup T \neq \varnothing$.

2.3.1.1 Propriedades das Redes de Petri

O estudo das propriedades de redes de Petri permite a análise do sistema modelado. Os tipos de propriedades podem ser divididos em duas categorias: as propriedades dependentes de marcação inicial, conhecidas como propriedades comportamentais, e as propriedades não dependentes de marcação, conhecidas como propriedades estruturais (MACIEL; LINS; CUNHA, 1996; MURATA, 1989).

Propriedades Comportamentais

As propriedades comportamentais são aquelas que dependem apenas da marcação inicial da rede de Petri. As propriedades apresentadas são alcançabilidade, limitação, segurança, vivacidade, cobertura, reversibilidade e *home state* (MURATA, 1989).

Alcançabilidade (ou *reachability*) - Indica a possibilidade de uma determinada marcação ser atingida pelo disparo de um número finito de transições a partir de uma marcação inicial. Dada uma rede de Petri marcada $RM = (R, M_0)$, o disparo de uma transição t_0 altera a marcação da rede de Petri. Uma marcação M' é acessível a partir de M_0 se existe uma sequência de transições que, disparadas, levam à marcação M_0. Ou seja, se a marcação M_0 habilita a transição t_0, disparando-se essa transição, atinge-se a marcação M_1. A marcação M_1 habilita t_1 a qual, sendo disparada, atinge-se a marcação M_2 e assim por diante até a obtenção da marcação M'. Assim, a sequência de disparo é denotada pelo conjunto $\sigma = t_1, t_2, ..., t_n$. Nesse caso, M' é alcançável a partir de M_0 por σ. Onde σ é formalmente descrito por $M_0[\sigma > M'$ (M_0 que habilita M' através da sequência de disparados do conjunto de transições σ).

Limitação - É uma propriedade que diz que uma rede de Petri é *k-limitada* se todos os seus lugares forem limitados. Um lugar é dito limitado se o número de *tokens* não ultrapassar um número finito k, para qualquer marcação alcançável a partir de M_0.

Segurança (ou *safeness*) - É uma particularização da propriedade de limitação. O conceito de limitação define que um lugar p_i é *k-limitado* se o número de *tokens* que esse lugar pode acumular estiver limitado ao número k. Um lugar que é *1-limitado* pode ser simplesmente chamado de seguro.

Vivacidade (ou *liveness*) - É definida em função das possibilidades de disparo das transições. Uma rede de Petri é considerada *live* se, independente das marcações que sejam alcançáveis a partir de M_0, for sempre possível disparar qualquer transição da rede de Petri através de uma sequência de transições $L(M_0)$. A ausência de bloqueio (*deadlock*) em sistemas está fortemente ligada ao conceito de vivacidade, pois o *deadlock* em uma rede de Petri é a impossibilidade do disparo de qualquer transição da rede. O fato de um sistema ser livre de *deadlock* não significa que seja *live*, entretanto um sistema *live* implica em um sistema livre de *deadlocks*. A análise de vivacidade de uma rede permite verificar se os eventos modelados efetivamente ocorrem durante o funcionamento do sistema ou se foram definidos eventos mortos no modelo. A propriedade vivacidade ainda pode ser definida em níveis, conforme apresentados abaixo:

- **L0-*live* (morta)** - Se uma transição t nunca pode ser disparada em qualquer sequência $L(M_0)$;
- **L1-*live* (potencialmente disparável)** - Se uma transição t pode ser disparável em pelo menos alguma sequência $L(M_0)$;
- **L2-*live*** - Dado um inteiro positivo k, se uma transição t puder ser disparada pelo menos k vezes em alguma sequência $L(M_0)$;
- **L3-*live*** - Se uma transição t aparece um número infinito de vezes em alguma sequência de disparo $L(M_0)$;

- L**4-*live* (ou simplesmente *live*)** - Se uma transição t é potencialmente disparável para todas as marcações da rede.

Cobertura - Está fortemente conectada ao conceito de alcançabilidade e vivacidade. Quando se deseja saber se alguma marcação M_i pode ser obtida a partir de uma marcação M_j, temos o problema denominado cobertura de uma marcação. Uma marcação M_i é dita coberta se existe uma marcação M_j tal que $M_j > M_i$. Fora isso, em alguns sistemas, deseja-se apenas observar o comportamento de determinados lugares. Para isso, restringe-se a pesquisa a apenas um conjunto de lugares de particular interesse (cobertura de submarcações).

Reversibilidade e *Home state* - Uma rede é dita reversível se, para cada marcação M em $R(M_0)$, M_0 é alcançável a partir de M. Assim, a rede possui a capacidade de retornar à marcação inicial. Além disso, em algumas aplicações não é necessário voltar à marcação inicial, mas sim a uma marcação específica. Essa marcação específica é denominada *Home state*.

Propriedades Estruturais

As propriedades estruturais são aquelas que dependem apenas da estrutura da rede de Petri. Essas propriedades refletem características independentes de marcação da rede. A seguir, são apresentadas as propriedades estruturais das redes de Petri (MARSAN et al., 1994):

- **Estruturalmente Limitada** - Uma rede de Petri é classificada como estruturalmente limitada se for limitada para qualquer marcação inicial, ou seja, se o número de *tokens* é limitado para qualquer marcação inicial.
- **Conservação** - Permite a verificação da não destruição de recursos através da conservação de *tokens*. Assim, com o disparo de qualquer transição o número de marcações não é alterado.
- **Repetitividade** - Uma rede é classificada como repetitiva se, para uma marcação e uma sequência de transições disparáveis, todas as transições dessa rede são disparadas ilimitadamente.
- **Consistência** - Uma rede é consistente se, a partir de uma sequência de transições disparáveis partindo-se de uma marcação inicial M_0, ela retorna a M_0 tendo todas as transições da rede disparadas pelo menos uma vez.

2.3.2 Extensões Temporizadas de Redes de Petri

O conceito de tempo não foi definido originalmente para as redes de Petri *Place/Transition*. No entanto, para realização de análises em muitas aplicações, se faz necessário a utilização

do tempo. Sem uma notação para o tempo, não é possível a realização de análises de desempenho (AALST; HEE; REIJERS, 2000). Nestes tipos de redes de Petri, o tempo pode estar relacionado a (MACIEL; LINS; CUNHA, 1996):

- **Lugares** - Onde *tokens* podem ter de permanecer naquele lugar por um tempo determinado.
- **Tokens** - Onde estes possuem uma informação que indica se pode ser consumido ou não.
- **Transições** - Quando habilitadas devem respeitar esse tempo como um atraso para o disparo.

Na maioria das extensões, o tempo é atribuído às transições. Quanto ao tipo de tempo associado às transições, eles podem ser (AALST; HEE; REIJERS, 2000):

- **Determinísticos** - Indicam o tempo exato de *delay* da transição, o que em termos de aplicação condiz com duração de uma atividade/evento;
- **Não-determinísticos** - Geralmente utilizam um intervalo para especificar a duração do *delay*;
- **Estocásticos** - O tempo é descrito através de distribuições de probabilidade (ex.: normal, exponencial, ou F).

Outro conceito associado à atribuição de tempo às redes de Petri é o grau de habilitação. Ele indica o número de vezes que uma transição pode ser disparada. Estas semânticas de disparo podem ser do tipo (CASSANDRAS; LAFORTUNE, 2010):

- **Single-Server** - Apenas um *token* é disparado por vez, ou seja, a capacidade de um lugar/transição é 1.
- **Multiple-Server** - É possível fazer k disparos por vez, ou seja, a capacidade de um lugar/transição é um número inteiro k.
- **Infinite-Server** - É possível fazer infinitos disparos de uma única vez.

Vale mencionar ainda o conflito entre transições quando se trabalha com transições temporizadas. Quando a habilitação e disparo de uma transição causa a desabilitação de uma outra transição, existe a situação de conflito. Para resolver este problema, existem três tipos de abordagens (AALST; HEE; REIJERS, 2000):

- **Age Memory** - O tempo que resta para a transição ser habilitada é congelado no momento que a transição se torna desabilitada, e é retomado no momento em que a transição se torna habilitada novamente;

- **Enabling Memory** - A cada disparo de uma transição, os tempos atribuídos as transições desabilitadas na nova marcação são reiniciados. Por outro lado, são mantidos os valores dos tempos atribuídos à todas transições que continuam habilitadas na nova marcação.
- **Reset Memory** - Em todos os disparos de transições, os tempos atribuídos à todas as transições são descartadas. Na nova marcação, um novo tempo é atribuído para cada transição habilitada.

2.3.2.1 *Deterministic and Stochastic Petri nets*

A DSPN (MARSAN; CHIOLA, 1987; GERMAN, 2000) é uma extensão das redes de Petri *Place/Transition* onde existem transições temporizadas (determinísticas e estocásticas) e transições imediatas. Os tempos de disparado das transições estocásticas são exponencialmente distribuídos, por esse motivo, são também chamadas de transições exponenciais.

Formalmente, uma DSPN é definida por (GERMAN, 2000):

Definição 2.3.2. Uma DSPN é uma tupla $R = (P, T, I, O, H, \Pi, G, M_0, Atts)$, onde:

- $P = \{p_1, p_2, ..., p_n\}$ é um conjunto finito não-vazio de lugares;
- $T = \{t_1, t_2, ..., t_n\}$ é um conjunto finito não-vazio de transições;
- $I \in (\mathbb{N}^n \longrightarrow \mathbb{N})^{n \times m}$ é a matriz que representa os arcos de entrada (podem ser dependentes de marcação);
- $O \in (\mathbb{N}^n \longrightarrow \mathbb{N})^{n \times m}$ é a matriz que representa os arcos de saída (podem ser dependentes de marcação);
- $H \in (\mathbb{N}^n \longrightarrow \mathbb{N})^{n \times m}$ é a matriz que representa os arcos inibidores (podem ser dependentes de marcação);
- $\Pi \in \mathbb{N}^m$ é o vetor que associa o nível de prioridade para cada transição;
- $G \in (\mathbb{N}^n \longrightarrow \{true, false\})^m$ é o vetor que associa uma função de guarda para cada transição que pode depender da marcação atual. Uma função de guarda atribui uma função booleana a uma transição da DSPN. O resultado de uma função de guarda controla o disparo de uma transição. Por exemplo, uma função de guarda definida por $\#P_0 > 1$ atribuída a uma transição t_n indica que a transição t_n só será disparada caso o número de *tokens* no lugar P_0 for maior que um;
- $M_0 \in \mathbb{N}^n$ é o vetor que define a marcação inicial;
- $Atts = (Dist, W, Markdep, Policy, Concurrency)^m$ é definido pelo conjunto de atributos:

i. $Dist \in \mathbb{N}^n \longrightarrow \mathcal{F}$ é uma função de distribuição de probabilidade associada ao tempo de uma transição, sendo que o domínio de $\mathcal{F}$ é $[0, \infty]$ (a distribuição pode ser dependente da marcação);

ii. $W \in \mathbb{N}^m \longrightarrow \mathbb{R}^+$ é a função peso, que associa um peso (w_t) às transições imediatas e uma taxa λ_t às transições temporizadas, onde:

$$W(t) = \begin{cases} w_t \geq 0, & \text{se } t \text{ é uma transição imediata,} \\ \lambda_t > 0, & \text{caso contrário.} \end{cases}$$

iii. $Markdep \in \{constant, enabdep\}$ define se a distribuição de probabilidade associada ao tempo de uma transição é constante (*constant*) ou dependente de marcação (*enabdep*);

iv. $Policy \in \{prd, prs\}$ é a política de preempção adotada pela transição (*prd - preemptive repeat different* - corresponde à *reset memory* e *prs - preemptive resume* - possui significado idêntico à *age memory policy*);

v. $Concurrency \in \{ss, is\}$ é o grau de concorrência das transições, onde *ss* representa a semântica de *Single-Server* e *is* significa a semântica de *Infinite-Server*.

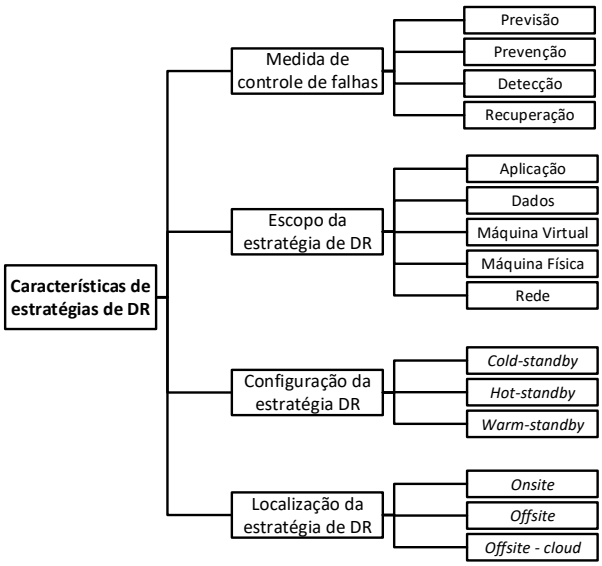
Para mais informações sobre redes de Petri e suas extensões, os leitores podem se referir a Murata (1989), Marsan (1990), Maciel, Lins e Cunha (1996), Aalst, Hee e Reijers (2000), Marsan e Chiola (1987), German (2000), Jensen (1995) e Sifakis (1977).

2.4 RECUPERAÇÃO DE DESASTRES

Infraestruturas computacionais e sistemas de TIC são suscetíveis a um conjunto de interrupções brandas (ex.: interrupção de energia ou falha de discos) e severas (ex.: incêndio ou terremoto) (SWANSON et al., 2010). Algumas das causas dessas interrupções severas podem ser classificadas como desastres (REESE, 2009). As interrupções severas podem ser eliminadas, ou pelo menos minimizadas, através de estratégias de garantia de qualidade (ex.: testes ou revisões). Apesar de ser praticamente impossível eliminar todos os riscos de ocorrência de interrupções, a utilização de estratégias de DR pode mitigar ou contingenciar essas interrupções. O uso de tais estratégias pode auxiliar os sistemas de TIC a evitar perda de dados e/ou diminuir o tempo para a recuperação dos serviços depois de uma interrupção. Assim, as estratégias de DR podem atuar na **contingência** ou **mitigação** das interrupções (ANDRADE et al., 2017). Enquanto estratégias de contingenciamento focam na realização de ações para lidar com interrupções caso elas se concretizem, as de mitigação buscam a realização de ações para reduzir a probabilidade de ocorrência dessas interrupções.

A ocorrência de desastres (ex.: atentados terroristas, incêndio, tsunami ou vandalismo) podem causar falhas em uma infraestrutura computacional (ex.: um DC), que podem gerar

Figura 3 – Características de estratégias de DR.



Fonte: Adaptado de (MENDONçA et al., 2019a).

defeitos, ocasionando interrupções severas. Isto pode resultar na indisponibilidade dos sistemas e serviços da infraestrutura computacional afetada. Visto que eventos de desastres podem prejudicar drasticamente os sistemas de TIC, estratégias de DR são implantadas para proteger os serviços críticos (BAUER; ADAMS; EUSTACE, 2011). Estratégias de DR são adotadas para tratar as interrupções das infraestruturas computacionais e podem funcionar tanto para contingência quanto para mitigação de interrupções brandas ou severas.

Diferentes estratégias de DR podem ter objetivos distintos como, por exemplo, aumentar a disponibilidade e proteger os dados da empresa. Os objetivos da adoção de uma estratégia de DR variam de empresa para empresa. A Figura 3 apresenta uma forma de classificar as estratégias de DR. Ou seja, pode-se classificar uma estratégia de DR quanto à medida de controle de falha, a localização da estratégia, a configuração e o escopo de atuação (MENDONçA et al., 2019a). As *medidas de controle de falhas* definem se uma estratégia atua na previsão, prevenção, detecção, ou recuperação de falhas/desastres (AVIZIENIS et al., 2004). A classificação de *localização* define se a estratégia de DR encontra-se no mesmo local da infraestrutura primária (*onsite*) ou em uma localização diferente da infraestrutura primária (*offsite*). Há ainda a definição para quando a estratégia de DR encontra-se em um ambiente de nuvem (*offsite-cloud*). Também é possível classificar a *configuração* da estratégia de DR. Isto é, se ela atua em modo *hot-standby*, *warm-standby*, ou *cold-standby*. Por fim, ainda podem-se classificar as estratégias de acordo com seu *escopo de atuação*, ou seja, se o foco é prover capacidade de DR para as aplicações, dados, VMs, máquinas físicas ou rede.

As classificações das estratégias de DR visam facilitar sua identificação e entendimento. Cada estratégia apresenta desvantagens e vantagens, de modo que as organizações sempre

buscam um equilíbrio entre as principais características das estratégias de DR. Assim, a escolha correta de uma estratégia é fundamental para atender as necessidades e ao orçamento das empresas. Nas próximas subseções, são discutidas a avaliação das estratégias de DR e, em seguida, a implementação de algumas estratégias.

2.4.1 Avaliação das estratégias de DR

O planejamento adequado é crucial para o sucesso na implementação de qualquer estratégia de DR. Dessa forma, a avaliação eficiente e precisa das soluções de DR é fundamental para garantir a continuidade do negócio e a recuperação de desastres. A avaliação dessas estratégias é em grande parte suportada através de experimentos ou modelagem (MENDONçA et al., 2019a). Enquanto experimentos geralmente apresentam um maior custo (financeiro e de tempo), a modelagem é considerada mais viável quando se deseja testar vários cenários e novas abordagens (JAIN, 1991).

Os motivos para a adoção de estratégias de DR são numerosos e, consequentemente, os objetivos que se deseja alcançar através de sua implementação também podem variar. Assim, conhecer e avaliar as métricas de interesse corretamente faz parte de um bom planejamento para alcançar os objetivos desejados com a implantação de uma estratégia de DR. Um estudo realizado por Mendonça et al. (2019a) apontou as principais métricas utilizadas em trabalhos que abordam o tema de DR (ver Tabela 1). No levantamento realizado, as métricas de RTO, RPO, custo, disponibilidade e indisponibilidade foram as mais adotadas por esses trabalhos.

Tabela 1 – Métricas mais utilizadas em estudos de DR.

Posição	**Métrica**
1ª	RTO
2ª	RPO
3ª	Custo
4ª	Disponibilidade
5ª	Indisponibilidade

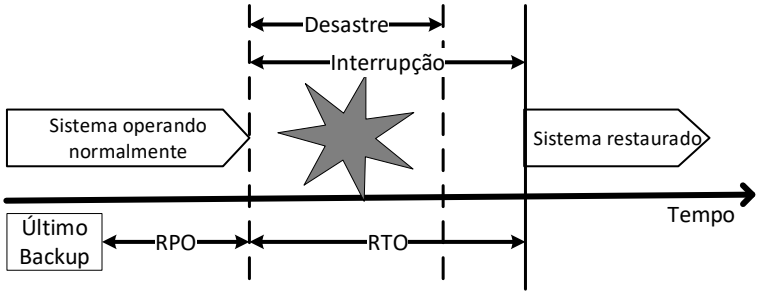
Fonte: Adaptado de Mendonça et al. (2019a).

O RTO e o RPO são cruciais na avaliação de estratégias de DR. O RTO compreende o tempo máximo para trazer um sistema ou aplicativo ao seu estado operacional, enquanto o RPO compreende a quantidade máxima de dados que podem ser perdidos com o processo de recuperação (KEETON et al., 2004). Vale ressaltar que o RPO também pode ser medido em tempo, representando o intervalo de tempo decorrido entre o último ponto de recuperação (ex.: *backup*) dos dados e o momento de ocorrência do desastre. A Figura 4 ilustra esses objetivos no contexto do acontecimento de um desastre. O sistema está operando normalmente, quando um desastre ocorre (ex.: inundação) e causa a indisponibilidade

do serviço. Assim, no tempo decorrido desde o fim do último *backup* de dados até o acontecimento do desastre, que causa a interrupção do sistema, dados serão perdidos. Esse tempo ou quantidade de dados perdidos é o RPO. Por outro lado, o tempo que decorre desde o início da interrupção do sistema, devido ao desastre sofrido, até sua restauração, define o RTO.

Outras métricas como custo, disponibilidade e indisponibilidade, também são importantes para avaliação de estratégia de DR. A análise de custo de uma estratégia de DR é capaz de oferecer uma comparação de diversos pontos importantes como, por exemplo, custo de manutenção, de operação, e de produtividade. As métricas de disponibilidade e indisponibilidade também têm um papel essencial, especialmente se o serviço estiver atrelado a um *Service-level Agreement* (SLA). Um SLA define penalidades quando uma métrica não é cumprida pelo provedor de serviços. Assim, a disponibilidade e indisponibilidade de um serviço ou infraestrutura podem impactar diretamente na produtividade e operação das empresas. Por fim, vale ressaltar que diferentes estratégias de DR, por possuírem diferentes características e comportamentos, impactam todas essas métricas de diferentes formas (MENDONçA et al., 2019a).

Figura 4 – Objetivos na recuperação de desastres.



Fonte: Adaptado de Andrade (2014).

2.4.2 Estratégias para recuperação de desastres

Esta subseção introduz as estratégias de DR adotadas neste trabalho. Essas estratégias provêm benefícios como segurança de dados, aumento da disponibilidade e, principalmente, meios para retomar as operações após interrupções brandas ou severas. A seguir, são apresentadas três estratégias: o *backup*, a replicação de BDs e a replicação de ambiente.

2.4.2.1 *Backup*

As tecnologias de *backup* e restauração já foram a base das estratégias de proteção de dados para ajudar as organizações a atender aos seus requisitos de disponibilidade e acessibilidade de dados. O *backup* e a restauração de dados são umas das principais

atividades operacionais do gerenciamento de dados corporativos (MICROSOFT, 2018). A operação de *backup* consiste na cópia de dados para um dispositivo de armazenamento. Geralmente, utiliza-se um dispositivo de armazenamento secundário ou terciário (ex.: DVD-ROM ou *Hard Disks* (HDs) externos) utilizado para manter uma cópia dos dados em segurança. Ao longo dos anos, os dispositivos de armazenamento passaram por uma evolução. Os dispositivos mais comuns para armazenar os *backups* evoluíram de fitas, CD-ROM, DVD-ROM, e HDs externos a *backups* através da Internet. A chegada da computação em nuvem facilitou bastante a adoção do último.

As operações de *backup* se dividem em duas estratégias principais (PRESTON, 2007): *Backup* completo e *Backup* parcial. A modalidade de *backup* parcial ainda é subdivida em dois tipos: a incremental e a diferencial. No *backup* completo é realizada a cópia completa de todos os dados escolhidos, enquanto nos *backups* parciais é efetuada apenas a cópia de parte dos dados, de acordo com o tipo escolhido. No *backup* parcial, a parte dos dados a ser copiada corresponde à diferença de dados em relação ao último *backup* realizado. Sendo assim, um *backup* parcial nunca poderá ser o primeiro tipo de *backup* a ser realizado em um sistema.

Os dois tipos de *backup* parcial possuem pequenas diferenças entre eles. O *backup* incremental realiza a cópia de arquivos adicionados ou modificados desde o último *backup*, seja ele completo ou parcial. Já o *backup* diferencial realiza a cópia de arquivos adicionados ou modificados desde a realização do último *backup* completo. Os *backups* ainda podem ser classificados quanto à permissão de acesso aos dados que serão copiados. Essas classificações são: *backup online* e *backup offline*. No primeiro, a execução da tarefa de *backup* pode ser realizada enquanto os dados estão acessíveis aos usuários. Já no *backup offline*, os dados que serão copiados não estão acessíveis para o usuário (CHERVENAK; VELLANKI; KURMAS, 1998).

O processo de restauração dos dados também apresenta diferença de acordo com o tipo de *backup* escolhido. Na restauração de um *backup* completo, todos os dados são restaurados em uma única etapa. Porém, para restaurar *backups* parciais, faz-se necessário realizar a restauração em duas etapas. Na restauração do *backup* incremental, é necessária a restauração do último *backup* completo e, em seguida, a restauração de todos os *backups* incrementais na ordem correta. Já na restauração do *backup* diferencial, é necessária a restauração do último *backup* completo e, em seguida, apenas a restauração do último *backup* diferencial.

2.4.2.2 Replicação de Banco de dados

Assim como as operações de *backup*, a replicação de BD evita a perda dos dados. No entanto, esse auxílio é dado de uma forma mais específica, pois apenas parte dos dados do sistema é coberto nessa estratégia. Na replicação de BDs, os dados e a estrutura do BD são compartilhados entre um BD principal e instâncias que podem estar em locais

diferentes (MAZILU, 2010). Assim, se uma falha ocorre no BD principal, uma outra instância pode assumir as operações, pois possui os dados para isso. No uso dessa estratégia, o BD principal que recebe as requisições diretamente dos sistemas ou usuários é chamado de *master*. Já o BD que recebe a cópia das transações é denominado réplica. Existem 3 modelos para a replicação de BDs (MICROSOFT TECHNET, 2018):

- ***Snapshot*** - Neste tipo de replicação, os dados são distribuídos exatamente como aparecem em um momento específico e as atualizações dos dados não são monitoradas. Assim, em um dado momento, uma cópia dos dados do *master* é gerada e enviada às réplicas.

- ***Transactional*** - Esse modelo geralmente inicia com uma cópia dos objetos e dos dados do BD *master*. Posteriormente, as alterações de dados subsequentes e as modificações de esquema feitas no BD *master* são enviadas às réplicas à medida que ocorrem (quase em tempo real). As alterações de dados são aplicadas às réplicas na mesma ordem e dentro dos mesmos limites da transação que ocorreu no BD *master*, de modo a garantir a consistência transacional.

- ***Merge*** - Como na replicação *transactional*, esse modelo geralmente também inicia com uma cópia dos objetos e dos dados do BD *master*. Porém, esse modelo considera que as réplicas também podem receber atualizações de dados ou modificações de esquema, independentemente do BD *master*. Assim, as alterações dos dados e as modificações de esquema aplicadas ao BD *master* e as réplicas são rastreados com *triggers*. Os BDs sincronizam quando conectados à rede, de modo que é realizada a atualização das informações que foram modificadas entre o *master* e as réplicas desde a última sincronização.

No modelo de replicação *transacional* para BDs relacionais, vale ressaltar ainda os diferentes métodos de sincronização para a replicação de BDs que existem para cada Sistema Gerenciador de Banco de Dadoss (SGBDs). Considerando o MySQL 8.0 (ORACLE, 2020), o processo de sincronização é realizado em três principais etapas: ***(1)*** ao receber uma transação, o *master* salva a transação em seu *binary log*[1] e a envia à(s) *replica(s)*; ***(2)*** a(s) *replica(s)*, ao receber a transação, escreve a mesma no seu *relay log*[2]; e ***(3)*** uma ou mais *applier threads* aplicam a transação na(s) *replica(s)*, salvando-a em seu *binary log*, persistindo os dados (realizando o *commit* dos dados) em seguida. Dentre os diferentes métodos de sincronização para BDs relacionais, podem-se destacar as configurações oferecidas pelo MySQL 8.0 (Oracle, 2020b):

[1] O *binary log* contêm as informações de todas as transações e mudança de dados realizadas (Oracle, 2020).

[2] O *relay log* funciona como uma fila de transações a serem persistidas na réplica (Oracle, 2020c).

- **Assíncrona** - Nesta configuração, o *master* persiste uma transação sem nenhum tipo de sincronização com as réplicas. O *master* envia a transação para as réplicas, mas não espera pela sincronização antes de realizar o *commit* da transação. Se o *master* por algum motivo parar, pode resultar em perda de dados.

- **Semisíncrona** - Essa configuração, que tem nomenclatura definida pela Oracle (2020b), aguarda que um número determinado de réplicas receba a transação. Assim, a transação no *master* só é persistida depois que as réplicas enviam uma confirmação (*ack*) de recebimento da transação ou o tempo limite de espera da confirmação é atingido. Dessa forma, não há perda de dados, pois o *master* apenas confirma a transação após receber a confirmação da aplicação da transação pelas réplicas. Por padrão, quando o tempo limite para espera da confirmação de recebimento das transações pelas réplicas é atingido ou as réplicas falham, o *master* passa a operar em modo assíncrono.

2.4.2.3 Replicação de ambiente

Os sistemas de TIC necessitam de uma infraestrutura computacional para operar. Assim, o ambiente necessário para execução destes sistemas é configurado sob essa infraestrutura computacional. A estratégia de replicação de ambiente visa oferecer a implantação de uma segunda infraestrutura, idêntica à infraestrutura original (primária). Geralmente, a réplica da infraestrutura original é chamada de infraestrutura secundária. A implantação de uma infraestrutura secundária têm por objetivo principal manter os sistemas executando normalmente, mesmo que falhas ou desastres ocorram na infraestrutura primária. Para evitar que duas infraestruturas sejam afetadas pelo mesmo tipo de desastre (ex.: terremoto), comumente a infraestrutura secundária é montada em um local físico diferente da infraestrutura primária.

Sem um modelo de redundância, uma infraestrutura funciona em um esquema simples, ou seja, existe ao menos um componente com um ponto único de falha. Qualquer falha em tais componentes resultará inevitavelmente na indisponibilidade dos sistemas hospedados naquela infraestrutura. A utilização de um modelo de redundância torna possível manter (ou mover) o ambiente necessário, como sistemas de arquivos, bancos de dados e endereços de *Internet Protocol* (IP) para a infraestrutura secundária. Assim, tornando possível a execução dos sistemas na infraestrutura secundária. Dessa forma, a implantação de uma infraestrutura secundária com um modelo de redundância resulta em maior disponibilidade dos sistemas. Os seguintes modelos de redundância são geralmente adotados (BAUER; ADAMS; EUSTACE, 2011):

- **Active-Active**: têm-se duas infraestruturas computacionais, ambas compartilham a carga de trabalho quando operam em situação normal. Ou seja, a infraestruturas primária e secundária recebem requisições dos usuários e as processam.

- **Active-Standby**: têm-se duas infraestruturas, uma infraestrutura ativa (primária) será a responsável por processar todas as requisições de usuários destinado ao serviço. A infraestrutura secundária não processa requisições dos usuários enquanto a infraestrutura primária for capaz de fazê-lo. Assim, a infraestrutura secundária só passa a processar requisições dos usuários quando uma falha ocorre na infraestrutura primária. Dessa forma, são definidos diferentes graus de prontidão aplicáveis a infraestrutura secundária para passar a processar as requisições, que podem ser classificados como:

  - **Hot-Standby**: a infraestrutura secundária está ligada, pronta para receber carga de trabalho, e seus dados são mantidos em sincronia com a infraestrutura primária;
  - **Warm-Standby**: a infraestrutura secundária está ligada, mas não está pronta para receber carga de trabalho. Assim, a infraestrutura secundária necessita de um tempo para ativação. Além disso, seus dados só são sincronizados periodicamente com a infraestrutura primária;
  - **Cold-Standby**: a infraestrutura secundária está desligada e não pode receber carga de trabalho. Geralmente, requer um tempo de ativação maior que a configuração *warm-standby*. Além disso, não há sincronia de dados entre as infraestruturas quando essa configuração é adotada.

No modelo *active-standby*, ainda há duas formas possíveis de ativar a infraestrutura secundária: manual (*switchover*) ou automática (*failover*). O *switchover* requer mais tempo para ativação da infraestrutura secundária, visto que configurações e comandos precisam ser realizados por um profissional. Já no processo de *failover*, o sistema já é configurado para, ao detectar a falha na infraestrutura primária, encaminhar as requisições dos usuários para a infraestrutura secundária (BENZ; BOHNERT, 2014).

2.5 MÉTODOS DE DECISÃO MULTICRITÉRIO

Métodos de decisão multicritério (ou métodos *Multi-criteria decision-making* (MCDM)) consideram o uso de mais de um critério (requisito) para avaliação de alternativas (ZIONTS, 1979; ROY, 2013). Geralmente, métodos MCDM são utilizados para compor modelos de decisão e auxiliar no processo de seleção e ranqueamento de alternativas (JATOTH et al., 2019). Existem vários métodos MCDM que podem ser usados para auxiliar na seleção e ranqueamento de alternativas. O AHP (SAATY, 2005) e a TOPSIS (HWANG; YOON, 1981) são métodos bastante estabelecidos na literatura. Embora o AHP seja uma abordagem eficiente para a tomada de decisões, ele não considera a incerteza do processo decisório em comparações pareadas (JATOTH et al., 2019). Já a TOPSIS é uma técnica mais simples, se comparada a outros MCDMs, e não possui limitação no número de alternativas ou

critérios (JATOTH et al., 2019) a serem avaliados. Em seguida, apresentamos conceitos sobre medidas de dissimilaridade e normalização de dados uma vez que são conceitos necessários para implementação da TOPSIS, a técnica escolhida para análise multicritério nesta tese. Por fim, é detalhado o funcionamento da TOPSIS.

2.5.1 Medidas de dissimilaridade

As medidas de (dis)similaridade são utilizadas para organizar valores em grupos homogêneos. A similaridade entre duas amostras pode ser expressa como uma função da distância entre dois pontos representativos das amostras no espaço n-dimensional. Uma medida pode ser de *similaridade* ou de *dissimilaridade*. Para as medidas de *similaridade*, quanto maior o valor observado, mais parecidos são os objetos (ex.: o coeficiente de correlação). Para medidas de *dissimilaridade*, quanto maior o valor observado, menos parecidos (mais dissimilares) são os objetos (ex.: distância Euclidiana). A escolha de uma medida é uma decisão importante e leva em conta diversos fatores. Um dos fatores mais importantes envolve os tipos dos dados à disposição (quantitativo, qualitativo nominal, qualitativo ordinal, entre outros) (PADILHA; CARVALHO, 2017). Geralmente para objetos com atributos quantitativos são adotadas medidas de dissimilaridade baseadas na distância Euclidiana (PADILHA; CARVALHO, 2017).

A distância Euclidiana entre duas amostras (i e j) é dada pela raiz quadrada do somatório dos quadrados das diferenças entre os valores de i e j para todas as variáveis ($r = 1, 2, \ldots, n$). A distância Euclidiana é dada pela Equação 2.8 (PADILHA; CARVALHO, 2017).

$$\text{dist}\,(x_i, x_j) = \left[\sum_{r=1}^{m} (x_{i,r}, -x_{j,r})^2\right]^{\frac{1}{2}} \tag{2.8}$$

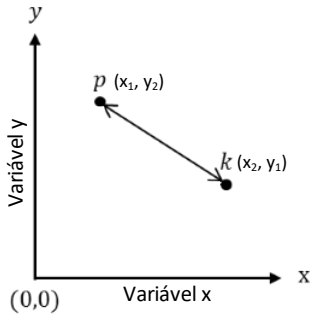
onde $x_{i,r}$ representa a característica da variável i, $x_{j,r}$ representa a característica da variável j, m é o número de parcelas na amostra e r é o tamanho da amostra. A distância Euclidiana é recomendada para agrupamentos centroides[3] (HAIR et al., 2009). A Figura 5 ilustra um exemplo da distância Euclidiana considerando duas amostras.

2.5.2 Normalização de dados

O uso de dados com medidas ou escalas diferentes pode distorcer a realidade do agrupamento realizado. Grande parte das medidas utilizadas para a análise de agrupamento sofre influência das diferentes escalas dos dados de entrada (FÁVERO et al., 2009). Esse problema é contornado com a normalização dos dados. Uma forma bastante utilizada de normalização dos dados é a técnica chamada *Range 0 à 1* (ou *Min-Max*). Esse método faz com que o valor da variável seja normalizado e apresentado entre zero (0) e um (1). A

[3] Ponto associado a uma forma geométrica.

Figura 5 – Exemplo de distância Euclidiana entre duas amostras.



Fonte: Adaptado de (HAIR et al., 2009)

Equação 2.9 abaixo mostra como obter o valor normalizado do i-ésimo (x'_i) dado utilizando o método *Min-Max* (FÁVERO et al., 2009).

$$x'_i = \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{2.9}$$

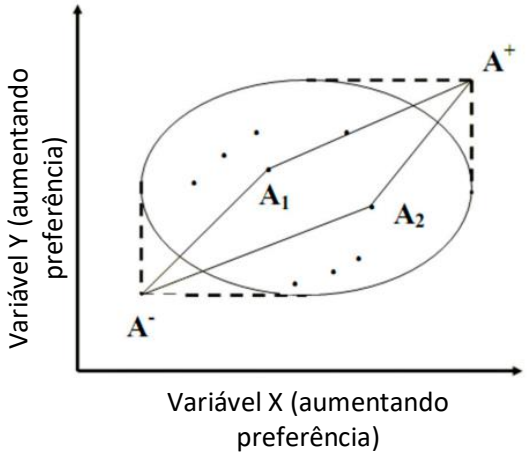
onde x_i é valor da amostra, $x_{\min}$ e $x_{\max}$ são valor mínimo e o valor máximo dos dados para cada variável, respectivamente.

2.5.3 A TOPSIS

A TOPSIS é utilizada para selecionar e ranquear alternativas por ordem de preferência, de acordo com critérios pré-estabelecidos (JATOTH et al., 2019). Ela define como melhor alternativa aquela que esteja mais próxima da *solução ideal positiva* e a mais distante da *solução ideal negativa* (LIMA; CARPINETTI, 2015). A solução ideal positiva é formada considerando os melhores valores alcançados pelas alternativas durante a avaliação em relação a cada critério de decisão, enquanto a solução ideal negativa é composta de forma similar, considerando os piores valores (KAHRAMAN; CEBECI; ULUKAN, 2003). A Figura 6 exibe um exemplo da aplicação do conceito da TOPSIS para duas variáveis.

A seguir são descritos os passos da TOPSIS de acordo com os autores Lima e Carpinetti (2015). Primeiro, é definida uma matriz de decisão D que corresponde às alternativas (ver Equação 2.10) e um vetor de pesos W (ver Equação 2.11). Na Equação 2.10, A_i representa as alternativas a serem avaliadas, C_j ilustra os critérios usados, i representa o número de alternativas (onde $i = 1, 2, \ldots, n$) e j o número dos critérios de decisão (onde $j = 1, 2, \ldots, m$). Na Equação 2.11, W é o conjunto de pesos, dado pelos pesos w_j de cada

Figura 6 – Conceito da TOPSIS, onde A^+ representa a solução ideal positiva e A^- representa a solução ideal negativa.



Fonte: Adaptado de Hwang e Yoon (1981).

critério C_j de forma que $\sum_{i=1}^{m} w_j = 1$.

$$\mathrm{D} = \begin{array}{c} \\ A_1 \\ \vdots \\ A_i \\ \vdots \\ A_n \end{array} \begin{array}{c} \begin{array}{cccccc} C_1 & C_2 & \ldots & C_j & \ldots & C_m \end{array} \\ \begin{bmatrix} d_{11} & d_{12} & \ldots & d_{1j} & \ldots & d_{1m} \\ \vdots & \vdots & & \vdots & & \vdots \\ d_{i1} & d_{i2} & \ldots & d_{ij} & \ldots & d_{im} \\ \vdots & \vdots & & \vdots & & \vdots \\ d_{n1} & d_{n2} & \ldots & d_{nj} & \ldots & d_{nm} \end{bmatrix} \end{array} \tag{2.10}$$

$$\tilde{\mathbf{W}} = \left[\tilde{\mathbf{w}}_1, \tilde{\mathbf{w}}_2, \ldots, \tilde{\mathbf{w}}_\mathrm{m}\right] \tag{2.11}$$

Posteriormente, a matriz de decisão D é normalizada e ponderada. Os elementos da matriz normalizada $N = [n_{ij}]$ são definidos de acordo com a Equação 2.12.

$$n_{ij} = w_j \times \frac{d_{ij} - d_j min}{d_j max - d_j min} \tag{2.12}$$

Após a normalização das alternativas, é determinada a solução ideal positiva (A^+) e a solução ideal negativa (A^-) como mostrado pelas Equações 2.13 e 2.14, respectivamente (ARESE et al., 2018).

$$A^+ = \left\{\left(\max n_{ij} \mid \mathrm{j} = 1, 2, \ldots, \mathrm{m}\right), \left(\min n_{ij} \mid \mathrm{j} = 1, 2, \ldots, \mathrm{m}\right)\right\} \tag{2.13}$$

$$A^- = \left\{\left(\max n_{ij} \mid \mathrm{j} = 1, 2, \ldots, \mathrm{m}\right), \left(\min n_{ij} \mid \mathrm{j} = 1, 2, \ldots, \mathrm{m}\right)\right\} \tag{2.14}$$

No passo seguinte, é calculado, para cada alternativa, a distância entre os valores normalizados e ponderados da matriz N (matriz D normalizada) e os valores da solução ideal positiva (ver Equação 2.15) e negativa (ver Equação 2.16). Nesta etapa, é utilizada uma distância de (dis)similaridade, como por exemplo, a Euclidiana.

$$D_i^+ = \sqrt{\sum_{i=1}^{n} \left(n_{ij} - n_j^+\right)^2} \tag{2.15}$$

$$D_i^- = \sqrt{\sum_{i=1}^{n} \left(n_{ij} - n_j^-\right)^2} \tag{2.16}$$

Após a obtenção das distâncias entre cada alternativa e a solução ideal positiva e negativa, é possível calcular o coeficiente de aproximação (CC_i) para cada alternativa. A Equação 2.17 é usada para calcular o CC_i. O valor resultante do CC_i corresponde ao desempenho global de cada alternativa i avaliada. O resultado do coeficiente é dado entre zero e um.

$$CC_i = \frac{D_i^-}{(D_i^+ + D_i^-)} \tag{2.17}$$

Por último, as alternativas devem ser classificadas em ordem decrescente observando os valores obtidos para o coeficiente de aproximação. As melhores alternativas são aquelas cujo desempenho global fica mais próximo de um (1).

O Algoritmo 1 apresenta o pseudocódigo da TOPSIS implementado nesta tese para selecionar e ranquear os cenários das estratégias de DR utilizando funções multicritério. Os dados de entrada para o algoritmo fornecidos pela(o) profissional que conduz a avaliação são: (1) o número de cenários `n` a serem avaliados; (2) a matriz de entrada; (3) a matriz de pesos; e (4) a função multicritério. O número de cenários é definido pela quantidade de diferentes configurações ou estratégias de DR consideradas na avaliação. Isso contempla o uso de diferentes estratégias de DR para um ambiente ou o uso da mesma estratégia empregando configurações distintas. Por exemplo, o cenário 1 pode ser avaliar o uso do *backup* como estratégia de DR, enquanto o cenário 2 pode ser avaliar a adoção da replicação de BD para o mesmo ambiente. A matriz de entrada (`m_e`) contêm os resultados dos critérios (métricas, como por exemplo, disponibilidade) de cada cenário avaliado. A matriz de pesos (`ps`) é opcional. Quando definida, a soma dos pesos de todos os critérios deve ser igual a 1. Por fim, a função multicritério `fm` define quais critérios serão maximizados, minimizados ou omitidos (não considerado) no processo de ranqueamento.

Na `linha 2`, é executada a normalização da matriz de entrada, através da função `normalizarMatriz`. Essa função recebe como parâmetros, a matriz de resultados dos cenários `m_e` e a matriz de pesos `ps`, ambas definidas pela(o) profissional que executa a avaliação. A matriz de entrada segue um modelo mostrado na Tabela 2. Já a matriz de pesos segue um modelo mais simples, onde para cada critério presente na matriz de entrada,

Algoritmo 1: Algoritmo TOPSIS para ranquear estratégias de DR.

Entrada: Número de cenários a serem avaliados n
Matriz de entrada contendo resultados de cada cenário m_e = $[[x_1, y_1, ..., z_1], ..., [x_n, y_n, ..., z_n]]$
Pesos atribuídos a cada critério (métrica) ps = $[[cr_1, p_1], ..., [cr_n, p_n]]$
Função multicritério fm = $[[cr_1, f_1], ..., [cr_n, f_n]]$

```
1  Início
2      m_e ⟵ normalizarMatriz(m_e, ps);
3      sip ⟵ encontrarSolucaoIdealPositiva(m_e, fm);
4      sin ⟵ encontrarSolucaoIdealNegativa(m_e, fm);
5      Para i de 1 até n faça
6          distPositiva ⟵ calcularDistanciaEuclidiana(m_e.get(i), sip);
7          distNegativa ⟵ calcularDistanciaEuclidiana(m_e.get(i), sin);
8          m_r[i][1] ⟵ i;
9          m_r[i][2] ⟵ calcularCCi(distPositiva, distNegativa);
10     Fim
11     m_r ⟵ ordenarPorCci(m_r);
12     retorna m_r;
13 Fim
```

Resultado: Matriz ranqueada dos cenários m_r = $[[c_m, cci_m], ..., [c_n, cci_n]]$

é descrito o critério (ex.: disponibilidade) e o peso associado a esse critério (ex.: 0.5). Após a normalização da matriz de entrada são definidas a solução ideal positiva (`sip`, na `linha 3`) e a solução ideal negativa (`sin`, na `linha 4`). A função multicritério é definida de forma similar aos pesos, onde, para cada critério presente na matriz de entrada, é descrito o critério e o objetivo (ex.: (disponibilidade, maximizar) e (custo, minimizar)). A solução ideal positiva considera os melhores valores possíveis para cada critério dentre todos os cenários, de acordo com a função multicritério definida. Por outro lado, a solução ideal negativa considera os piores valores possíveis para cada critério dentre todos os cenários, de acordo com a função multicritério definida. Por exemplo, considere dois cenários onde a disponibilidade de cada um equivale a 0,9999 e 0,9955, respectivamente. Caso apenas esses dois cenários estejam sendo avaliados e parte da função multicritério fosse *maximizar a disponibilidade*, o valor 0,9999 faria parte da solução ideal positiva, enquanto o valor 0,9955 faria parte da solução ideal negativa.

Tabela 2 – Modelo para matriz de entrada dos resultados de cada cenário a serem avaliados.

	Critério 1	Critério 2	⋯	Critério m
Cenário 1	x_1	y_1	⋯	z_1
Cenário 2	x_2	y_2	⋯	z_2
⋮	⋮	⋮	⋮	⋮
Cenário n	x_n	y_n	⋯	z_n

Da `linha 5` à `linha 10` é executado um laço de repetição que vai de `1` até o número de cenários avaliados `n`. Dentro desse laço de repetição, é calculada a distância euclidiana entre cada cenário (indexado pela variável `i`), a solução ideal positiva (`linha 6`) e a solução ideal negativa (`linha 7`). Na `linha 8`, é salvo na matriz de resultado `m_r` o identificador

de cenário avaliado (aqui, definido pela variável i) e, em seguida, na `linha 9` é calculado o CC_i do cenário `i`. Por fim, na *linha 11* a matriz de resultados `m_r` é ordenada de forma decrescente de acordo com o valor do CC_i de cada cenário. Dessa forma, são ranqueados os cenários que possuem CC_i mais próximos de 1, representando os cenários que mais se adéquam à função multicritério definida.

2.6 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os principais conceitos acerca dos temas deste trabalho. Primeiramente, os temas de virtualização e computação em nuvem foram apresentados e discutidos, uma vez que esta tese faz uso de ambientes virtualizados e adota estratégias de DR para tais tipos de ambientes. Posteriormente, foram explanados conceitos sobre dependabilidade, disponibilidade, e modelagem de sistemas, uma vez que neste trabalho são propostos modelos de redes de Petri para avaliação das estratégias de DR. Em seguida, foram abordados fundamentos acerca da recuperação de desastres, estratégias de DR, e a avaliação dessas estratégias. Por fim, como um dos objetivos da tese é identificar as melhores estratégias de DR considerando diferentes requisitos das empresas (ver Seção 1.4), também foram abordados conceitos relacionados aos métodos de decisão multicritério.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma visão geral dos trabalhos encontrados na literatura que possuem relação com esta tese. A análise de infraestruturas computacionais, a adoção de estratégias de DR, e o uso de sistemas de TIC tolerantes a falhas são temas que têm chamado atenção de vários pesquisadores. Em (MENDONçA et al., 2019a), apresentamos um estudo de mapeamento sistemático onde são detalhados diversas características de trabalhos desenvolvidos no âmbito de DR para infraestruturas computacionais e sistemas de TIC (ver um resumo no Apêndice A).

Sendo assim, neste capítulo são apresentados trabalhos presentes no mapeamento sistemático (MENDONçA et al., 2019a), bem como outros trabalhos que não foram contemplados nele. Os trabalhos relacionados são organizados em três seções: Avaliação e modelagem de infraestruturas computacionais; Planejamento e avaliação de estratégias de recuperação de desastres; e Avaliação e seleção de infraestruturas computacionais apoiadas por métodos de decisão. Ao final deste capítulo, é apresentada uma comparação dos trabalhos mais relevantes encontrados na literatura, destacando a relevância da pesquisa realizada nesta tese.

3.1 AVALIAÇÃO E MODELAGEM DE INFRAESTRUTURAS COMPUTACIONAIS

O uso de diferentes técnicas de modelagem para avaliação de infraestruturas computacionais é bastante estabelecido na literatura. Modelos formais como redes de Petri (e suas extensões como, por exemplo, DSPNs), CTMCs, e RBDs são exemplos de algumas técnicas de modelagem utilizadas para avaliar métricas relacionadas à dependabilidade de infraestruturas computacionais. A seguir, são apresentados os principais trabalhos que abordam a análise e a avaliação de infraestruturas computacionais através de modelos formais e que possuem relação com esta tese.

Andrade (2014) propôs um *framework* que permite mapear diagramas da *System Modelling Language* (SysML) (OMG, 2012) anotados com o *Profile Profile for Modeling and Analysis of Real-time and Embedded systems* (MARTE) (OMG., 2011) em DSPNs. O trabalho focou na modelagem e análise de infraestruturas computacionais, observando métricas relacionadas à disponibilidade, a fim de maximizar o uso dos recursos computacionais com o menor custo atrelado possível. Dessa forma, o estudo também fornece meios para se estudar os mecanismos de tratamento de interrupções, bem como as infraestruturas computacionais de forma simples e sem a necessidade de conhecimento especializado em modelagem estocástica. Ademais, o autor também apresentou o OpenMADS, uma ferramenta que implementa o *framework* proposto. A ferramenta possui integração com as ferramentas ASTRO/Mercury (SILVA et al., 2015) e TimeNET (ZIMMERMANN; KNOKE, 2007), onde é

possível realizar análises estocásticas dos modelos DSPNs obtidos automaticamente pelo OpenMADS.

Uma estratégia baseada em um modelagem hierárquica e heterogênea para o planejamento de infraestruturas de nuvem foi proposta por Sousa et al. (2014a) e Sousa et al. (2014b). Os autores fizeram uso de modelos RBDs e SPNs. A estratégia permite a seleção de infraestruturas de nuvem de acordo com as exigências de dependabilidade, desempenho e custo. Estudos de caso baseado em ambientes virtuais de aprendizagem hospedado na plataforma de nuvem *Eucalyptus* foram usados para demonstrar a viabilidade da estratégia e dos modelos propostos.

Ghosh et al. (2014) realizaram análises de disponibilidade em uma infraestrutura de nuvem utilizando CTMC. Eles utilizaram máquinas físicas agrupadas em três tipos de conjuntos de provisionamento de recursos: *hot* (em execução), *warm* (ligado, mas não operante) e *cold* (desligado). Essa divisão foi realizada com base no consumo de energia e nas características de aguardo no provisionamento de recursos. A modelagem estocástica foi utilizada para análise de dependabilidade desses sistemas na nuvem de provisionamento de IaaS nos três modos acima citados.

Nguyen, Kim e Park (2016) apresentaram modelos de disponibilidade para infraestrutura de DCs usando *Stochastic Reward Nets* (SRNs), uma extensão das redes de Petri. Foram analisados diferentes configurações de DCs de alta disponibilidade geograficamente distribuídos. No estudo, os autores avaliaram métricas de disponibilidade, indisponibilidade e o custo da indisponibilidade. Também foi realizada uma análise de sensibilidade para identificar os parâmetros que mais impactam nos resultados das métricas estudadas. Os modelos criados levaram em consideração diferentes falhas, desastres, dependência entre subsistemas e a transmissão de VMs entre os DCs. Ao final, os autores concluíram que falhas inesperadas durante o processo de transmissão de VMs, a velocidade da rede que conecta os DCs e o tamanho das VMs impactam diretamente na disponibilidade do sistema.

Li et al. (2017) propuseram uma metodologia para modelagem e análise de confiabilidade de serviços hospedados em infraestruturas computacionais. Os autores utilizaram teoria das filas e teoria dos grafos para formular um modelo capaz de avaliar a confiabilidade de serviços de TIC hospedados em DCs geograficamente distribuídos. A abordagem e modelos foram aplicadas na avaliação de DCs que atuam sob o modelo de redundância *active-active*. Além de avaliar a confiabilidade de serviços usando os modelos, a abordagem proposta contempla a realização de análise de sensibilidade para identificar os parâmetros que mais impactam na confiabilidade dos serviços ofertados nos DCs. O desempenho dos componentes do DC e o fluxo de rede foram indicados como os principais parâmetros que impactam na confiabilidade do serviço.

Dantas (2018) concentrou-se na avaliação de disponibilidade, disponibilidade orientada a capacidade, desempenho e custo para infraestruturas computacionais que oferecem serviço de vídeo *streaming*. O trabalho utilizou modelos de RBDs, SPN, e CTMCs para

auxiliar no planejamento dessas infraestruturas. Além disso, o trabalho abordou o uso da meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP) para geração e avaliação de diferentes cenários de infraestruturas de nuvens privadas. De forma semelhante, os trabalhos de Bezerra et al. (2014) e Melo et al. (2014) também abordaram a avaliação de infraestrutura computacionais para oferta de serviços de vídeo *streaming*.

Anh Nguyen et al. (2019) propuseram modelos SRNs para examinar o impacto de falhas em uma infraestrutura *Software-Defined Network* (SDN). O modelo focou na análise da disponibilidade estacionária do ambiente, considerando diferentes falhas de hardware (em *hosts*, *switches*, unidades de armazenamento e links) e falhas de software em VMs. A migração de VMs também foi implementada no modelo para auxiliar no aumento da disponibilidade da infraestrutura. Além disso, os autores realizaram análises de sensibilidade para estudar o comportamento do sistema em relação aos diferentes fatores que impactam a disponibilidade da infraestrutura. Por fim, os resultados dos modelos foram verificados através do pacote *SimSDN*, pertencente ao simulador CloudSim Plus (FILHO et al., 2017).

3.2 PLANEJAMENTO E AVALIAÇÃO DE ESTRATÉGIAS DE RECUPERAÇÃO DE DESASTRES

Grande parte dos trabalhos desenvolvidos com o tema de recuperação de desastres foca na análise de uma estratégia específica, desconsiderando a variedade de estratégia e diferentes requisitos das empresas. No entanto, de uma forma geral, a avaliação de estratégias de DR têm sido abordada em vários trabalhos de pesquisas, seja através de modelagem, experimentação ou simulação. Assim, esta seção destaca trabalhos que estão inseridos no contexto da avaliação de estratégias de DR.

Rooney, McBride e Hanif (2008) apresentaram o *IBM TotalStorage Productivity Center for Replication* (TPC-R), que é uma ferramenta para auxiliar clientes tanto na implementação de estratégias de replicações de dados para DR, quanto no aumento da disponibilidade. A ferramenta foi desenvolvida para gerenciar os serviços de cópia de dados em controladores da própria IBM e em *SAN Volume Controllers* (SVCs). O TPC-R ajuda os clientes a iniciar, parar e suspender o processo de replicação de dados. Além de auxiliar no gerenciamento da cópia dos dados, a ferramenta também ajuda no planejamento de cenários para replicação de dados com até 3 níveis de redundância, apresentando estimativa de métricas de cada cenário analisado. Ela foi projetada para suportar replicação em um, dois, ou três DCs. O TPC-R considera as métricas de RPO, RTO, disponibilidade, capacidade de resposta e custo durante suas análises.

Gopisetty et al. (2008) apresentaram o *IBM Provisioning Planner*, uma ferramenta voltada ao planejamento e provisionamento de armazenamento. A ferramenta visa identificar locais apropriados para a alocação de novos volumes de dados, levando em consideração requisitos de capacidade, carga de trabalho e resiliência. Ela também possui um módulo para planejamento de estratégias de DR. Nesse módulo, o *software* analisa diferentes confi-

gurações (em base de conhecimento limitada) para as camadas de aplicação, VMs, BDs e armazenamento buscando combinar e organizar as diferentes tecnologias de replicação que podem ser utilizadas. Depois de implementar a estratégia escolhida, a ferramenta ajuda a gerenciar a replicação dos dados entre as diferentes camadas (aplicação, VMs, BDs e armazenamento).

Alhazmi e Malaiya (2012) discutiram como avaliar o custo em estratégias de DR. Os autores propõem equações para analisar os custos das estratégias de *backup*, considerando diferentes configurações para o plano de recuperação de desastres. Apesar de o trabalho ser apenas um guia para futuras análises utilizando as equações, os autores listam alternativas como *backup* local, *backup* remoto, e ambiente de *backup* para computação em nuvem para serem avaliadas quanto ao RPO, RTO e custo.

Lenk e Pallas (2013) apresentaram um modelo baseado em cadeias de Markov para analisar a disponibilidade e custos de sistemas hospedados em ambientes de nuvem. Através de análises quantitativas do modelo desenvolvido, os autores demonstram que a utilização de uma infraestrutura de nuvem secundária em modo *warm-standby*, pode aumentar a disponibilidade do sistema. Em um segundo trabalho (LENK; TAI, 2014), os autores implementaram a abordagem proposta, a qual consiste em uma replicação de ambiente em nuvens utilizando o modo *warm-standby*. Nesse trabalho, também foram avaliadas as métricas de RTO e RPO.

Xia et al. (2014) avaliaram um sistema de serviço de arquivos que realizava *backups* periódicos de dados para investigar métricas relacionadas à disponibilidade do sistema, perda de dados e rejeição de solicitações de usuários. Para obter as métricas, os autores utilizaram modelos de CTMC, redes de filas e SRN que representam detalhes operacionais do sistema. Assim, utilizando os modelos criados, foram avaliadas as métricas considerando diferentes técnicas de *backup*/restauração, políticas e cenários de carga de trabalho. Os resultados alcançados pelos modelos possibilitaram comparar os efeitos de diferentes técnicas e políticas de *backup*/restauração e o impacto no desempenho e disponibilidade do sistema analisado.

Silva (2016) desenvolveu uma ferramenta para avaliar métricas de desempenho, dependabilidade e capacidade de sobrevivência em DCs geograficamente distribuídos que oferecem IaaS. A ferramenta proposta, chamada de *GeoCloud Modcs*, tem como principal objetivo o planejamento da infraestrutura dos DCs, apresentando o impacto para a migração de VMs entre os DCs primários e secundários. O autor faz uso de modelos de SPNs e RBDs para realizar a avaliação de desempenho. Além disso, a ferramenta permite analisar o impacto da ocorrência de desastres nas métricas avaliadas.

Andrade e Nogueira (2018) apresentaram modelos SPNs para análise de estratégias de DR em ambientes de *Internet of Things* (IoT). Os modelos propostos focavam na avaliação de métricas relacionadas à DR, como disponibilidade, RTO, e custo. Os autores também apresentam uma análise de sensibilidade onde avaliam os efeitos dos parâmetros

na disponibilidade do sistema através dos modelos propostos. Os autores concluem que a adoção de uma estratégia de DR para a infraestrutura estudada poderia melhorar a disponibilidade do sistema, aumentando em mais de 4 horas o tempo de disponibilidade do sistema anualmente. Em um segundo trabalho, Andrade e Nogueira (2019) propuseram modelos SPNs para avaliar estratégias de DR baseadas em nuvem para ambientes computacionais. Os modelos propostos eram capazes de avaliar métricas de disponibilidade e desempenho (como, por exemplo, tempo de resposta e vazão do sistema). Andrade e Nogueira (2019) ainda apresentaram um estudo de caso e comparam os resultados obtidos através dos modelos propostos com os resultados alcançados através do simulador de nuvem CloudSim Plus (FILHO et al., 2017).

3.3 AVALIAÇÃO E SELEÇÃO DE INFRAESTRUTURAS COMPUTACIONAIS APOIADA POR MÉTODOS DE DECISÃO

Métodos de tomada de decisão tem sido utilizados para auxiliar na seleção ou ranqueamento de serviços, infraestruturas, ou configurações de sistemas que atendam às necessidades específicas dos usuários. A adoção desses métodos visa auxiliar na tomada de decisão quando a quantidade de opções é grande ou simplesmente quando existem requisitos conflitantes entre as opções (ex.: custo e disponibilidade). Nesse contexto, diversos métodos têm sido adotados, especialmente para auxiliar no planejamento de infraestruturas/serviços para o paradigma de computação em nuvem ou a escolha de provedores que fornecem serviços em nuvens computacionais. No entanto, são menos comumente adotados no contexto de DR. Nesta seção, apresentamos alguns trabalhos que foram desenvolvidos utilizando diferentes métodos de decisão para auxílio ao planejamento de infraestruturas/serviços computacionais.

Jaiswal, Sen e Verma (2011) propuseram um *framework* para otimização do custo em planos de DR. Os autores também propõem o uso de algoritmos para chegar a uma solução que atenda aos requisitos do plano de DR com um custo mínimo. O trabalho foca na projeção de planos de DR baseados em estratégias de replicação de dados. Através de modelos abstratos para a estratégia de replicação de dados, são definidas algumas características, tais como RPO, RTO, distância e impacto da aplicação (latência causada pela estratégia de replicação adotada). Assim, os algoritmos desenvolvidos analisam as estratégias disponíveis considerando mais de um objetivo para minimizar, entretanto, priorizando o objetivo do custo.

Liu, Chan e Ran (2016) apresentaram uma abordagem multicritério para auxiliar empresas no processo de escolha de um fornecedor de serviços de nuvem considerando diversos fatores. O trabalho proposto utilizou de forma integrada a variância estatística, a TOPSIS, o *simple additive weighting* e o método Delphi-AHP para encontrar as melhores soluções. Os autores consideraram dados hipotéticos de atributos como os custos associados

à aplicação, e custos para computar, armazenar e transferir os dados para demonstrar a aplicabilidade da abordagem proposta.

Yang et al. (2017) propuseram um *framework* para avaliação e seleção de uma infraestrutura secundária de DR voltada para sistemas de *Big Data* acadêmico. O *framework* proposto é suportado por diferentes métodos de tomada de decisão como o *Decision-Making Trial and Evaluation Laboratory* (DEMATEL) (FONTELA; GABUS, 1976), *DEMATEL-based Network Process* (DNP) (SAATY, 1990) e *VIšekriterijumsko KOmpromisno Rangiranje* (VIKOR) (OPRICOVIC, 1998). Foram quantificados através de opiniões de especialistas diversos fatores e características de estratégias de DR como impacto da localização, disponibilidade, RTO, RPO, e operações de gerenciamento para servir como entrada para os métodos de decisão e otimização. Por fim, os autores apresentaram um estudo de caso baseado na avaliação e seleção de uma infraestrutura de DR para um DC de alto desempenho voltado para aplicativos de *Big Data*.

Yoo e Kim (2018) realizaram uma análise comparativa entre demandantes e provedores para adoção da computação em nuvem nas empresas. Os autores utilizaram o AHP e a análise Delphi para identificar os principais fatores que são levados em consideração para adoção da computação em nuvem por empresas, considerando três principais áreas de decisão corporativa (tecnologia, organização e ambiente). O estudo sugere importantes fatores como suporte da alta gerência, pressão competitiva e compatibilidade para a adoção de um sistema de computação em nuvem.

Kumar, Mishra e Kumar (2018) apresentaram um *framework* para determinar o provedor de nuvem mais adequado para um serviço, através dos métodos AHP e TOPSIS. Os autores utilizaram o AHP para definir os pesos das arquiteturas a serem consideradas no processo de seleção e, depois disso, utilizaram a TOPSIS para obter a classificação final do serviço de nuvem baseado no desempenho geral. A abordagem leva em consideração diferentes fatores (ex.: disponibilidade, tempo de resposta, vazão, escalabilidade, custo, entre outros) para analisar as opções.

Ferreira et al. (2019) modelaram um problema de alocação de recursos para infraestruturas computacionais. O objetivo principal era maximizar os recursos de disponibilidade, atendendo requisitos mínimos em termos de CPU, memória, rede e armazenamento e custo. Os autores utilizaram modelos RBDs para avaliar a disponibilidade dos DCs e aplicaram algoritmos de otimização multi-objetivo *Non-dominated Sorting Genetic Algorithm* (NSGA-II) e *Generalized Differential Evolution* (GDE3) para buscar uma alocação de recursos eficiente. Os resultados mostraram que a adoção dos algoritmos multi-objetivos pode ser eficiente para buscar uma solução que atenda diferentes requisitos. Por fim, o algoritmo GDE3 se sobressaiu em alguns cenários por apresentar um número maior de soluções possíveis.

Araujo (2019) desenvolveu uma abordagem baseada em modelos de dependabilidade, performabilidade, custos e métodos de decisão (ex.: TOPSIS, *Min-Max*) para auxiliar

usuários na escolha de infraestruturas de nuvem públicas ou privadas. O trabalho utiliza SPNs e RBD para representar e avaliar dependabilidade e performabilidade para diferentes infraestruturas em nuvem. Também foi desenvolvida a *MiPACE*, que é uma ferramenta que faz uso de métodos de tomada de decisão multicritério para auxiliar no processo de escolha da infraestrutura a hospedar o serviço desejado.

Similarmente, Jatoth et al. (2019) propuseram um modelo de tomada de decisão multicritério para a seleção de serviços em nuvem. A metodologia proposta atribuiu classificações a parâmetros (custo, processamento, I/O, armazenamento e memória) de nuvens com base no AHP. Então, utilizando o Grey TOPSIS, os autores analisaram diferentes configurações de serviços de nuvem para obter os mais adequados aos requisitos das empresas. Já Regunathan, Murugaiyan e Lavanya (2019) utilizaram o TOPSIS para analisar diferentes provedores de *Mobile Cloud Computing*s (MCCs) baseados nos parâmetros de tempo de resposta, latência, disponibilidade, *throughput* e confiabilidade de diferentes provedores.

3.4 COMPARAÇÃO ENTRE OS TRABALHOS RELACIONADOS MAIS RELEVANTES

Esta seção apresenta e compara os trabalhos mais relevantes discutidos nas seções anteriores com a pesquisa desenvolvida nesta tese. De diferentes formas, todos os trabalhos apresentados anteriormente abordaram o estudo de infraestruturas computacionais ou estratégias de DR, propondo métodos, modelos, abordagens ou *softwares* para auxiliar no planejamento, implantação ou avaliação de estratégias de DR.

Dentre os trabalhos relacionados apresentados, os trabalhos de Gopisetty et al. (2008) e Rooney, McBride e Hanif (2008) certamente são os que possuem uma completude maior dentro de seus propósitos. Porém, por se tratar de produtos comerciais, o acesso a seus detalhes é limitado. Andrade (2014), Silva (2016) e Nguyen, Kim e Park (2016) desenvolveram estudos utilizando conceitos semelhantes aos utilizados neste trabalho. No entanto, Silva (2016) e Nguyen, Kim e Park (2016) analisam mais profundamente questões relacionadas à transmissão de VMs entre DCs geograficamente distribuídos, focando na aplicação de uma única estratégia de DR. Já Andrade (2014) tem como principal contribuição o desenvolvimento de uma estratégia para o mapeamento de modelos semiformais em SPNs. Outros trabalhos (XIA et al., 2014; LENK; PALLAS, 2013; LENK; TAI, 2014; ANDRADE; NOGUEIRA, 2018; ANDRADE; NOGUEIRA, 2019) também apresentam limitações por analisar estratégias de DR de formas isoladas. Ou seja, não consideram estratégias mais abrangentes.

Os trabalhos desenvolvidos por Araujo (2019) e Jatoth et al. (2019) apresentam importantes aplicações de métodos de tomada de decisão multicritério para auxiliar na seleção de infraestruturas de computação em nuvem. O estudo desenvolvido por Araujo (2019) ainda contempla o uso de modelos analíticos, probabilísticos e métodos de tomada de decisão. No entanto, nenhum deles foca em estratégias de DR. Já trabalho realizado

por Yang et al. (2017), apesar de ter foco em DR e utilizar métodos de tomada de decisão, analisa apenas um tipo de estratégia.

A Tabela 3 exibe e compara os trabalhos relacionados considerados mais relevantes, em função de algumas características (técnicas de avaliação, foco em DR, adoção de diferentes estratégias de DR, e uso de métodos de tomada de decisão) por eles adotados. Na Tabela 3, a coluna **Técnicas de Avaliação** lista técnica de avaliação utilizada pelos trabalhos. A coluna **Métricas avaliadas** indica se o trabalho correspondente realizou a avaliação de métricas quantitativas em estratégias de DR e quais são essas métricas. A coluna **Adoção de diferentes estratégias de DR** aponta se o trabalho considerou a análise ou avaliação de mais de uma estratégia de DR. Por fim, a coluna **Uso de métodos de tomada de decisão** especifica se o estudo faz uso de algum método de tomada de decisão para seleção ou ranqueamento de soluções. Ao final da tabela, também são adicionadas as características referentes a esta tese, a fim de facilitar a comparação com os trabalhos relacionados selecionados.

Tabela 3 – Comparação entre os trabalhos relacionados mais relevantes.

Trabalhos	**Técnicas de Avaliação**	**Métricas avaliadas**	**Adoção de diferentes estratégias de DR**	**Uso de métodos de tomada de decisão**
Rooney, McBride e Hanif (2008)	-	-	Não	Não
Gopisetty et al. (2008)	-	Latência, RTO, RPO e custo	Sim	Não
Jaiswal, Sen e Verma (2011)	-	Custo	Não	Sim
Lenk e Pallas (2013)	CTMCs	Disp. e custo	Não	Não
Andrade (2014)	Experimentos e SPNs	Disp., indisp. e custo	Não	Não
Xia et al. (2014)	CTMCs, SRNs	Disp., perda de dados e rejeição de dados	Não	Não
Lenk e Tai (2014)	Experimentos	RTO e custo	Não	Não
Silva (2016)	Experimentos, RBDs e SPNs	Disp., indisp. e desempenho	Não	Não
Nguyen, Kim e Park (2016)	SRNs	Disp., indisp. e custo	Não	Não
Yang et al. (2017)	-	-	Não	Sim
Andrade e Nogueira (2018)	SPNs	Disp., indisp, RTO, RPO e custo	Não	Não
Andrade e Nogueira (2019)	Simulação e SPNs	Disp., indisp, desempenho, RTO, RPO e custo	Não	Não
Esta Tese	**Experimentos e DSPNs**	**Disp., indisp., RTO, RPO e custo**	**Sim**	**Sim**

3.5 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os principais trabalhos relacionados ao estudo proposto. Foram mostrados vários trabalhos que proporcionam a análise de infraestruturas computacionais,

planejamento e avaliação de estratégias de DR. Mostramos que o uso de modelos formais para facilitar a análise de infraestruturas computacionais e estratégias de DR já é bastante estabelecido na literatura. A adoção desses modelos é um forte aliado para a economia de tempo e recursos, visto que exigem menos esforço que a realização de experimentos e apresentam resultados com um bom nível de confiança. Além disso, embora existam trabalhos que apresentem ferramentas, *frameworks* ou metodologias para auxiliar no planejamento e avaliação de estratégias de DR, a maior parte deles limita-se ao estudo de apenas uma estratégia isoladamente. Por fim, também foi apresentado que métodos de tomada de decisão são utilizados com frequência para auxiliar no processo de escolha de infraestruturas computacionais, mas menos frequentemente usados no contexto de DR. Dessa forma, esta tese torna-se relevante por reunir as características desses trabalhos, visto que propomos avaliar diferentes estratégias de DR quantitativamente através de modelos formais e aplicar métodos de decisão multicritério para auxiliar na seleção e ranqueamento das estratégias de DR.

4 METODOLOGIA PARA AVALIAÇÃO E SELEÇÃO DE ESTRATÉGIAS DE RECUPERAÇÃO DE DESASTRES

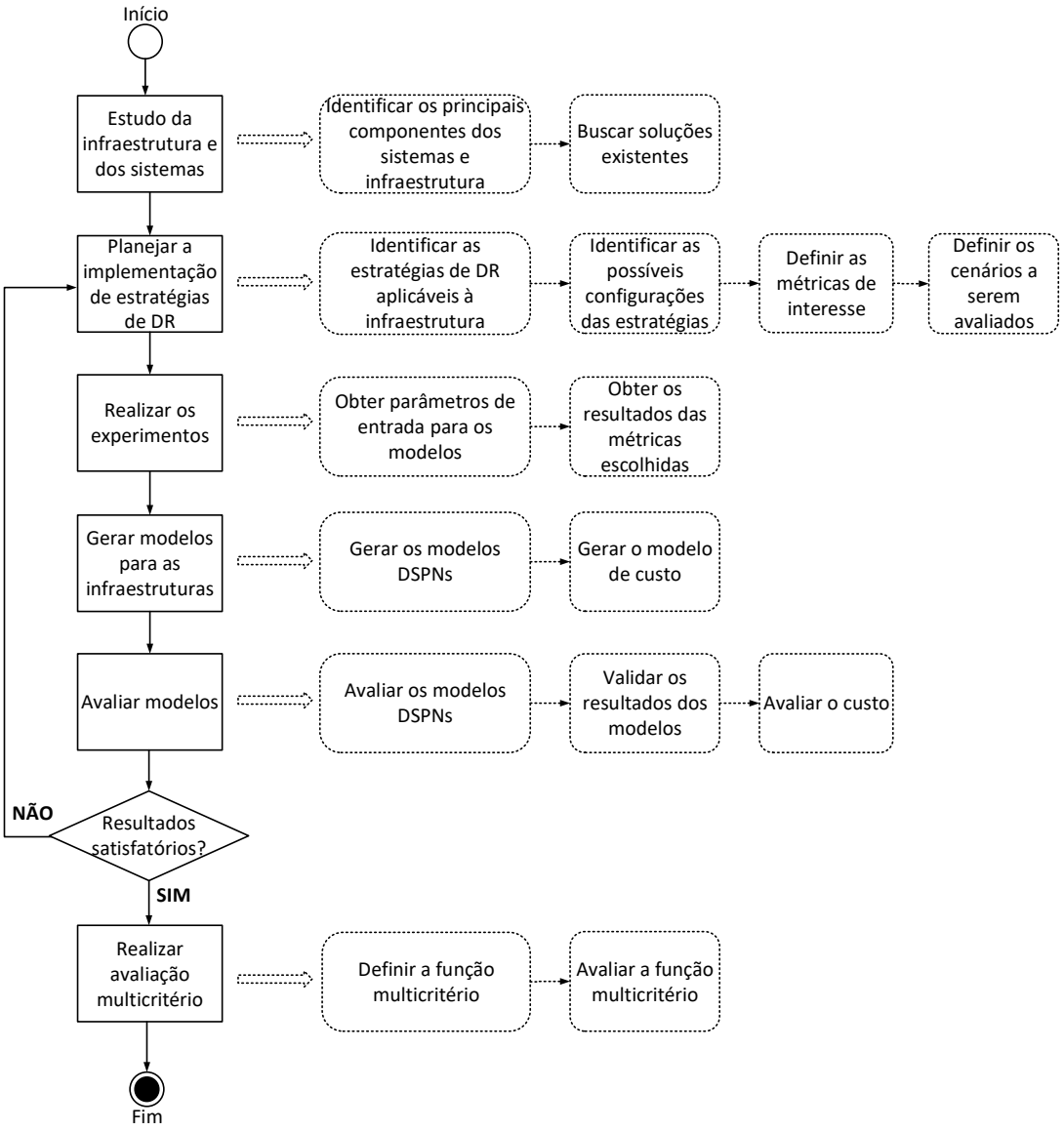
Este capítulo descreve a metodologia proposta para auxiliar na avaliação e seleção de estratégias de DR. A metodologia guia o uso integrado de modelos probabilísticos, modelos de custo e métodos de tomada de decisão multicritério para auxiliar na avaliação de tais estratégias. Primeiro, é apresentada uma visão geral da metodologia e, posteriormente, são detalhadas suas as atividades e etapas.

4.1 VISÃO GERAL

O processo de avaliação e seleção das estratégias de DR é realizado através da execução de várias atividades essenciais para se obter os resultados de forma consistente. Para alcançar esse objetivo, esta tese propõe o uso integrado de modelos DSPNs e de custos, além de métodos de tomada de decisão multicritério. O uso de modelos DSPNs é proposto para realizar a avaliação de infraestruturas computacionais e das estratégias de DR acerca de importantes métricas de DR como: disponibilidade, indisponibilidade, RTO e RPO. Além disso, são utilizados um conjunto de modelos matemáticos para estimar os custos das infraestruturas e estratégias de DR. Já para prover a seleção das estratégias de DR, esta tese faz a utilização da TOPSIS, um método de tomada de decisão multicritério. O uso da TOPSIS torna possível a identificação e seleção das estratégias de DR que mais se adequam aos critérios estabelecidos por empresas ou projetistas. Dessa forma, para apoiar todo o processo de avaliação e seleção das estratégias, é proposta uma metodologia.

O fluxograma mostrado na Figura 7 representa de maneira visual a metodologia proposta. Os retângulos representam cada atividade a ser executada obedecendo a ordem de execução apontada pelas setas. Os blocos tracejados ao lado de cada atividade representam as possíveis etapas a serem executadas em cada atividade da metodologia, também obedecendo a ordem de execução apontada pelas setas tracejadas. Somente quando as etapas de uma atividade são finalizadas, a(o) profissional que conduz o processo de avaliação segue para a próxima atividade. O losango é um símbolo usado para separar e reconectar o fluxo de execução das atividades. Ele pode alterar o fluxo de execução das atividades dependendo do resultado obtido pela condição nele determinada. Dessa forma, a metodologia proposta consiste em 6 atividades principais: *Estudo da infraestrutura e dos sistemas*, *Planejar a implementação de estratégias de DR*, *Realizar experimentos*, *Gerar modelos para as infraestruturas*, *Avaliar modelos* e *Realizar a avaliação multicritério*. Conforme apresentado a seguir, cada uma dessas atividades são subdivididas em etapas, visando proporcionar um melhor detalhamento.

Figura 7 – Metodologia proposta para guiar a avaliação e seleção de estratégias de DR.



Fonte: Criado pelo autor.

4.2 ESTUDO DA INFRAESTRUTURA E DOS SISTEMAS

Para realizar a avaliação das infraestruturas computacionais que adotam ou visam adotar estratégias de DR, é fundamental entender como a infraestrutura e seus sistemas funcionam ou irão funcionar. Dessa forma, essa atividade pode ser subdividida em duas etapas: (1) identificar os principais componentes dos sistemas e infraestrutura e (2) buscar soluções existentes. A seguir, são descritas cada uma dessas etapas.

Identificar os principais componentes dos sistemas e infraestrutura: a compreensão da infraestrutura e de como os sistemas computacionais presentes nela se relacionam é o primeiro passo para conduzir corretamente o processo de avaliação. Essa etapa requer grande atenção e cuidados especiais por parte da(o) profissional que conduz a avaliação, a fim de evitar erros de interpretação que comprometam as demais etapas da metodologia.

Assim, o principal objetivo desta etapa é identificar os principais componentes (ex.: servidores, VMs e conexões de rede) que as infraestruturas e os sistemas adotam, bem como suas dependências. Essa identificação visa definir e delimitar corretamente a configuração que represente o ambiente adotado para adoção de uma estratégia de DR.

Buscar soluções existentes: nessa etapa, são consultados os materiais técnicos acerca de infraestruturas computacionais, estratégias de DR e modelagem de sistemas por meio de livros, artigos científicos, sites, fóruns, entre outros. A execução desta etapa possibilita o conhecimento de configurações, estratégias de DR, métodos e técnicas que poderão ser adotadas, adaptadas ou criadas para avaliar as infraestruturas estudadas na etapa anterior.

4.3 PLANEJAR A IMPLEMENTAÇÃO DE ESTRATÉGIAS DE DR

As estratégias de DR são implementadas em infraestruturas com o principal objetivo de mitigar as interrupções dos sistemas de TIC e a perda de dados (MENDONçA et al., 2019a). No entanto, algumas infraestruturas ou sistemas podem apresentar limitações para implementar algumas estratégias de DR. Dessa forma, a partir do entendimento da infraestrutura e dos sistemas realizados na atividade anterior, é preciso planejar quais estratégias podem ser implementadas no ambiente definido. Assim, esta atividade pode ser subdividida em: (1) identificar as estratégias de DR aplicáveis à infraestrutura; (2) identificar as possíveis configurações das estratégias; (3) definir as métricas de interesse; e (4) definir os cenários a serem avaliados.

Identificar as estratégias de DR aplicáveis à infraestrutura: nesta etapa, são identificadas quais estratégias podem ser aplicadas na infraestrutura computacional a ser avaliada. Essa identificação é possível devido ao conhecimento adquirido a partir do material técnico consultado na atividade anterior. É preciso atenção nessa etapa, pois identificações errôneas podem afetar diretamente a geração de modelos e o processo de avaliação da(s) estratégia(s) de DR como um todo.

Identificar as possíveis configurações das estratégias: após a identificação das estratégias que podem ser aplicadas à infraestrutura adotada, faz-se necessário a identificação das configurações das estratégias. Isso ocorre pois algumas estratégias de DR podem ter diferentes configurações, como por exemplo, a replicação de BD, que pode ser assíncrona ou semisíncrona (ver Subseção 2.4.2.2). Assim, é importante definir quais as possíveis configurações a serem avaliadas em cada estratégia aplicável.

Definir as métricas de interesse: nesta etapa da atividade, é necessário escolher as métricas de interesse que se deseja obter através da avaliação das estratégias de DR. A escolha das métricas é uma etapa crucial, pois influencia diretamente nas próximas atividades, como por exemplo, na geração dos modelos. Vale ressaltar que, neste trabalho, são adotadas as métricas de disponibilidade, indisponibilidade, RTO, RPO e custo como métricas de interesse no processo de avaliação das estratégias de DR.

Definir os cenários a serem avaliados: uma vez identificadas as estratégias, suas respectivas configurações e as métricas de interesse, é necessário definir os cenários a serem avaliados. Os cenários são compostos por uma estratégia, uma configuração e uma possível variante. Por exemplo, se a estratégia de *backup* foi escolhida para ser avaliada, dois possíveis cenários seriam: (1) a estratégia de *backup* com a política de *backup* completo e periodicidade de 12 horas entre os *backups*; e (2) a estratégia de *backup* com a política de *backup* completo e periodicidade de 24 horas entre os *backups*. Dessa forma, é fundamental realizar as atividades e etapas anteriores da metodologia, de modo a definir adequadamente os cenários a serem avaliados.

4.4 REALIZAR OS EXPERIMENTOS

O processo de avaliação das infraestruturas computacionais e estratégias de DR proposto neste trabalho, é baseado principalmente no uso de modelos e métodos. No entanto, para que os modelos representem corretamente os cenários avaliados, é preciso que os parâmetros de entrada desses modelos sejam os mais próximos possíveis da realidade. Dessa forma, quando os parâmetros que caracterizam um componente ou uma operação não são fornecidos pelo fabricante ou encontrados na literatura, faz-se necessário a execução de experimentos para obtenção dos mesmos. Nesse contexto, esta atividade consiste na realização de experimentos (quando necessários) e é divida em duas etapas: (1) obter parâmetros de entrada para os modelos e (2) obter os resultados das métricas escolhidas.

Obter parâmetros de entrada para os modelos: nesta etapa, são montados ambientes de testes que representem os cenários a serem avaliados. No ambiente de teste montado são conduzidos experimentos, a fim de coletar dados que servirão como parâmetros de entrada para os modelos. Vale ressaltar que a execução de experimentos pode necessitar da criação de ferramental de apoio.

Obter os resultados das métricas escolhidas: nesta etapa, são conduzidos experimentos no ambiente de testes montado para avaliar as métricas de interesse escolhidas. Os resultados obtidos a partir da execução desses experimentos podem ser utilizados para verificar se os modelos produzem resultados de acordo com ambientes reais. Para essa etapa, esta tese adota a execução de experimentos de injeção de falhas (ARLAT et al., 1993) para verificar os resultados referentes a métrica de disponibilidade dos cenários avaliados.

4.5 GERAR MODELOS PARA AS INFRAESTRUTURAS

nesta atividade, a partir dos cenários e dos ambientes de testes definidos, são gerados os modelos para avaliação desses cenários. Assim, essa atividade é divida em duas etapas: (1) gerar os modelos DSPNs e (2) gerar o modelo de custo.

Gerar os modelos DSPNs: nesta etapa, primeiramente são identificados e buscados os parâmetros de entrada que podem ser obtidos através da literatura e material

técnico. Uma vez obtidos os valores desses parâmetros e dos parâmetros colhidos através dos experimentos realizados na atividade anterior, é possível gerar os modelos DSPNs para avaliar os cenários escolhidos. Para dar suporte a essa etapa, são propostos, nesta tese, modelos DSPNs capazes de representar e avaliar infraestruturas computacionais e estratégias de DR, os quais são apresentados no Capítulo 5. Além disso, pode ser utilizada uma ferramenta visual que auxilia no desenvolvimento dos modelos e permite o cálculo das métricas escolhidas. Dentre algumas ferramentas existentes para esse propósito, pode-se citar o Mercury (SILVA et al., 2015).

Gerar o modelo de custo: caso se deseje avaliar os custos dos cenários definidos, nesta etapa é estabelecido o modelo de custo a ser adotado. Da mesma forma que na etapa anterior, esta tese apresenta modelos de custo para auxiliar a(o) profissional que conduz as atividades desta metodologia (ver Capítulo 5).

4.6 AVALIAR MODELOS

A partir dos modelos DSPNs e de custos gerados na atividade anterior, esta atividade permite a avaliação e obtenção dos resultados de cada cenário definido. Assim, essa atividade é composta pelas seguintes etapas: (1) avaliar os modelos DSPNs, (2) Validar os resultados dos modelos, e (3) avaliar o custo.

Avaliar os modelos DSPNs: nesta etapa, são avaliados os modelos DSPNs criados na atividade anterior para obtenção das métricas escolhidas. Esse processo de avaliação é realizado através de análise numérica ou simulação dos modelos DSPN, a depender do cenário avaliado. De acordo com o cenário, também é preciso identificar onde há necessidade de uma avaliação hierárquica dos modelos. O processo de avaliação hierárquica divide os modelos em submodelos, de modo a realizar a avaliação das métricas de interesse. Essa técnica visa evitar problemas de *largness* (explosão do espaço de estados) e *stiffness* (parâmetros de magnitudes muito diferentes) na avaliação dos modelos DSPNs (MALHOTRA; TRIVEDI, 1994; TRIVEDI et al., 2001; SAHNER; TRIVEDI; PULIAFITO, 2012).

Validar os resultados dos modelos: nesta etapa da atividade, quando necessário, é feita a comparação das métricas obtidas através dos experimentos e dos modelos DSPNs. Esta etapa visa fornecer um embasamento quantitativo para o modelo analisado, evidenciando se o modelo pode representar com precisão um ambiente real. Quando os valores computados pelos modelos sejam matematicamente correspondes (satisfatórios) aos valores obtidos nos experimentos, o processo seque para a próxima atividade. Caso contrário, é necessário que o processo volte para a atividade de **Planejar a implementação das estratégias de DR** e sejam realizadas as devidas alterações e, dessa forma, o modelo seja reavaliado.

Avaliar o custo: nesta etapa é avaliado o custo de cada cenário, de acordo com o modelo de custo definido na atividade anterior. Vale ser ressaltado que para o modelo de custo apresentado nesta tese, são necessários parâmetros de entrada que dependem dos

resultados das métricas (ex.: disponibilidade) obtidas através da avaliação dos modelos DSPNs. Assim, somente após a avaliação dos modelos DSPNs dos cenários, é possível avaliar o custo dos mesmos.

4.7 REALIZAR AVALIAÇÃO MULTICRITÉRIO

A partir dos resultados obtidos na atividade anterior, a(o) profissional que conduz o processo de avaliação e seleção das estratégias de DR pode escolher utilizar um método de decisão multicritério. O uso de tal método visa selecionar as estratégias ou cenários que melhor se adéquem a um conjunto de critérios. Assim, para a utilização do método são realizadas duas etapas: (1) definir a função multicritério e (2) avaliar a função multicritério.

Definir a função multicritério: nesta etapa é formulada a função multicritério. A função é definida a partir da escolha de critérios (métricas definidas na atividade da Seção 4.3) e de objetivos que maximizem ou minimizem cada critério escolhido. Além disso, opcionalmente, é possível ponderar os critérios escolhidos para a função.

Avaliar a função multicritério: após a definição da função, os valores dos critérios de cada cenário são normalizados e avaliados. Caso seja necessário avaliar os cenários com uma função multicritério diferente, é possível definir uma nova função e avaliá-la. Como resultado da avaliação da função, é gerado um conjunto de cenários ordenados do melhor para o pior, considerando os critérios definidos na função. Assim, o método de seleção multicritério pode auxiliar empresas e projetistas no processo de tomada de decisão, especialmente quando existem vários critérios conflitantes.

4.8 CONSIDERAÇÕES FINAIS

Este capítulo apresentou a metodologia proposta para guiar o uso integrado de modelos e métodos no processo de avaliação e seleção de estratégias de DR que melhor se adéquem a um conjunto de critérios. A metodologia é detalhada através de um conjunto de atividades e etapas que abordam desde o entendimento das infraestruturas e dos sistemas até a realização da avaliação multicritério das estratégias ou de cenários. As atividades apresentadas possibilitam a definição e avaliação de cenários com infraestruturas computacionais que adotam (ou irão adotar) estratégias de DR por meio de modelos DSPNs e de custo. Além disso, as atividades também descrevem a adoção de métodos de tomada de decisão multicritério para auxiliar na seleção de estratégias de DR que atendam às necessidades específicas das empresas, de projetistas ou de pessoas interessadas na adoção de estratégias de DR.

5 MODELOS

Este capítulo apresenta os modelos desenvolvidos para avaliar as infraestruturas computacionais que empregam as estratégias de DR adotadas neste trabalho. Assim, primeiramente são descritos os modelos DSPNs desenvolvidos para as estratégias de DR adotadas. Em seguida, é detalhado o modelo de custo empregado para estimar os custos das infraestruturas computacionais que adotam as estratégias.

5.1 MODELOS DSPNS PARA ESTRATÉGIAS DE RECUPERAÇÃO DE DESASTRES

Nesta seção, são detalhados os modelos DSPNs desenvolvidos para avaliar as infraestruturas computacionais que fazem uso de estratégias de DR. Primeiro, são apresentados os modelos básicos e, em seguida, são descritos os modelos criados para cada estratégia de DR adotada.

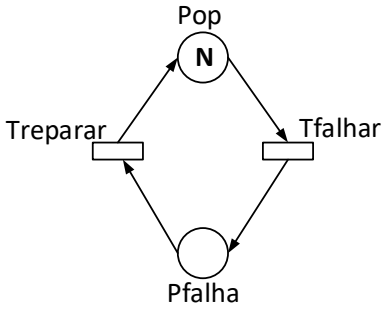
5.1.1 Modelos para os componentes básicos

As DSPNs dos componentes básicos são usadas para representar dispositivos comuns presentes em infraestruturas computacionais, como elementos de conexão de rede (ex.: *switches* ou roteadores), VMs ou a própria infraestrutura computacional (ex.: um computador ou um conjunto de computadores), onde alguns desses dispositivos podem ser hospedados. Assim, inicialmente, esses componentes são representados por DSPNs básicas que representam comportamentos de falha e reparo. Essa representação torna possível utilizar os modelos para analisar características de disponibilidade das infraestruturas, componentes, e estratégias de DR.

A Figura 8 apresenta uma DSPN básica para representar componentes ativados ou em modo *hot-standby* de uma infraestrutura. O lugar *Pop* representa o estado de funcionamento normal do componente, enquanto o lugar *Pfalha* representa um estado de falha do mesmo. A variável *N* no lugar *Pop* representa a quantidade de *tokens* presente nesse lugar, podendo também indicar o nível de redundância do componente. Por exemplo, se a DSPN representa um servidor web e *N* é definido como $N = 2$, então significa que existem dois servidores web idênticos. Ainda considerando a mesma DSPN, a transição *Tfalhar* representa o evento de falha (ex.: falha no SO). Já a transição *Treparar* representa o evento de recuperação da falha ocorrida no componente. Os valores de *delay* atribuídos às transições temporizadas *Tfalhar* e *Treparar* devem representar, respectivamente, o MTTF e o MTTR do componente. Esses valores geralmente são obtidos através de experimentos, literatura ou ainda fornecidos pelo fabricante do componente.

Considerando que exista apenas um *token* no lugar *Pop* ($N = 1$) a execução da DSPN ocorre da seguinte forma: o *token*, inicialmente no lugar *Pop*, representa que o componente está no estado operacional, funcionando normalmente. Nesse estado, a transição *Tfalhar*

Figura 8 – Modelo DSPN básico para representar componentes ativos ou em modo *hot-standby* de infraestruturas computacionais.



Fonte: Criado pelo autor.

Tabela 4 – Descrição das transições do modelo DSPN da Figura 8.

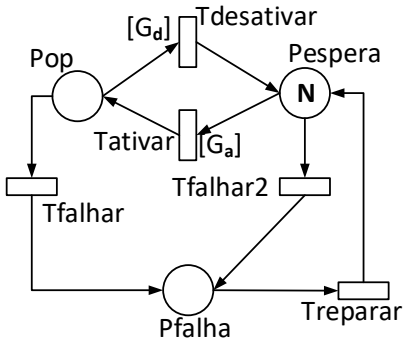
Transição	**Tipo**	**Semântica**
Tfalhar	Temporizada (exp)	*Infinite-server*
Treparar	Temporizada (exp)	*Single-server*

encontra-se habilitada. O disparo da transição *Tfalhar* indica a ocorrência de um evento de falha. Nessa ação, o *token* é consumido do lugar *Pop* e um *token* é gerado no lugar *Pfalha*. A presença de um *token* no lugar *Pfalha* indica que o componente não está operacional. Ao mesmo tempo, a presença do *token* no lugar *Pfalha* torna possível a habilitação da transição *Treparar*. Essa transição representa o evento de reparo da falha ocorrida no componente. Quando essa transição é disparada, o *token* é consumido do lugar *Pfalha* e um *token* é gerado no lugar *Pop*, indicando a operacionalidade do componente novamente. Vale ressaltar ainda que as transições temporizadas podem possuir semânticas de disparo diferentes (ver Subseção 2.3.2). É utilizado a semântica *infinite-server* para as transições que representam os eventos de falha dos componentes (ex.: *Tfalhar*), uma vez que tal configuração aumenta a probabilidade de falha quando um componente redundante é modelado. Para representar as ações de reparo dos componentes, é considerado uma única equipe de manutenção. Essa equipe é capaz de reparar as falhas ocorridas em todos os componentes dos DCs. Tal consideração implica no uso da semântica *single-server* para transições que representam o reparo dos componentes (ex.: *Treparar*). Assim, a Tabela 4 apresenta os tipos e as semânticas de disparo das transições utilizadas na DSPN que representa os componentes básicos.

Outra configuração possível para os componentes básicos é a *warm-standby*. Nesse tipo de configuração, o componente está alocado, mas não está funcionando. Isto é, precisa ser ativado para atingir o estado operacional. Essa ativação, geralmente, ocorre de forma automática, quando o sistema atinge algum estado específico (ex.: falha de um servidor de BD). A Figura 9 ilustra a DSPN criada para representar componentes configurados em modo *warm-standby*.

A DSPN de um componente configurado em modo *warm-standby* contêm três lugares e

Figura 9 – DSPN para componentes que utilizam a configuração *warm-standby*.



Fonte: Criado pelo autor.

Tabela 5 – Descrição das transições para o modelo DSPN da Figura 9.

Transição	**Tipo**	**Semântica**
Treparar, *Tativar*, *Tdesativar*	Temporizada (exp)	*Single-server*
Tfalhar, *Tfalhar2*	Temporizada (exp)	*Infinite-server*

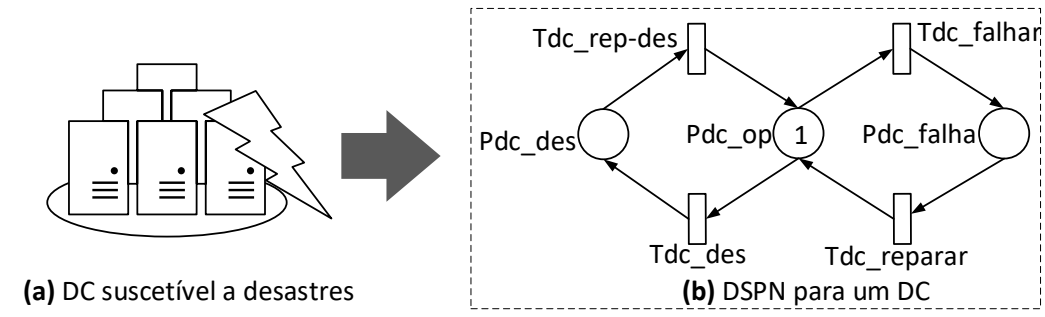
quatro transições temporizadas. O lugar *Pespera* representa o estado em que o componente aguarda para ser ativado. Nesse estado, o componente não está em operação. Já o lugar *Pop* representa o estado operacional do componente, o qual ele está ativado e funcionando normalmente, enquanto o lugar *Pfalha* representa o estado de falha do componente. A variável N no lugar *Pespera* indica o nível de redundância do componente. Ou seja, $N = 2$ indica a existência de dois componentes idênticos. As transições *Tfalhar* e *Tfalhar2* representam os eventos de falhas os quais o componente está suscetível, tanto no estado de espera (*Pespera*) quanto no estado operacional (*Pop*). Por exemplo, a ocorrência de uma falha de memória da VM. A transição *Treparar* representa o evento de recuperação da falha ocorrida no componente. Já as transições *Tativar* e *Tdesativar* representam os eventos de ativação e desativação do componente, respectivamente. Os valores de *delay* para as transições temporizadas *Treparar*, *Tfalhar* indicam, respectivamente, o MTTR e o MTTF do componente representado pelo modelo. Já os valores de *delay* para a transição *Tfalhar2* indica o tempo médio de falha do componente quando o mesmo se encontra em estado de espera. Por outro lado, os valores de *delay* para as transições *Tativar* e *Tdesativar* representam, respectivamente, os tempos médios que o componente pode levar para ser ativado ou desativado e podem variar de acordo com o tipo de componente que o modelo retrata.

Em um componente *warm-standby* (ver Figura 9), a execução do modelo acontece como se segue (considerando $N = 1$). A presença de um *token* no lugar *Pespera* indica que o componente está em estado de espera, aguardando para ser ativado. Neste estado, o componente pode ser ativado (disparo da transição *Tativar*) ou apresentar uma falha

transiente (disparo da transição *Tfalhar2*). Se o componente for ativado, então a transição *Tativar* é disparada e um *token* é consumido do lugar *Pespera* e um *token* é gerado no lugar *Pop*, indicando que o componente está ativo. Vale ressaltar que o disparo da transição *Tativar* é condicionado à função de guarda G_a, a qual define os requisitos de ativação do componente. No estado operacional (*Pop*), o componente pode apresentar uma falha transiente (disparo da transição *Tfalhar*) ou retornar ao estado de espera (*Pespera*). Para retornar ao estado de espera, a transição *Tdesativar* precisa ser disparada. O disparo pode ocorrer quando a função de guarda G_d, que define as condições de desativação do componente, são atendidas. Caso o componente atinja o estado de falha (disparo das transições *Tfalhar* ou *Tfalhar2*), um *token* é consumido pela transição disparada e um *token* é gerado no lugar *Pfalha*, indicando que o componente encontra-se em estado de falha. Nessa marcação, o componente só pode ser restaurado quando a transição *Treparar* for disparada. Quando isso acontece, um *token* é consumido do lugar *Pfalha* e outro *token* é gerado no lugar *Pespera*, indicando que o componente está esperando para ser ativado novamente. A Tabela 5 apresenta os detalhes das transições utilizadas na DSPN que representa os componentes básicos do modo *warm-standby*.

As infraestruturas computacionais representam locais onde estão presentes um conjunto de *hardware* sob a qual VMs e aplicações são executados. Por estarem em um local físico, essas infraestruturas estão suscetíveis a desastres (ex.: inundações), bem como as falhas mais comuns(ex.: falha no SO ou erro de memória) (ALEXANDER, 2008). As DSPNs criadas para representar componentes básicos mostradas anteriormente são capazes de representar a ocorrência de falhas comuns. No entanto, para representar, por exemplo, que um DC seja suscetível à falhas comuns e também a eventos de desastres, é preciso definir uma outra DSPN. A Figura 10 (*a*) ilustra um exemplo de uma representação de alto nível da infraestrutura de um DC, enquanto a Figura 10 (*b*) mostra o modelo DSPN proposto para um DC suscetível a desastres.

Figura 10 – Exemplo de uma representação de alto nível de um DC suscetível a desastres (a) e a DSPN correspondente ao mesmo (b).



Fonte: Criado pelo autor.

O modelo DSPN para o DC apresenta dois lugares onde esse não se encontra operacional (*Pdc_des* e *Pdc_falha*), visto que o modelo considera tanto as falhas transientes quantos

Tabela 6 – Descrição das transições para o modelo DSPN da Figura 10 (b).

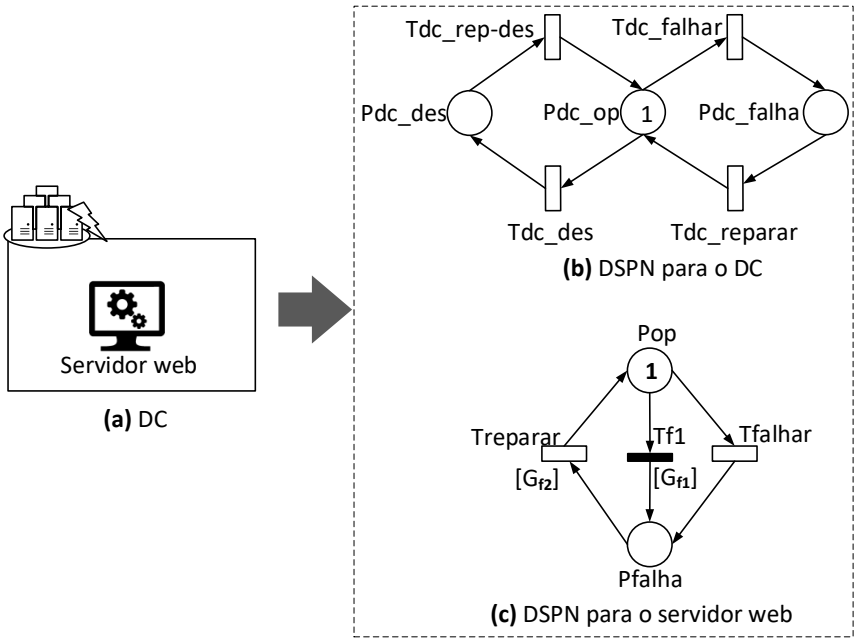
Transição	**Tipo**	**Semântica**
Tdc_reparar, *Tdc_rep-des*	Temporizada (exp)	*Single-server*
Tdc_falhar, *Tdc_des*	Temporizada (exp)	*Infinite-server*

os eventos de desastres que podem ocorrer na infraestrutura. Nesse modelo, a presença de um *token* no lugar *Pdc_op*, indica a operacionalidade do DC. Nesse estado, podem ser habilitadas as transições *Tdc_des* ou *Tdc_falhar*. Se um evento de desastre acontecer, a transição *Tdc_des* é disparada. Então, o *token* presente no lugar *Pdc_op* é consumido e é gerado um *token* no lugar *Pdc_des*, indicando que a infraestrutura do DC não está funcionando devido à ocorrência de um evento de desastre. Nesse estado, a transição *Tdc_rep-des* pode ser habilitada. Essa transição representa o evento de reparo do DC. Quando ocorre o disparo dessa transição, o *token* presente no lugar *Pdc_des* é consumido e é gerado um *token* no lugar *Pdc_op*, indicando que o DC voltou a operar normalmente. Porém, quando o *token* está no lugar *Pdc_op* e ocorrer uma falha transiente, a transição *Tdc_falhar* é disparada. Dessa forma, o *token* presente no lugar *Pdc_op* é consumido e um *token* é gerado no lugar *Pdc_falha*. A presença de um *token* neste lugar indica que o DC não está operacional e que a causa foi um evento de falha transiente. A partir dessa marcação, a transição *Tdc_reparar* pode ser disparada, representando a ação de reparo da falha. Quando o disparo da transição acontece, o *token* no lugar *Pdc_falha* é consumido e um *token* é gerado no lugar *Pdc_op*, indicando que o DC voltou a funcionar normalmente. A Tabela 6 resume as características das transições utilizadas na DSPN criada para representar um DC. Os valores de *delay* atribuídos às transições temporizadas *Tdc_des* e *Tdc_rep-des* representam, respectivamente, o tempo médio para ocorrência de desastres e o tempo médio para reparar a infraestrutura após eventos de desastres. Geralmente esses valores são obtidos através da literatura. Já os valores de *delay* atribuídos às transições *Tdc_falhar* e *Tdc_reparar* representam, respectivamente, o MTTF e o MTTR correspondente a infraestrutura representada.

É preciso ressaltar ainda que alguns componentes executam sobre uma infraestrutura computacional (ex.: uma VM ou um servidor web). Sendo assim, existe uma relação de dependência entre a infraestrutura e tais componentes. Por exemplo, se um evento de falha ou desastre ocorre em um DC e ele fica em estado de falha, então automaticamente os servidores web que estavam executando sobre aquela infraestrutura também devem falhar. Dessa forma, faz-se necessário definir essa relação de dependência entre os componentes nos modelos DSPNs.

A Figura 11 (a) mostra um exemplo de uma representação de alto nível para um servidor web que é executado em um DC. Assim, para a representação de tal configuração têm-se duas DSPNs: a primeira representando o DC propriamente dito (Figura 11 (b))

Figura 11 – Exemplo um servidor web que executa em um DC.



Fonte: Criado pelo autor.

Tabela 7 – Descrição das transições para os modelos DSPNs das Figuras 11 (b) e (c).

Transição	**Tipo**	**Semântica**	**Peso**	**Prioridade**
Tdc_reparar, *Tdc_rep-des*	Temporizada (exp)	*Single-server*	-	-
Treparar	Temporizada (exp)	*Single-server*	-	-
Tfalhar	Temporizada (exp)	*Infinite-server*	-	-
Tdc_falhar, *Tdc_des*	Temporizada (exp)	*Infinite-server*	-	-
Tf1	Imediata	-	1	1

Tabela 8 – Funções de guarda da DSPN da Figura 11 (c).

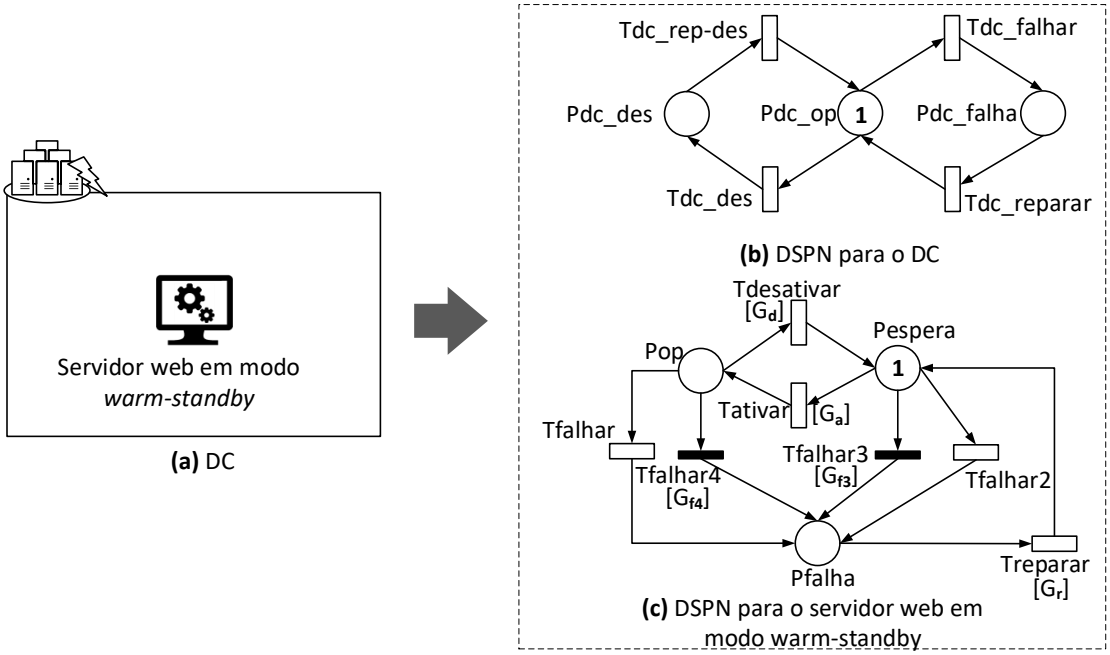
Guarda	**Descrição/Condição**	**Função de habilitação**
G_{f1}	Verifica se a infraestrutura a qual o componente é hospedado está em estado de falha	$\#Pdc\_op = 0$
G_{f2}	Verifica se a infraestrutura a qual o componente é hospedado está operacional	$\#Pdc\_op = 1$

e a segunda modelando o servidor web (Figura 11 (c)). Para caracterizar a relação de dependência entre o DC e o servidor web foi adicionada uma transição imediata a DSPN (*Tf1*) e duas funções de guarda (G_{f1} e G_{f2}). Essas funções de guarda verificam se o DC está ou não funcionando corretamente. Caso o DC esteja em um estado de falha, a função de guarda G_{f1} habilita o disparo da transição *Tf1*. Assim, o *token* presente no lugar *Pop*, que representa a operacionalidade do servidor web, é consumido e um *token* é gerado no lugar *Pfalha*, indicando que o servidor web falhou. Já a função de guarda G_{f2} presente na

transição *Treparar* serve para indicar que o reparo do servidor web só pode ser executado se o DC estiver funcionando normalmente. Caso essa condição não seja satisfeita, a transição *Treparar* não pode ser disparada e o servidor web não pode voltar a funcionar. A Tabela 7 apresenta as características das transições empregadas nas DSPNs para o DC e para o servidor web. A Tabela 8 exibe as funções de guarda utilizadas na DSPN do servidor web (Figura 11 (c)).

Por último, é apresentado o cenário no qual um componente é hospedado em um DC, porém esse componente é configurado em modo *warm-standby*. A Figura 12 (a) ilustra um exemplo para representar um servidor web operando em modelo *warm-standby* em um DC. As Figuras 12 (b) e (c) apresentam as DSPNs do DC e do servidor web em modo *warm-standby*, respectivamente.

Figura 12 – Exemplo de um servidor web em modo *warm-standby* que executa em um DC.



Fonte: Criado pelo autor.

Tabela 9 – Descrição das transições dos modelos DSPNs das Figuras 12 (b) e (c).

Transição	**Tipo**	**Semântica**	**Peso**	**Prioridade**
Tdc_reparar, *Tdc_rep-des*	Temporizada (exp)	*Single-server*	-	-
Tdc_falhar, *Tdc_des*	Temporizada (exp)	*Infinite-server*	-	-
Treparar, *Tativar*, *Tdesativar*	Temporizada (exp)	*Single-server*	-	-
Tfalhar, *Tfalhar2*	Temporizada (exp)	*Infinite-server*	-	-
Tfalhar3, *Tfalhar4*	Imediata	-	1	1

Na DSPN da Figura 12 (c) o número de *tokens* no lugar *Pespera* representa o nível de redundância do componente. A presença de um *token* nesse lugar indica que o componente

Tabela 10 – Funções de guarda da DSPN da Figura 12 (c).

Guarda	Descrição/Condição	Função de habilitação
G_{f3}, G_{f4}	Verifica se a infraestrutura a qual o componente é hospedado está em estado de falha	$\#Pdc\_op = 0$
G_r	Verifica se a infraestrutura a qual o componente é hospedado está operacional	$\#Pdc\_op = 1$

está em estado de espera, aguardando para ser ativado. Nesse estado, o componente pode: ser ativado (disparo da transição *Tativar*), sofrer uma falha transiente (disparo da transição *Tfalhar2*) ou ficar inoperante devido à uma falha na infraestrutura onde o mesmo está hospedado (disparo da transição *Tfalhar3*). Se o componente for ativado, a transição *Tativar* é disparada, consumindo um *token* do lugar *Pespera* e gerando um *token* no lugar *Pop*, indicando que o componente está ativo. No estado operacional (*Pop*), o componente pode: apresentar uma falha transiente (disparo da transição *Tfalhar*) ou ficar inoperante devido à uma falha sofrida no DC (disparo da transição *Tfalhar4*). As guardas G_{f3} e G_{f4} verificam se o DC está funcionando corretamente. Se alguma falha ocorrer quando um *token* estiver presente nos lugares *Pespera* ou *Pop*, um *token* é consumido pela transição disparada e um *token* é gerado no lugar *Pfalha*. Quando um *token* está presente no lugar *Pfalha*, o componente só pode ser reparado se a infraestrutura sob a qual ele está hospedado estiver funcionando. Esse comportamento é modelado pela função de guarda G_r.

As guardas G_a e G_d, associadas às transições *Tativar* e *Tdesativar*, estão relacionadas as regras de ativação e desativação do componente, respectivamente. Assim, elas podem variar de acordo com a configuração adotada pela(o) projetista (ex.: ativar o servidor web *warm-standby* apenas quando um outro servidor web falhar). Dessa forma, suas funções de habilitação não estão definidas. A Tabela 9 exibe as características das transições usadas nas DSPNs para o DC e para o servidor web em modo *warm-standby* (ver Figuras 12 (b) e (c)). Já a Tabela 10 apresenta as funções de guardas utilizadas na DSPN da Figura 12 (c).

5.1.2 Modelos para estratégia de Backup

Nesta subseção, são apresentadas as DSPNs criadas para representação das operações de *backup* e restauração de dados quando a estratégia de *backup* é adotada. Essas DSPNs são usadas junto com as DSPNs dos componentes básicos para avaliar uma infraestrutura que implementa a estratégia de *backup*. As operações de *backup* ajudam a reduzir a perda de dados cruciais aos negócios, garantindo uma cópia de segurança de tais dados, e consequentemente, podem tornar mais rápida a restauração dos sistemas de TIC. Os modelos desenvolvidos consideram a operação de *backup* completo, onde uma cópia total dos dados é realizada. A utilização dessa estratégia implica na realização de *backups* periódicos dos componentes da infraestrutura e na realização do processo de restauração

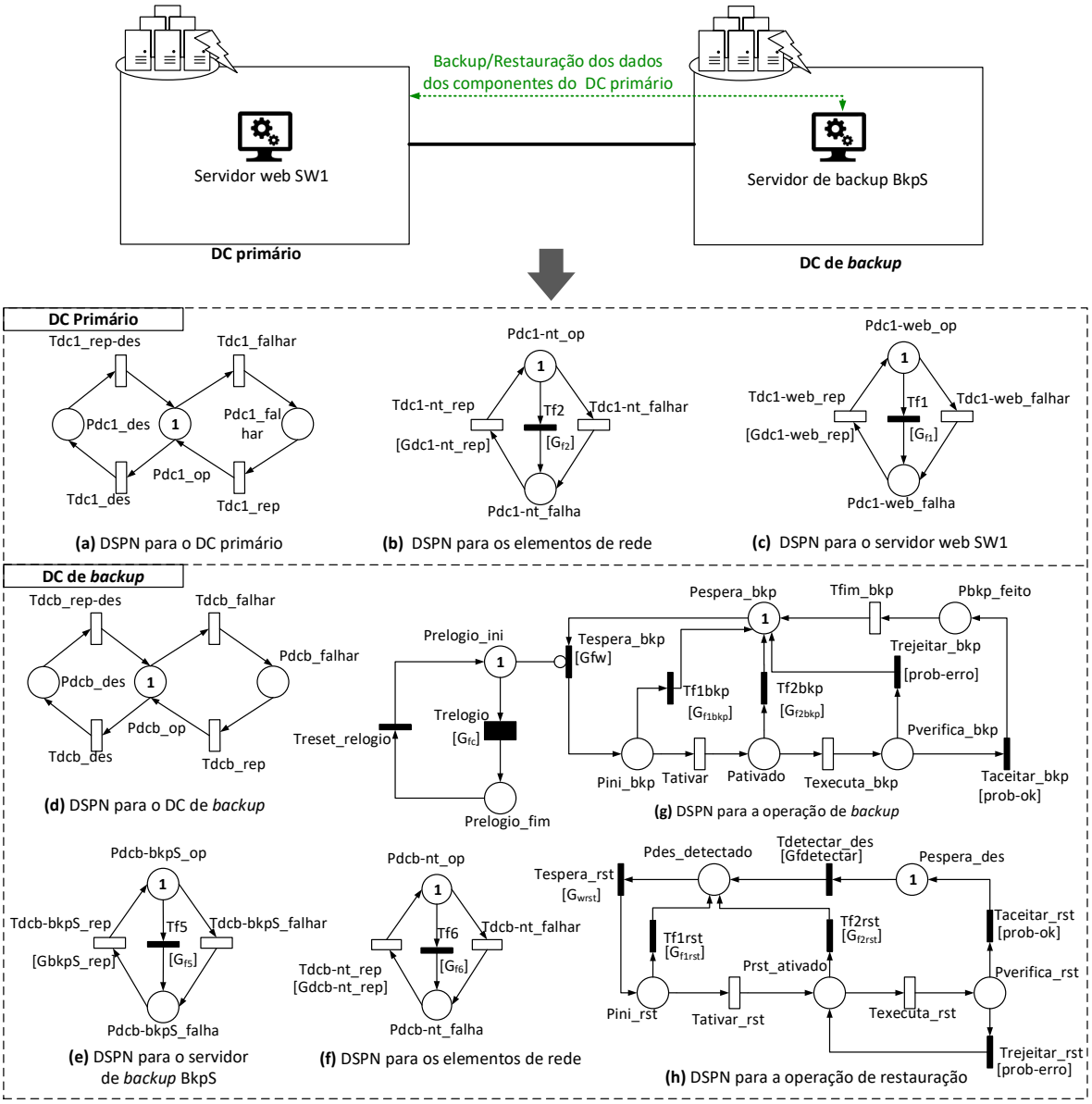
do *backup*, caso falhas ou desastres ocorram. Vale ressaltar que para a realização das operações de *backup* e restauração é considerado um servidor de *backup* para gerenciar as cópias dos dados e controlar as operações.

A Figura 13 apresenta um exemplo onde a estratégia de *backup* é adotada. Na figura é representado um DC primário, que hospeda um servidor web, e um DC de *backup* que contêm um servidor de *backup* responsável pelo *backup* e restauração do servidor SW1 em caso de falhas desse servidor. Além disso, são apresentados os modelos DSPNs criados para representar o ambiente descrito. As DSPNs das Figuras 13 (a), (b) e (c) modelam o DC primário, os elementos de rede do DC primário e o servidor SW1, respectivamente. As DSPNs das Figuras 13 (d), (e) e (f) representam, respectivamente, o DC de *backup*, o servidor de *backup* BkpS e os elementos de rede do DC. Já as Figuras 13 (g) e (h) mostram as DSPNs para as operações de *backup* e restauração do dos dados de *backup*, respectivamente. As DSPNs exibidas nas Figuras 13 (a - f) são baseadas nos modelos desenvolvidos para representar componentes básicos (apresentados na seção anterior). Quando essas DSPNs são combinadas com as DSPNs mostradas nas Figuras 13 (g) e (h), as quais foram desenvolvidas exclusivamente para modelar as operações de *backup* e restauração de dados, é possível reproduzir comportamentos da estratégia de *backup*.

As operações de *backup* e restauração só podem acontecer quando o servidor de *backup* estiver operacional, o que é indicado pela presença de um *token* no lugar *Pdcb-bkpS_op* (ver Figura 13 (e)). Esse comportamento é modelado pelo uso de funções de guarda nos modelos. Um *token* presente no lugar *Pdcb-bkpS_falha* indica a falha do servidor de *backup*. As transições *Tdcb-bkpS_falhar* e *Tdcb-bkpS_rep* representam os eventos de falha e de reparo do servidor de *backup*, respectivamente. A DSPN da Figura 13 (g) modela o comportamento da operação de *backup*. A transição determinística *Trelogio* representa a periodicidade da ação de *backup*. Quando essa transição dispara, o *token* presente no lugar *Prelogio_ini* é consumido e um outro *token* é gerado no lugar*Prelogio_fim*. A partir desse momento, enquanto não existirem *tokens* no lugar *Prelogio_ini*, a transição *Tespera_bkp* pode ser disparada. Além disso, a transição imediata *Treset_relogio* pode ser disparada, indicando uma nova contagem de tempo para a realização de uma nova operação de *backup*. Ao ser disparada, *Treset_relogio* consome um *token* do lugar*Prelogio_fim* e um *token* é gerado no lugar *Prelogio_ini*.

A execução da operação de *backup* se inicia em um estado de espera (*Pespera_bkp*), onde o tempo de espera nesse estado depende da periodicidade definida para a realização da operação de *backup*. Quando a função de guarda G_{fw} é satisfeita e não há presença de *token* no lugar *Prelogio_ini*, a transição *Tespera_bkp* pode ser disparada. Essa guarda tem a função de verificar se o servidor de *backup* está operacional. Com o disparo da transição *Tespera_bkp*, um *token* é consumido do lugar *Pespera_bkp* e outro é gerado no lugar *Pini_bkp*. Nesse estado, o *backup* pode ter seu início ativado (*Tativar*) ou falhar (*Tf1bkp*). A transição *Tativar* representa o tempo que o servidor de *backup* leva para iniciar

Figura 13 – Exemplo de um ambiente que adota a estratégia de *backup* e as DSPNs correspondentes ao ambiente exibido.



Fonte: Criado pelo autor.

o processo de backup. As falhas são detectadas pela função de guarda G_{f1bkp}, que verifica a operacionalidade do servidor de *backup* e dos componentes que estão tendo seus dados copiados. Quando o *backup* é ativado com sucesso, a transição *Texecuta_bkp* pode ser disparada. Essa transição indica a execução da operação de *backup* propriamente dita. Caso falhas sejam detectadas no servidor de *backup* ou nos componentes que estão tendo seus dados copiados (G_{f2bkp}), a transição *Tf2bkp* é disparada e o backup não é realizado. Após o disparo da transição *Texecuta_bkp*, um *token* é consumido do lugar *Pativado* e outro *token* é gerado no lugar *Pverifica_bkp*. Nesse momento é possível que as transições *Trejeitar_bkp* e *Taceitar_bkp* sejam disparadas. O disparo de cada uma dessas transições depende do *peso* definido para elas, os quais são representados pelas siglas *[prob-erro]*

Tabela 11 – Descrição das transições dos modelos DSPNs da Figura 13.

Transições	**Tipo**	**Semântica**	**Peso**	**Prioridade**
DC Primário				
Tdc1_des, *Tdc1_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tdc1_rep, *Tdc1_rep-des*	Temporizada (exp)	*Single-server*	-	-
Tdc1-web_rep, *Tdc1-nt_rep*	Temporizada (exp)	*Single-server*	-	-
Tdc1-nt_falhar, *Tdc1-web_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tf1, *Tf2*	Imediata	-	1	1
DC de *backup*				
Tdcb_des, *Tdcb_falhar*	Temporizada (exp)	*Infinity-server*	-	-
Tdcb-bkpS_falhar	Temporizada (exp)	*Infinity-server*	-	-
Tdcb_rep, *Tdcb_rep-des*	Temporizada (exp)	*Single-server*	-	-
Tdcb-bkpS_rep, *Tativar*	Temporizada (exp)	*Single-server*	-	-
Texecuta_bkp, *Tfim_bkp*	Temporizada (exp)	*Single-server*	-	-
Tativar_rst, *Texecuta_rst*	Temporizada (exp)	*Single-server*	-	-
Trelogio	Temporizada (det)	*Single-server*	-	-
Treset_relogio	Imediata	-	1	0
Tf5, *Tespera_bkp*	Imediata	-	1	1
Tf1bkp, *Tf2bkp*, *Tf1rst*, *Tf2rst*	Imediata	-	1	1
Taceitar_bkp, *Taceitar_rst*	Imediata	-	[prob-ok]	1
Trejeitar_bkp, *Trejeitar_rst*	Imediata	-	[prob-erro]	1
Tdetectar_des, *Tespera_rst*	Imediata	-	1	1

e *[prob-ok]*. Caso a transição *Trejeitar_bkp* seja disparada, um *token* é consumido do lugar *Pverifica_bkp* e um *token* é gerado no lugar *Pespera_bkp*, indicando que a operação de *backup* deve ser repetida, pois ocorreu um erro. Caso a transição *Taceitar_bkp* seja disparada, indicando o sucesso na realização da operação de *backup*, o *token* é consumido do lugar *Pverifica_bkp* e outro é gerado no lugar *Pbkp_feito*. Por fim, a transição *Tfim_bkp* significa o encerramento da operação de *backup*, consumindo o *token* do lugar *Pbkp_feito* e gerando um outro no lugar *Pespera_bkp*. Nesse ponto, a execução pode ser reiniciada quando a transição *Tespera_bkp* for disparada.

A DSPN que modela o processo de restauração de um *backup* é exibida na Figura 13 (h). Nessa operação, a execução do modelo é iniciada no lugar *Pespera_des*. Quando a função de guarda $G_{fdetectar}$ é satisfeita, a transição *Tdetectar_des* pode ser disparada. Essa função de guarda verifica a ocorrência de desastres na infraestrutura que se deseja restaurar os dados de *backup*. Ao ser disparada, a transição *Tdetectar_des* consome um *token* do lugar *Pespera_des* e gera um outro *token* no lugar *Pdes_detectado*, simbolizando que foi detectado um desastre na infraestrutura computacional que consome os serviços de *backup*.

Quando há um *token* no lugar *Pdes_detectado*, apenas a transição *Tespera_rst* pode ser disparada, se a guarda G_{wrst} for satisfeita. Essa guarda verifica se os componentes que terão seus dados restaurados já estão operacionais para iniciar o processo de restauração. Ao

Tabela 12 – Funções de guarda das DSPNs presentes na Figura 13.

Guarda	**Descrição/Condição**	**Função de habilitação**
DC Primário		
G_{f1}	Verifica se o DC de *backup* não está operacional	*#Pdc1-op = 0*
G_{f2}	Verifica se o DC primário sofreu desastres	*#Pdc1_des = 1*
G_{dc1-nt_rep}, $G_{dc1-web\_rep}$	Verifica se o DC primário está operacional	*#Pdc1-op = 1*
DC de *backup*		
G_{f5}	Verifica se o DC de *backup* não está operacional	*#Pdcb_op = 0*
G_{f6}	Verifica se o DC de *backup* sofreu desastres	*#Pdcb_des = 1*
G_{bkpS_rep}, $G_{dcb\text{-}nt\_rep}$	Verifica se o DC de *backup* está operacional	*#Pdcb_op = 1*
G_{fc}	Verifica se o servidor de *backup* está operacional	*#Pdcb-bkpS_op = 1*
G_{fw}, G_{wrst}	Verifica se o servidor de *backup* e os componentes a terem seus dados copiados/restaurados estão operacionais	(*#Pdcb-bkpS_op*=1) AND (*#Pdcb-nt_op*=1) AND (*#Pdc1-nt_op*=1) AND (*#Pdc1-web_op>0*)
G_{f1bkp}, G_{f2bkp} G_{f1rst}, G_{f2rst}	Verifica se o servidor de *backup* e os componentes a terem seus dados copiados/restaurados estão em estado de falha	(*#Pdcb-bkpS_op*=0) OR (*#Pdcb-nt_op*=0) OR (*#Pdc1-nt_op*=0) OR (*#Pdc1-web_op=0*)
$G_{fdetectar}$	Verifica a ocorrência de desastre na infraestrutura que consome os serviços de *backup* e restauração	*#Pdc1_des = 1*

disparar a transição *Tespera_rst*, um *token* é consumido do lugar *Pespera_rst* e outro *token* é gerado no lugar *Pini_rst*. Esse estado indica que a operação de restauração foi iniciada. A partir daí, a restauração é ativada (*Tativar_rst*) e depois executada (*Texecuta_rst*). Caso alguma falha seja detectada nos componentes que estão recebendo os dados ou no servidor de *backup*, a guarda G_{f1rst} ou G_{f2rst} é habilitada e a transição *Tf1rst* ou a transição *Tf2rst* é disparada.

Ao disparar a transição *Trealizar_rst*, um *token* é consumido do lugar *Prst_ativado* e outro *token* é gerado no lugar *Pverifica_rst*. A partir dessa marcação, as transições *Trejeitar_rst* e *Taceitar_rst* podem ser disparadas. O *peso* definido para cada uma dessas transições, os quais são representados pelas siglas *[prob-erro]* e *[prob-ok]*, determinam a probabilidade de disparo de cada uma delas. Caso a transição *Trejeitar_rst* seja disparada, um *token* é consumido do lugar *Pverifica_rst* e outro *token* é gerado no lugar *Prst_ativado*, sinalizando que a operação de restauração deve ser repetida, pois ocorreu um erro. Porém, se a transição *Taceitar_rst* for disparada, simboliza o sucesso da operação de restauração do *backup*. Nesse caso, um *token* é consumido do lugar *Pverifica_rst* e outro *token* é gerado no lugar *Pespera_des*. Por fim, a Tabela 11 detalha as transições usadas nos modelos DSPNs. Além disso, a Tabela 12 resume as definições das funções de guardas a serem criadas para as DSPNs da Figura 13.

5.1.2.1 Obtenção das Métricas

Esta subseção descreve como são obtidas as métricas de disponibilidade, indisponibilidade, RTO e RPO utilizando os modelos DSPNs propostos para a estratégia de *backup*. A disponibilidade pode ser obtida baseada na probabilidade de operacionalidade dos componentes da infraestrutura (ex.: servidor web, elementos de rede ou DC). Dessa forma, ela depende dos elementos modelados na infraestrutura considerada e suas dependências. Além disso, para a infraestrutura primária e seus sistemas estarem disponíveis é necessário que o servidor de *backup* não esteja realizando uma operação de recuperação de dados. Isso ocorre porque os dados dos sistemas não ficam disponíveis aos usuários enquanto uma restauração é executada. Assim, a disponibilidade para esta estratégia pode ser obtida utilizando a Equação 5.1.

$$A_{sis} = A_{cmpt} \times P_{rst} \tag{5.1}$$

onde A_{sis} é a disponibilidade da infraestrutura considerada, A_{cmpt} é a disponibilidade dos componentes da infraestrutura primária (considerando suas dependências) e P_{rst} é a probabilidade do servidor de *backup* está executando uma restauração.

Utilizando como exemplo os modelos DSPNs da Figura 13, os valores de A_{cmpt} e P_{rst} podem ser obtidos a partir das métricas definidas nas Equações 5.2 e 5.3, respectivamente. Já a indisponibilidade, é calculada utilizando a Equação 2.6.

$$A_{cmpt} = P\{(\#\text{Pdc1\_op} > 0)AND(\#\text{Pdc1-nt\_op} > 0)AND(\#\text{Pdc1-web1\_op} > 0)\} \tag{5.2}$$

$$P_{rst} = P\{\#\text{Pespera\_des} = 1\} \tag{5.3}$$

Já a métrica do RTO é calculada pelo tempo médio que o sistema leva para voltar ao estado operacional (com os dados recuperados) após um desastre. Dessa forma, é utilizada a lei de Little (LITTLE, 1961) aplicada ao modelo DSPN que representa o processo de restauração de dados (ex.: Figura 13 (h)). A lei de Little define que o tempo médio de resposta (T_{resp}) de um sistema é igual ao número de requisições no sistema (N_{rq}) dividido pela vazão do sistema (TP_s), como apresentado na Equação 5.4.

$$\mathrm{T_{resp}} = \frac{N_{\mathrm{rq}}}{\mathrm{TP_s}} \tag{5.4}$$

Quando a lei de Little é aplicada ao modelo DSPN que representa o processo de restauração de dados, é computado o tempo médio que os dados são recuperados após a infraestrutura primária sofrer um desastre, ou seja, o RTO. Considerando os modelos

DSPNs da Figura 13, a Equação 5.5 apresenta a métrica definida para obter o RTO, onde a variável B_{abr} representa o *delay* atribuído a transição *Tativar_rst*.

$$RTO = \frac{((E\{\#\text{Pdes\_detectado}\})+(E\{\#\text{Pini\_rst}\})+(E\{\#\text{Prst\_ativado}\})+(E\{\#\text{Pverifica\_rst}\}))}{(E\{\#\text{Pini\_rst}\}\times(1/B_{abr}))} \tag{5.5}$$

Por fim, o RPO é obtido pelo tempo médio decorrido desde o último *backup* até o evento de desastre. Para obter essa métrica, é aplicada a lei de Little ao modelo DSPN que representa a operação de *backup* (ex.: Figura 13 (g)). Quando a lei de Litlle é aplicada a essa DSPN, é obtido o tempo médio entre as operações de *backup*. Como essa DSPN modela a suscetividade da operação de *backup* à falhas e desastres da infraestrutura, é possível considerar esse valor como o valor do RPO. Assim, considerando as DSPNs da Figura 13, a Equação 5.6 define a métrica para a obtenção do RPO, onde a variável B_{abr} representa o *delay* atribuído a transição *Tativar*.

$$RPO = \frac{((E\{\#Pini\_bkp\})+(E\{\#Pativado\})+(E\{\#Pverifica\_bkp\})+(E\{\#Pbkp\_feito\}))}{(E\{\#Pini\_bkp\}\times(1/B_{abr}))} \tag{5.6}$$

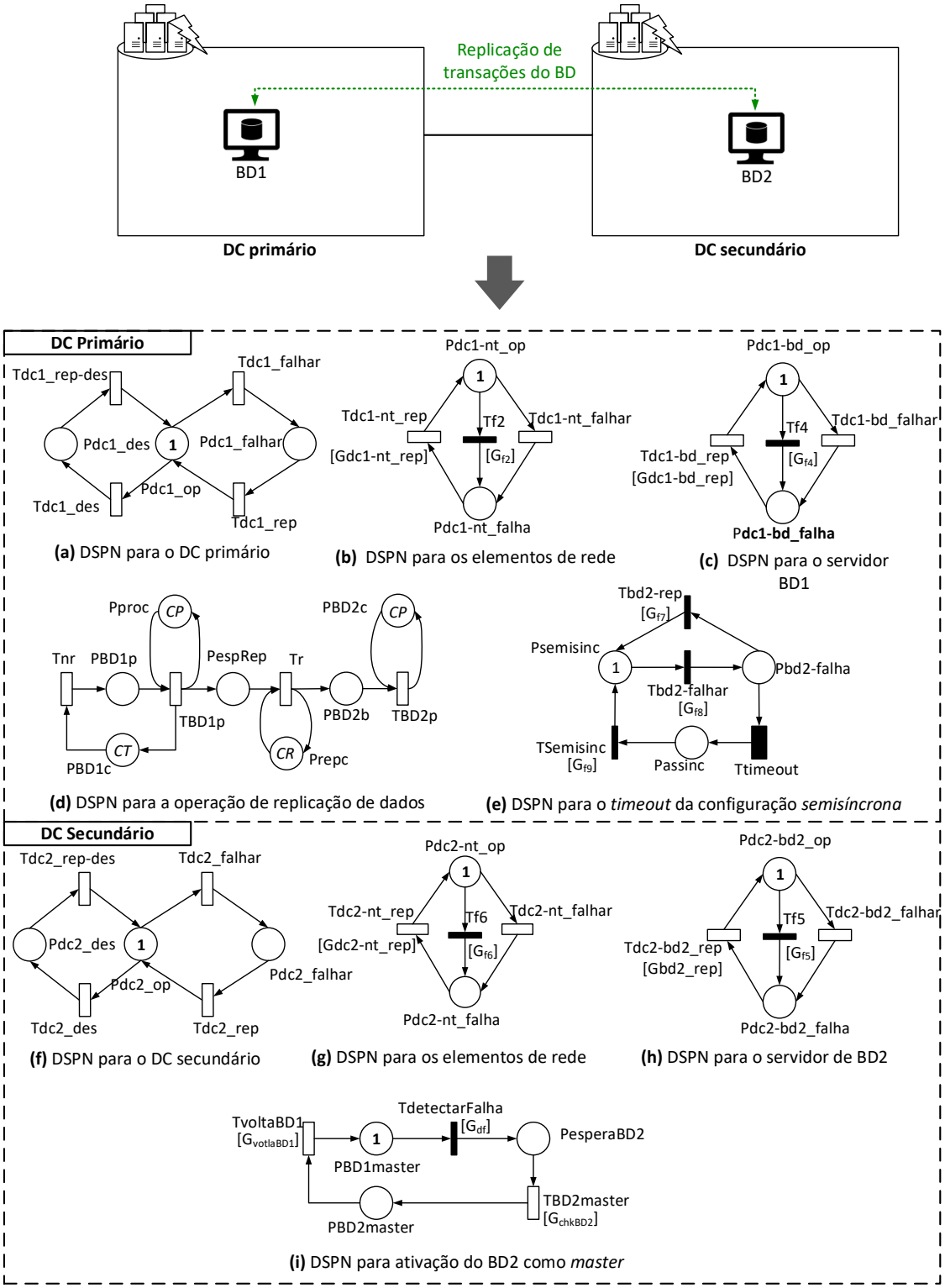
5.1.3 Modelos para estratégia de Replicação de Banco de dados

Nesta subseção, são apresentadas as DSPNs criadas para representar o uso da estratégia de replicação de BD. Os modelos DSPNs mostrados nesta subseção são usados em conjunto com os modelos DSPNs dos componentes básicos para modelar uma infraestrutura que adota a estratégia de replicação de BD. Como apresentado anteriormente na Subseção 2.4.2.2, existem diferentes modelos de replicação para BDs. Neste trabalho, é considerada a replicação *transacional* com as configurações de sincronização *assíncrona* e *semisíncrona*. A utilização desse modelo de replicação implica na utilização de dois servidores de BD: um servidor para atuar como *master* e outro para atuar como *replica*. Muito frequentemente, a implantação desses dois servidores é feita em locais físicos distintos, apesar de ser possível configurá-los até mesmo na mesma máquina física. Essa implantação em locais distintos visa prover mais tolerância à perda dos dados.

A Figura 14 exibe um exemplo de representação de alto nível de um ambiente empregando a replicação de BD e os modelos DSPNs desenvolvidos para representar a infraestrutura. São mostrados dois servidores de BD (BD1 e BD2), onde o BD1 atua como *master* e o BD2 como *replica*. Assim, as transações de persistência de dados recebidas pelo BD1 são replicadas para o BD2, seguindo a configuração assíncrona ou semisíncrona. As DSPNs das Figuras 14 (a), (b), (c), (f), (g), e (h) são baseadas nos modelos DSPNs dos componentes básicos. As DSPNs das Figuras 14 (a) e (b) modelam, respectivamente, o DC primário e os elementos de rede presentes nesse DC, os quais permite a conexão entre o DC primário e secundário. A DSPNs da Figura 14 (c) representa o servidor BD1, que executa

no DC primário. Já as DSPNs das Figuras 14 (f), (g) e (h) representam, respectivamente, o DC secundário, os elementos de rede presentes no DC secundário e o servidor BD2, que executa no DC secundário.

Figura 14 – Exemplo do uso da estratégia de replicação de BD, usando dois servidores de BD hospedados em dois DCs diferentes e as DSPNs correspondentes.



Fonte: Criado pelo autor.

Tabela 13 – Descrição das transições dos modelos DSPNs da Figura 14.

Transições	**Tipo**	**Semântica**	**Peso**	**Prioridade**
DC Primário				
Tdc1_rep, *Tdc1_rep-des*	Temporizada (exp)	*Single-server*	-	-
Tdc1_des, *Tdc1_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tdc1-bd_rep, *Tdc1-nt_rep*, *Tnr*	Temporizada (exp)	*Single-server*	-	-
Tdc1-bd_falhar, *Tdc1-nt_falhar*	Temporizada (exp)	*Infinity-server*	-	-
TBD1p, Tr, TBD1p	Temporizada (exp)	*Infinity-server*	-	-
Ttimeout	Temporizada (det)	*Single-server*	-	-
Tf2, Tf4	Imediata	-	1	1
Tbd2-falhar, Tbd2-rep, Tsemisinc	Imediata	-	1	1
DC secundário				
Tdc2_rep, Tdc2_rep-des	Temporizada (exp)	*Single-server*	-	-
Tdc2_des, Tdc2_falhar	Temporizada (exp)	*Infinity-server*	-	-
Tdc2-bd2_rep, Tdc2-nt_rep	Temporizada (exp)	*Single-server*	-	-
Tdc2-bd2_falhar, Tdc2-nt_falhar	Temporizada (exp)	*Infinity-server*	-	-
TBD2master, TvoltaBD1	Temporizada (exp)	*Single-server*	-	-
Tf5, Tf6, TdetectarFalha	Imediata	-	1	1

Para modelagem da replicação de BD foram criadas novas DSPNs apresentadas nas Figuras 14 (d), (e) e (i). A DSPN da Figura 14 (d) modela o processo de replicação assíncrona das transações recebidas pelo servidor BD1 para o servidor BD2. Assim, nessa DSPN, a transição *Tnr* representa a chegada de uma nova transação no servidor BD1. Quando essa transição dispara, um *token* é gerado no lugar *PBD1p*, habilitando a transição *TBD1p*. Tal habilitação indica que o servidor BD1 está processando a transação recebida. O disparo da transição *TBD1p* consome o *token* do lugar *PBD1p* e gera um *token* no lugar *PespRep*, simbolizando que a transação foi processada pelo servidor BD1 e salva em seu *binary log*. A transição *Tr* representa o processo de replicação da transação que já foi recebida e processada pelo servidor BD1. Quando ela dispara, o *token* presente no lugar *PespRep* é consumido e um *token* é gerado no lugar *PBD2b*, indicando que a transação foi recebida no *relay log* do servidor BD2. O final do processo de replicação é representado pela transição *TBD2b*, que modela o processo de aplicação da transação recebida no servidor BD2. Quando *TBD2b* é disparada, um *token* é consumido do lugar *PBD2b*, indicando que a transação foi salva com sucesso no servidor BD2. Por fim, as variáveis *CT*, *CP* e *CR* simbolizam a quantidade de *tokens* nos lugares *PBD1c*, *Pproc*, *Prepc* e *PBD2c*. Essas variáveis representam o tamanho máximo da fila no servidor BD1 (CT), a capacidade de processamento de transações pelos servidores de BD (CP) e a capacidade de replicação de transações do servidor BD1 (CR). As transições utilizadas nas DSPNs da Figura 14 são descritas na Tabela 13.

A DSPN da Figura 14 (e) representa o processo de *timeout* para mudança automática

da configuração semisíncrona para a assíncrona. Na configuração semisíncrona, quando ocorre uma falha no BD réplica e o BD *master* não consegue sincronizar os dados, o *master* espera um tempo pré-definido para conectar com o BD réplica. Assim, quando o tempo de espera esgota, têm-se o evento de *timeout* e o *master* passa a operar de modo assíncrono. Dessa forma, a DSPN da Figura 14 (e) só é utilizada quando se deseja avaliar a replicação de BD utilizando a configuração semisíncrona. A presença de um *token* no lugar *Psemisinc* indica que o processo de replicação está funcionando no modo semisíncrono. A função de guarda G_{f8} habilita a transição *Tbd2-falhar* caso o servidor BD2 falhe. Quando a transição *Tbd2-falhar* dispara, um *token* é consumido do lugar *Psemisinc* e um *token* é gerado no lugar *Pbd2-falha*, indicando que o servidor BD2 está em estado de falha, e o processo de replicação não está funcionando corretamente. A presença de um *token* no lugar *Pbd2-falha* habilita a transição determinística *Ttimeout*, que simboliza o tempo esperado pelo sistema para mudar da configuração semisíncrona para a configuração assíncrona. A função de guarda G_{f7} habilita a transição *Tbd2-rep* caso o servidor BD2 seja recuperado da falha antes do disparo da transição *Ttimeout*. O disparo da transição *Tbd2-rep* consome o *token* do lugar *Pbd2-falha* e gera um *token* no lugar *Psemisinc*, indicando que a replicação voltou a funcionar na configuração semisíncrona. No entanto, caso o tempo de recuperação do servidor BD2 seja menor que o tempo definido de *Ttimeout*, essa transição de *timeout* é disparada, consumindo um *token* do lugar *Pbd2-falha* e gerando um *token* no lugar *Passinc*. A presença de um *token* no lugar *Passinc* indica que processo de replicação de BD funciona sob a configuração assíncrona. Quando o servidor BD2 é recuperado, a função de habilitação da guarda G_{f9} é satisfeita e a transição *TSemisinc* é disparada. O disparo da transição *TSemisinc* consome um *token* do lugar *Passinc* e gera um *token* no lugar *Psemisinc*, indicando que a replicação voltou a operar sob a configuração semisíncrona. As funções de guarda utilizadas nas DSPNs da Figura 14 são resumidas na Tabela 14.

Quando a replicação de BD é adotada, tanto na configuração assíncrona como na semisíncrona, geralmente o servidor que funciona como *replica* pode assumir as operações caso ocorra alguma falha no *master*. Comumente, esse processo é configurado para ser executado automaticamente. Para representar esse tipo de situação, foi criada a DSPN da Figura 14 (i), que modela a ativação do servidor BD2 como *master* caso o servidor BD1 falhe. A presença de um *token* no lugar *PBD1master* indica a que o servidor BD1 é o BD *master* da infraestrutura. Caso o servidor BD1 falhe, a função de habilitação da guarda G_{fdf} é satisfeita e a transição *TdetectarFalha* é disparada. Quando a transição *TdetectarFalha* dispara, um *token* é consumido do lugar *PBD1master* e um *token* é gerado no lugar *PesperaBD2*. A presença de um *token* no lugar *PesperaBD2* torna possível a habilitação da transição *TBD2master*. No entanto, para *TBD2master* ser habilitada, também é necessário que a função de habilitação da guarda G_{chkBD2} seja satisfeita. Essa função de guarda verifica se todas as condições para o servidor BD2 assumir como *master* são satisfeitas (ex.: conexões de rede do DC secundário funcionando e servidor BD2 ativo).

Tabela 14 – Funções de guarda das DSPNs presentes na Figura 14.

Guarda	**Descrição/Condição**	**Função de habilitação**
DC prim.		
G_{f2}	Verifica se o DC primário está inoperante	*#Pdc1_op = 0*
G_{f4}	Verifica a ocorrência de desastre no DC primário	*#Pdc1_des = 1*
G_{dc1-nt_rep}, G_{dc1-bd_rep}	Verifica se o DC primário está operacional	*#Pdc1_op = 1*
G_{f7}, G_{f9}	Verifica se o servidores BD1, BD2 e as conexões de rede dos dois DCs estão operacionais	*(#Pdc1-bd_op > 0) AND (#Pdc1-nt_op > 0) AND (#Pdc2-bd2_op > 0) AND (#Pdc2-nt_op > 0)*
G_{f8}	Verifica se o servidores BD2 ou as conexões de rede do DC secundário falharam enquanto o BD1 está operacional	*(#Pdc1-bd_op > 0) AND ((#Pdc2-bd2_op = 0) OR (#Pdc2-nt_op = 0))*
DC sec.		
G_{dc2-nt_rep}, G_{bd2_rep}	Verifica se o DC secundário está operacional	*#Pdc2_op = 1*
G_{f5}	Verifica se o DC secundário está inoperante	*#Pdc1_op = 0*
G_{f6}	Verifica a ocorrência de desastre no DC secundário	*#Pdc1_des = 1*
G_{df}	Verifica se o BD1 falhou	*#Pdc1-bd_op = 0*
G_{chkBD2}	Verifica se o servidor BD2 e as conexões de rede do DC secundário estão operacionais	*(#Pdc2-bd2_op > 0) AND (#Pdc2-nt_op > 0)*
$G_{voltaBD1}$	Verifica se o servidor BD1 e as conexões de rede do DC primário estão operacionais	*(#Pdc1-bd_op > 0) AND (#Pdc1-nt_op > 0)*

Caso a guarda G_{chkBD2} seja satisfeita, a transição *TBD2master* é disparada, consumindo um *token* do lugar *PesperaBD2* e gerando um *token* no lugar *PBD2master*, indicando que a partir desse momento o servidor BD2 funciona como o *master*. Quando o servidor BD1 é recuperado da falha e todo o sistema (ex.: servidores de BD e conexões de rede) está funcionando normalmente nos dois DCs, a guarda $G_{voltaBD1}$ é satisfeita, habilitando a transição *TvoltaBD1*. Quando a transição *TvoltaBD1* é disparada, um *token* é consumido do lugar *PBD2master* e um *token* é gerado no lugar *PBD1master*, indicando que o servidor BD1 voltou a atuar como *master*. A Tabela 14 exibe as funções de guarda utilizadas nas DSPNs da Figura 14. Vale destacar que essas funções podem ser alteradas de acordo com o comportamento, restrições ou características que a infraestrutura analisada apresentar.

5.1.3.1 Obtenção das Métricas

Esta subseção descreve como são obtidas as métricas de disponibilidade, indisponibilidade, RTO e RPO utilizando os modelos DSPNs propostos para a estratégia de replicação de BD. Para essa estratégia de DR foi utilizada a execução dos modelos de forma hierárquica, de modo a evitar problemas como *largness* e *stifness* (TRIVEDI et al., 2001). Dessa forma, a

execução dos modelos é feita em duas etapas: Na primeira etapa, são calculadas as métricas de disponibilidade e indisponibilidade e métricas secundárias que servem como parâmetros de entrada para a segunda etapa de avaliação dos modelos DSPNs, onde são calculadas as métricas de RTO e RPO. As métricas secundárias calculadas na primeira etapa da avaliação são: $\mathbf{Prob_{entrada}}$, $\mathbf{Prob_{BD1op}}$, $\mathbf{Prob_{envio}}$, $\mathbf{Prob_{assinc}}$ e $\mathbf{Tmp_{ativacao}}$. A métrica $\mathbf{Prob_{entrada}}$ computa a probabilidade da chegada de novas transações para o servidor BD1. Já a métrica $\mathbf{Prob_{BD1op}}$ exibe a probabilidade do servidor BD1 estar operacional para processar as transações dos usuários. A métrica $\mathbf{Prob_{envio}}$ é a métrica secundária que representa a probabilidade dos componentes necessários para a replicação de uma transação estarem operacionais. A métrica $\mathbf{Prob_{assinc}}$ define a probabilidade do sistema passar automaticamente a adotar a configuração assíncrona, quando por padrão, opera com a configuração semisíncrona. Já a métrica $\mathbf{Tmp_{ativacao}}$ representa o tempo médio decorrido desde a falha do servidor BD1 até a ativação do servidor BD2 como *master* e é utilizada para cálculo do RTO.

Na segunda etapa da execução das DSPNs são obtidos os valores para o RPO e para a métrica secundária $\mathbf{Tmp_{aplicacao}}$ que faz parte do cálculo do RTO. $\mathbf{Tmp_{aplicacao}}$ computa o tempo médio necessário para o servidor BD2 aplicar as transações já recebidas do servidor BD1 antes de se tornar *master*. O RTO da infraestrutura é dado pelo valor máximo entre as métricas secundárias $\mathbf{Tmp_{ativacao}}$ e $\mathbf{Tmp_{aplicacao}}$ (ver Equação 5.7).

$$RTO = max(\text{Tmp\_ativacao}, \text{Tmp\_aplicacao}) \tag{5.7}$$

Utilizando as DSPNs da Figura 14 como exemplo, as Equações 5.8, 5.9, 5.10, 5.11, 5.12 e 5.13 definem como obter as métricas secundárias $\mathbf{Prob_{entrada}}$, $\mathbf{Prob_{BD1op}}$, $\mathbf{Prob_{envio}}$, $\mathbf{Prob_{assinc}}$, $\mathbf{Tmp_{ativacao}}$ e $\mathbf{Tmp_{aplicacao}}$, respectivamente. A variável R_m da Equação 5.12 representa o valor do *delay* da transição *TBD2master*. Vale ressaltar que pode ser necessário alterar essas métricas, a depender do cenário avaliado. Além disso, a métrica secundária $\mathbf{Prob_{assinc}}$ só é calculada quando a configuração semisíncrona é avaliada.

$$\begin{aligned}\text{Prob}_{\text{entrada}} = P\{&((\#\text{Pdc1-nt\_op} > 0)AND(\#\text{Pdc1-bd\_op} > 0)AND(\#\text{PBD1master} = 1))\\ &OR((\#\text{Pdc2-nt\_op} > 0)AND(\#\text{Pdc2-bd2\_op} > 0)AND(\#\text{PBD2master} = 1))\}\end{aligned} \tag{5.8}$$

$$\text{Prob}_{\text{BD1op}} = P\{(\#\text{Pdc1-bd\_op} > 0)\} \tag{5.9}$$

$$\begin{aligned}\text{Prob}_{\text{envio}} = P\{&(\#\text{Pdc1-nt\_op} > 0)AND(\#\text{Pdc1-bd\_op} > 0)AND(\#\text{Pdc2-nt\_op} > 0)\\ &AND(\#\text{Pdc2-bd2\_op} > 0)AND(\#\text{Pdc1-des} = 0)\}\end{aligned} \tag{5.10}$$

$$\text{Prob}_{\text{assinc}} = P\{(\#\text{Passinc} > 0)\} \tag{5.11}$$

$$\begin{aligned}\text{Tmp}_{\text{ativacao}} = P\{\#\text{PesperaBD2}{>}0\}/(P\{(\#\text{Pdc2-nt\_op}{>}0)AND(\#\text{Pdc2-bd\_op}{>}0)\\ AND(\#\text{Pdc1-bd\_op}{=}0)AND(\#\text{PesperaBD2}{>}0)\}/R_m)\end{aligned} \tag{5.12}$$

$$\begin{aligned}\text{Tmp}_{\text{aplicacao}} = E\{\#\text{PBD2b}\} \times (1/((1/DCT_{nr}) \times \mathbf{Prob_{entrada}} \times P\{\#\text{PBD1c}{>}0\}\times\\ \mathbf{Prob_{BD1op}} \times P\{\#\text{Pproc}{>}0\} \times \mathbf{Prob_{envio}} \times P\{\#\text{Prepc}{>}0\}))\end{aligned} \tag{5.13}$$

Para a estratégia de replicação de BD, as métricas de disponibilidade e RPO irá depender da configuração utilizada (assíncrona ou semisíncrona) e também do cenário avaliado. Considerando o cenário e modelos DSPNs da Figura 14, na replicação assíncrona a disponibilidade é obtida pela probabilidade de se ter os elementos de rede do DC primário e o servidor BD1 operacionais ou os elementos de rede do DC secundário, o servidor BD2 operacional e o servidor BD2 ativado como *master*. Assim, a disponibilidade na configuração assíncrona pode ser obtida utilizando a Equação 5.14.

$$\begin{aligned}\text{A}_{\text{assinc}} = P\{((\#\text{Pdc1-nt\_op} > 0)AND(\#\text{Pdc1-bd\_op} > 0)AND(\#\text{PBD1master} = 1))\\ OR((\#\text{Pdc2-nt\_op} > 0)AND(\#\text{Pdc2-bd2\_op} > 0)AND(\#\text{PBD2master} = 1))\}\end{aligned} \tag{5.14}$$

Para a configuração semisíncrona, a disponibilidade é obtida pela probabilidade de se ter os elementos de rede dos DCs primário e secundário, o servidor BD1 e o servidor BD2 operacionais; ou os elementos de rede do DC primário e o servidor BD2 operacionais, além do servidor BD2 estar ativado como *master*; ou ainda os elementos de rede do DC primário e o servidor BD1 operacionais, além do método de replicação ter mudado para o assíncrono. Dessa forma, a disponibilidade na configuração semisíncrona pode ser obtida utilizando a Equação 5.15. Vale ressaltar que para obtenção da indisponibilidade tanto para a configuração assíncrona quanto para a semisíncrona, é utilizada a Equação 2.6.

$$\begin{aligned}\text{A}_{\text{semisinc}} = P\{((\#\text{Pdc1-nt\_op}{>}0)AND(\#\text{Pdc1-bd\_op}{>}0)AND(\#\text{PBD1master}{=}1)\\ AND(\#\text{Pdc2-nt\_op}{>}0)AND(\#\text{Pdc2-bd2\_op}{>}0)AND(\#\text{Psemisinc}{=}1))\\ OR((\#\text{Pdc1-nt\_op}{>}0)AND(\#\text{Pdc1-bd\_op}{>}0)AND(\#\text{PBD1master}{=}1)\\ AND(\#\text{Passinc}{=}1))OR((\#\text{Pdc2-nt\_op}{>}0)AND(\#\text{Pdc2-bd2\_op}{>}0)\\ AND(\#\text{PBD2master}{=}1))\}\end{aligned} \tag{5.15}$$

O RPO na configuração assíncrona é definido pelo tempo médio que as transações levam para serem replicadas, visto que se um desastre acontecer neste intervalo de tempo, as transações que ainda não foram replicadas são desfeitas. A Equação 5.16 apresenta como é obtido o valor dessa métrica na configuração assíncrona.

$$\begin{aligned} \mathrm{RPO}_{\mathrm{assinc}} = E\{\#PespRep\} \times (1/((1/DCT_{nr}) \times \mathbf{Prob_{entrada}} \times P\{\#PBD1c{>}0\} \times \\ \mathbf{Prob_{BD1op}} \times P\{\#Pproc{>}0\})) \end{aligned} \tag{5.16}$$

Por fim, o RPO na configuração semisíncrona é definido pelo tempo médio que as transações levam para serem replicadas do BD1 para o BD2 quando o método de replicação é alternado automaticamente para o assíncrono. A Equação 5.17 apresenta como é obtido o valor dessa métrica na configuração semisíncrona.

$$\begin{aligned} \mathrm{RPO}_{\mathrm{semisinc}} = E\{\#PespRep\} \times (1/((1/DCT_{nr}) \times \mathbf{Prob_{entrada}} \times P\{\#PBD1c{>}0\} \times \\ \mathbf{Prob_{BD1op}} \times P\{\#Pproc{>}0\})) \times \mathbf{Prob_{assinc}} \end{aligned} \tag{5.17}$$

5.1.4 Modelos para estratégia de Replicação de ambiente

Esta subseção apresenta os modelos DSPNs criados para representar o uso da estratégia de replicação de ambiente. Essa estratégia consiste na adoção de duas infraestruturas similares para executar o(s) mesmo(s) sistema(s). Geralmente, essas infraestruturas estão em locais físicos distintos, a fim de evitar que a ocorrência de desastres afete ambas as infraestruturas ao mesmo tempo. Vale destacar ainda que, para essa estratégia, quando é necessário a sincronização de dados entre a infraestrutura primária e sua(s) réplica(s) é preciso utilizar uma segunda estratégia de DR, como por exemplo, a replicação de BD.

Para representar cenários que adotam a estratégia de replicação de ambiente, são usados majoritariamente as DSPNs baseadas nos componentes básicos apresentados na Subseção 5.1.1. Porém, especialmente para representar situações onde são empregadas configurações de replicação *active-standby* (ver Subseção 2.4.2.3), são necessárias novas DSPNs para modelar o comportamento observado nessas configurações. No modelo de replicação *active-standby*, a réplica da infraestrutura não divide carga de trabalho com a infraestrutura primária. Quando a infraestrutura primária falha, é necessário que a infraestrutura secundária seja configurada para processar a carga de trabalho. Geralmente, essa ativação ou configuração da infraestrutura secundária é feita de forma automática por um componente/serviço de monitoramento externo às infraestruturas primária e secundária (ANDRADE et al., 2017). Dessa forma, nesta subseção, são apresentadas as DSPNs criadas para o monitoramento da infraestrutura primária e para ativação de uma infraestrutura secundária.

A Figura 15 exibe um ambiente onde é empregado a replicação de ambiente. Nesse exemplo, o DC primário hospeda um servidor web (SW1). O DC secundário é uma a

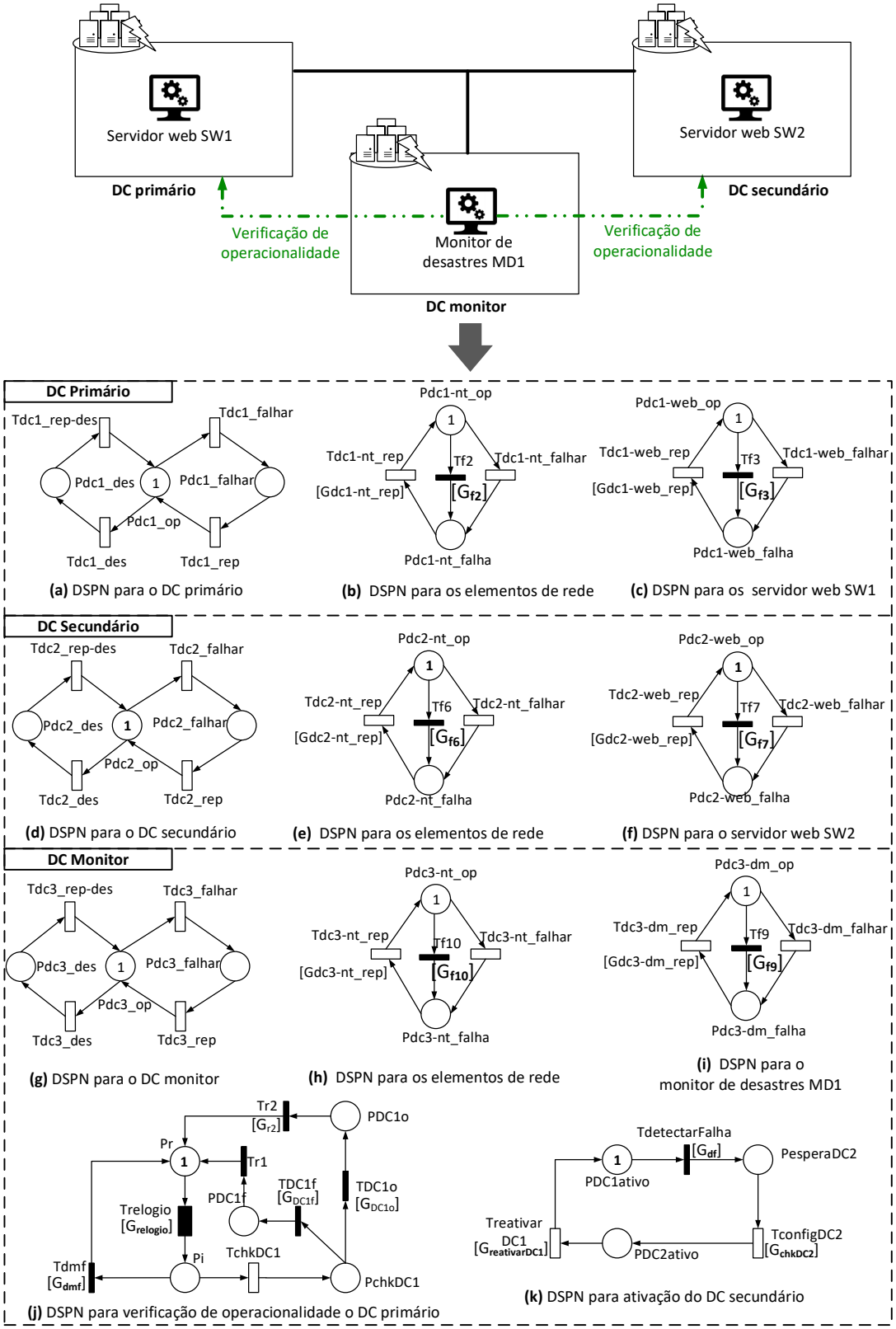
réplica do DC primário, hospedando um servidor web (SW2) com as mesmas características do servidor SW1. Externamente aos dois DCs, em um terceiro DC, a VM do monitor de desastres MD1 é responsável por verificar a operacionalidade dos DCs primário e secundário. Assim, quando o DC primário falha ou sofre um desastre, não sendo capaz de processar a carga de trabalho, o monitor MD1 ativa o DC secundário e o servidor SW2 para processar a carga de trabalho até que o DC primário seja reparado. Vale destacar ainda que o DC secundário, neste exemplo, opera no modo *hot-standby*.

As DSPNs exibidas nas Figuras 15 (a), (b), (c), (d), (e), (f), (g), (h) e (i) são baseadas nos modelos DSPNs dos componentes básicos. As Figuras 15 (a), (b) e (c) representam, respectivamente, as DSPNs do DC primário, dos elementos de rede presentes do DC primário e do servidor web SW1. As DSPNs das Figuras 15 (d), (e) e (f) modelam o DC secundário, os elementos de rede presentes no DC secundário e o servidor web SW2, respectivamente. Já as DSPNs das Figuras 15 (g), (h) e (i) correspondem, respectivamente, ao DC monitor, os elementos de rede presentes no DC monitor e o monitor de desastres MD1. Têm-se, por fim, as DSPNs criadas para monitorar a operacionalidade do DC primário e para a ativação do DC secundário, conforme exibidas nas Figuras 15 (j) e (k).

O modelo DSPN da Figura 15 (j), que monitora a operacionalidade do DC primário, inicia sua execução com a presença de um *token* no lugar *Pr*. Caso o monitor MD1 esteja operacional (verificação feita pela guarda $G_{relogio}$) e exista um *token* no lugar *Pr*, a transição *Trelogio* pode ser disparada. O disparo dessa transição indica o início do processo de verificação do DC primário e seus componentes. O intervalo de verificação depende do *delay* atribuído à transição *Trelogio*. Quando disparada, a transição *Trelogio* consome um *token* do lugar *Pr* e gera um *token* no lugar *Ti*. Caso o monitor MD1 venha a falhar, a guarda G_{dmf} habilita a transição *Tdmf* que dispara, consumindo o *token* do lugar *Pi* e gerando um *token* no lugar *Pr*. Caso o monitor MD1 continue operacional, a transição *TchkDC1* dispara, consumindo o *token* do lugar *Pi* e gerando um *token* no lugar *PchkDC1*. Nesse ponto, é executada a verificação da operacionalidade. Caso o DC primário e seus componentes estejam operacionais, a guarda G_{DC1o} é satisfeita e a transição *TDC1o* é disparada, gerando um *token* no lugar *PDC1o*. Porém, caso o DC primário ou algum de seus componentes apresentem falha, a guarda G_{DC1f} é satisfeita e a transição *TDC1f* é disparada, gerando um *token* no lugar *PDC1f*. Quando o *token* chega ao lugar *PDC1o*, a transição *Tr2* só é disparada quando o DC primário está operacional e ativo. Isto é, a presença de um *token* no lugar *PDC1ativo* (Figura 15 (k)) habilita a guarda G_{r2}, presente na transição *Tr2*. Por outro lado, quando o *token* chega ao lugar *PDC1f*, a transição *Tr1* pode ser disparada imediatamente. Com o disparo de *Tr1* ou *Tr2*, o *token* é consumido de *PDC1f* ou *PDC1o* e um *token* é gerado em *Pr*, tornando possível uma nova verificação. A Tabela 15 detalha as transições utilizadas nas DSPNs exibidas na Figura 15 enquanto a Tabela 16 descreve as funções de guardas das DSPNs.

O modelo DSPN da Figura 15 (k) controla qual das infraestruturas (DC primário ou

Figura 15 – Exemplo da estratégia de replicação de ambiente e as DSPNs correspondentes a mesma.



Fonte: Criado pelo autor.

secundário) está ativa e recebendo carga de trabalho. Por padrão, o DC primário recebe a carga de trabalho. Isso é representado pela presença de um *token* no lugar *PDC1ativo*.

Tabela 15 – Descrição das transições dos modelos DSPNs da Figura 15.

Transições	**Tipo**	**Semântica**	**Peso**	**Prioridade**
DC Primário				
Tdc1_rep, Tdc1_rep-des	Temporizada (exp)	*Single-server*	-	-
Tdc1_des, Tdc1_falhar	Temporizada (exp)	*Infinite-server*	-	-
Tdc1-web_rep, Tdc1-nt_rep	Temporizada (exp)	*Single-server*	-	-
Tdc1-web_falhar, Tdc1-nt_falhar	Temporizada (exp)	*Infinity-server*	-	-
Tf2, Tf3	Imediata	-	1	1
DC secundário				
Tdc2_rep, Tdc2_rep-des	Temporizada (exp)	*Single-server*	-	-
Tdc2_des, Tdc2_falhar	Temporizada (exp)	*Infinite-server*	-	-
Tdc2-web_rep, Tdc2-nt_rep	Temporizada (exp)	*Single-server*	-	-
Tdc2-web_falhar, Tdc2-nt_falhar	Temporizada (exp)	*Infinity-server*	-	-
Tf6, Tf7	Imediata	-	1	1
DC monitor				
Tdc3_rep, Tdc3_rep-des	Temporizada (exp)	*Single-server*	-	-
Tdc3_des, Tdc3_falhar	Temporizada (exp)	*Infinite-server*	-	-
Tdc3-web_rep, Tdc3-nt_rep	Temporizada (exp)	*Single-server*	-	-
Tdc3-web_falhar, Tdc3-nt_falhar	Temporizada (exp)	*Infinity-server*	-	-
TchkDC1, TconfigDC2, TreativarDC1	Temporizada (exp)	*Single-server*	-	-
Trelogio	Temporizada (det)	*Single-server*	-	-
Tf9, Tf10, Tdmf, TDC1f,TDC1o	Imediata	-	1	1
Tr1, Tr2	Imediata	-	1	0
TdetectarFalha	Imediata	-	1	1

Caso um desastre ou falha impeça a execução dos sistemas no DC primário, a guarda G_{df} é satisfeita e a transição *TdetectarFalha* é disparada. Assim, o *token* é consumido do lugar *PDC1ativo* e um *token* é gerado no lugar *PesperaDC2*. Quando os componentes do DC secundário necessários para executar os sistemas estiverem operacionais (ex.: conexões de rede e servidor SW2), a guarda G_{chkDC2} é satisfeita e a transição *TconfigDC2* habilitada. O disparo da transição *TconfigDC2* indica que a configuração do DC secundário para processar a carga de trabalho foi finalizada. Assim, um *token* é consumido do lugar *PesperaDC2* e um *token* é gerado no lugar *PDC2ativo*. O DC secundário permanece processando a carga de trabalho até que os componentes do DC primário sejam recuperados. Quando tais componentes são reparados, a guarda $G_{reativarDC1}$ habilita a transição *TreativarDC1*. Dessa forma, quando a transição *TreativarDC1* dispara, um *token* é consumido do lugar *PDC2ativo* e um *token* é gerado no lugar *PDC1ativo*, simbolizando que o DC primário está ativo e processando a carga de trabalho novamente. As características das transições e das funções de guarda usadas nas DSPNs são apresentadas nas Tabelas 15 e 16, respectivamente.

Quando uma configuração *active-standby* é adotada para a estratégia de replicação de ambiente, a ativação de componentes da infraestrutura secundária pode também ser

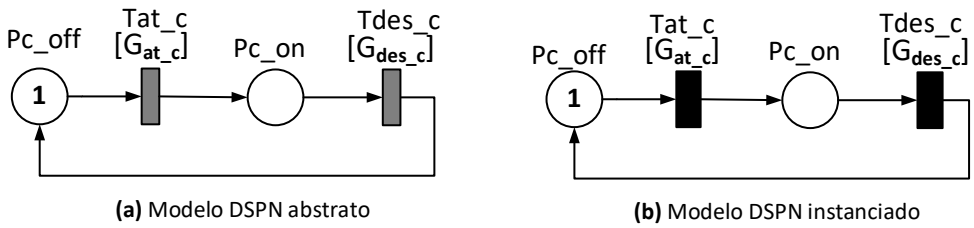
Tabela 16 – Funções de guarda das DSPNs presentes no exemplo exibido na Figura 15.

Guarda	Descrição/Condição	Função de habilitação
DC primário		
G_{f2}	Verifica a ocorrência de desastre no DC primário	*#Pdc1_des = 1*
G_{f3},	Verifica se o DC primário está inoperante	*#Pdc1_op = 0*
G_{dc1-nt_rep}, $G_{dc1-web\_rep}$	Verifica se o DC primário está operacional	*#Pdc1_op = 1*
DC secundário		
G_{f6}	Verifica a ocorrência de desastre no DC secundário	#Pdc2_des = 1
G_{f7},	Verifica se o DC secundário está inoperante	*#Pdc2_op = 0*
G_{dc2-nt_rep}, $G_{dc2-web\_rep}$	Verifica se o DC secundário está operacional	*#Pdc2_op = 1*
DC monitor		
G_{f10}	Verifica a ocorrência de desastre no DC monitor	*#Pdc3_des = 1*
G_{f9}	Verifica se o DC monitor está inoperante	*#Pdc3_op = 0*
G_{dc3-nt_rep}, G_{dc3-dm_rep}	Verifica se o DC monitor está operacional	*#Pdc3_op = 1*
$G_{relogio}$	Verifica se o monitor MD1 está operacional	*#Pdc3-dm_op = 1*
G_{dmf}	Verifica se o monitor MD1 está inoperante	*#Pdc3-dm_op = 0*
G_{DC1f}	Verifica se os componentes do DC primário falharam	*(#Pdc1-nt_op = 0) OR (#Pdc1-web_op = 0)*
G_{DC1o}	Verifica se os componentes do DC primário estão operacionais	*(#Pdc1-nt_op > 0) AND (#Pdc1-web_op > 0)*
G_{r2}	Verifica se o DC primário está operacional e ativo	*#PDC1ativo = 1*
G_{df}	Verifica se o monitor MD1 encontrou alguma falha nos componentes do DC primário	*#PDC1f = 1*
G_{chkDC2}	Verifica se os componentes do DC secundário estão operacionais	*(#Pdc2-nt_op > 0) AND (#Pdc2-web_op > 0)*
$G_{reativarDC1}$	Verifica se o monitor MD1 identificou que os componentes do DC primário voltaram ao estado operacional	*#PDC1o = 1*

realizada de forma voluntária ou ainda programada, e não apenas quando um desastre ou falha ocorre. Esses tipos de ativação dos componentes da infraestrutura secundária é bastante útil, principalmente quando a configuração *warm-standby* é empregada. No entanto, vale ressaltar que a adoção ou não desse processo depende da configuração do sistema que se deseja representar. Assim, além dos modelos DSPNs para o monitoramento e ativação dos DCs mostrados na Figura 15, também foi desenvolvido um modelo DSPN para a ativação/desativação programada de componentes da infraestrutura secundária.

A Figura 16 (a) apresenta o modelo DSPN abstrato proposto para representar a ativação/desativação programada de componentes da infraestrutura secundária. Esse modelo é chamado de modelo abstrato por possuir transições temporizadas (transições

Figura 16 – Modelo DSPN abstrato para a ativação/desativação programada de componentes da infraestrutura secundária (a) e um exemplo de uso do modelo (b).



Fonte: Criado pelo autor.

cinzas) capazes de adotar diferentes tipos de distribuição (ex.: Exponencial, Determinística ou Erlang). Tal modelo é composto por dois lugares e duas transições genéricas que possuem funções de guardas atribuídas à elas. A presença de um *token* lugar *Pc_off* indica que o componente da infraestrutura secundária deve ser desativado, enquanto a presença de um *token* lugar *Pc_on* indica que o componente deve ser ativado. As funções de guarda G_{at_c} e G_{des_c} determinam as condições para realizar a ativação (disparo da transição *Tat_c*) e desativação (disparo da transição *Tdes_c*) do(s) componente(s) da infraestrutura secundária.

A Figura 16 (b) apresenta um exemplo de instanciação modelo do modelo abstrato. A diferença entre os modelos apresentados nas Figura 16 (a) e (b) se refere à possibilidade de se trabalhar com determinadas distribuições e definir diferentes comportamentos para as funções de guarda. No exemplo mostrado, foi adotado a distribuição determinística para as transições *Tat_c* e *Tdes_c* para representar o comportamento de ativação e desativação por tempo pré-definido. Por fim, vale destacar que caso seja necessário o uso dessa DSPN, algum componente das infraestruturas consideradas (ex.: o monitor de desastres MD1) deve ser o responsável pela execução da tarefa representada pelo modelo. Isso implica numa condição de dependência (que pode ser determinada nas funções de guarda) entre o componente executor (ex.: o monitor de desastres MD1) e o processo representado pela DSPN. Por exemplo, se a ativação/desativação programada de componentes da infraestrutura secundária é executada pelo monitor de desastres MD1, este precisa estar operacional para executar tais tarefas.

5.1.4.1 Obtenção das Métricas

Esta subseção descreve como são obtidas as métricas de disponibilidade, indisponibilidade e RTO utilizando os modelos DSPNs propostos para a estratégia de replicação de ambiente. Como nessa estratégia são utilizados mais de um DC, para manter os dados entre os DCs sincronizados é necessário utilizar uma segunda estratégia (ex.: replicação de BD ou *backup*). Dessa forma, o RPO pode ser obtido de maneira similar as descritas anteriormente nas Subseções 5.1.2.1 (para o *backup*) e 5.1.3.1 (para a replicação de BD). Ainda assim,

vale ressaltar que pode ser preciso ser alterar as métricas a depender do cenário avaliado.

Considerando como exemplo o cenário apresentado na Figura 15, nenhum tipo de sincronização de dados é aplicado. Mesmo considerando as configurações *hot-standby* e *warm-standby* as métricas de disponibilidade, indisponibilidade e RTO podem ser obtidas de maneira similar. Para essa estratégia, a disponibilidade é dada pela probabilidade de se ter os elementos de rede e o servidor web (SW1) operacionais no DC primário, quanto este DC está ativado para receber as requisições dos usuários; ou os elementos de rede e o servidor web (SW2) operacionais no DC secundário, quando este DC está ativado para receber as requisições dos usuários. Dessa forma, a disponibilidade para a estratégia de replicação de ambiente pode ser obtida através da Equação 5.18. Já a indisponibilidade, é calculada utilizando a Equação 2.6.

$$\begin{aligned} \text{A} = \text{P}\{((\#\text{Pdc1-nt\_op}{>}0)AND(\#\text{Pdc1-web\_op}{>}0)AND(\#\text{PDC1ativo}{=}1)) \\ OR((\#\text{Pdc2-nt\_op}{>}0)AND(\#\text{Pdc2-web\_op}{>}0)AND(\#\text{PDC2ativo}{=}1))\} \end{aligned} \quad (5.18)$$

O RTO para esta estratégia é definido pelo tempo máximo entre: ($\mathbf{Tmp_{ativacao}}$) o tempo decorrido desde a identificação de uma falha em um componente do DC primário ou evento de desastre no mesmo, até o DC secundário estar ativado e capaz de processar as requisições dos usuários; e ($\mathbf{Tmp_{aplicacao}}$) o tempo necessário para aplicar os dados salvos do DC primário. Como no exemplo apresentado na Figura 15 não é utilizado nenhum tipo de sincronização de dados, o RTO = $\mathbf{Tmp_{ativacao}}$. A Equação 5.19 apresenta como obter o valor para tal métrica, onde a variável DM_a corresponde ao valor do *delay* atribuído a transição *TconfigDC2*.

$$\begin{aligned} \text{RTO} = \text{Tmp}_{\text{ativacao}} = P\{\#PesperaDC2{>}0\}/(P\{(\#Pdc2\text{-}nt\_op{>}0)AND \\ (\#Pdc2\text{-}web\_op{>}0)/DM_a) \end{aligned} \quad (5.19)$$

5.2 MODELOS DE CUSTO

Esta seção apresenta os modelos de custos formulados para estimar os custos das infraestruturas computacionais utilizando as estratégias de DR. Ambientes computacionais necessitam de controle e gerenciamento de custos. Assim, modelos de custos foram adotados para considerar os gastos relativos a operação das infraestruturas computacionais e os gastos com a manutenção dos equipamentos, que são despesas necessárias para o funcionamento de um negócio ou sistema. Além disso, os modelos também consideram os gastos referentes a perda de produtividade de funcionários e a perda de vendas devido a indisponibilidade dos serviços. Os modelos de custo são baseados em expressões matemáticas, onde consideramos três tipos de custos diferentes: Custos de Oportunidade (COP), Custos de Manutenção (CM) e custos de uma *Privately Owned Infrastructure* (PoI). Os

COP referem-se ao valor que uma empresa pode perder em vendas e produtividade dos empregados devido as interrupções na infraestrutura de TIC. Os CM estimam os valores de suporte para a realização de reparos na infraestrutura. Já os custos de PoI são adotados para estimar os valores para operar infraestruturas privadas e são subdivididos em Custos de Aquisição (CA) e Custos de Operação (CO). Os CA estimam o valor de aquisição de hardware e software para compor uma infraestrutura computacional, enquanto os CO estimam os valores a serem gastos com o consumo elétrico da(s) infraestrutura(s). Dessa forma, o custo total é dado pela Equação 5.20:

$$\text{Custo Total} = COP + CM + \sum_{i=1}^{n} PoI_i, \tag{5.20}$$

onde i representa a quantidade de infraestruturas existentes ou a serem implantadas (ex.: DCs).

Para estimar os COP, foi adotada a calculadora de custos da Eaton (EATON, 2018). Essa calculadora subdivide o custo de oportunidade em: perda de receita em vendas, do inglês *Lost Sales Revenue* (LSR) e perda na produtividade dos funcionários, do inglês *Lost Employee Productivity* (LEP). A Equação 5.21 mostra como são calculados os COP pela calculadora da Eaton. Para estimar esses custos, a calculadora considera diferentes parâmetros, como: o segmento da empresa, receita anual, número de funcionários, a porcentagem de funcionários afetados pelo tempo de inatividade da infraestrutura de TIC, quantidade de dias e horas trabalhadas pelos funcionários, tamanho (em ft^2) da infraestrutura e a localização da infraestrutura. Dessa forma, baseado nos parâmetros de entrada fornecidos, a calculadora fornece os valores de LSR, LEP e, consequentemente, do COP para a infraestrutura avaliada.

$$COP = LSR + LEP \tag{5.21}$$

Os CM visam estimar o valor gasto com manutenções corretivas ao ambiente adotado. Dessa forma, é considerado que o CM está diretamente atrelado a indisponibilidade do ambiente adotado. A Equação 5.22 é usada para estimar os CM:

$$CM = D_s \times C_{sp} \times T, \tag{5.22}$$

onde D_s representa a indisponibilidade da infraestrutura, C_{sp} define o valor médio de suporte por hora de indisponibilidade da infraestrutura (ex.: em dólar) e T representa o tempo de observação em horas.

Para calcular o PoI das infraestruturas presentes nos ambientes avaliados, é utilizada a Equação 5.23.

$$PoI_i = CA_i + CO_i, \tag{5.23}$$

onde CA_i e CO_i representam, respectivamente, os custos de aquisição e operação de cada infraestrutura (ex.: DC) do ambiente analisado. O custo de aquisição inclui o valor (ex.:

em dólar) dos elementos de hardware (ex.:, servidor, *racks*, roteadores e *switches*), e o valor de aquisição de produtos de software (ex.: SOs). A Equação 5.24 mostra como estimar os custos de aquisição:

$$CA_i = \sum_{a=0}^{n} Chw_i + \sum_{b=0}^{m} Csw_i, \tag{5.24}$$

onde $\sum_{a=0}^{n} Chw_i$ representa o somatório dos valores referente à aquisição de equipamentos de hardware e $\sum_{b=0}^{m} Csw_i$ representa o somatório dos valores referente à aquisição de softwares.

Por último, a Equação 5.25, baseada em Dantas et al. (2015), ilustra como são calculados os CO:

$$CO_i = C_{en} \times \sum_{n=1}^{M} PI_n \times (A_i + \alpha(1 - A_i)) \times T, \tag{5.25}$$

onde C_{en} é o valor médio da eletricidade por quilowatt-hora (kWh). PI_n representa a potência de entrada em quilowatts de cada máquina física. M define o número de máquinas físicas usadas na infraestrutura privada. A_i representa a disponibilidade de cada infraestrutura i. A variável α é um fator que determina a fração de energia elétrica que continua a ser consumida mesmo após a falha da infraestrutura i. Por fim, T é o tempo de observação a ser considerado e é dado em horas.

5.3 CONSIDERAÇÕES FINAIS

Este capítulo apresentou os modelos DSPNs e de custo propostos nesta tese para avaliar e selecionar estratégias de DR. Esse conjunto de modelos tem por objetivo facilitar o processo de avaliação de infraestruturas computacionais que adotam ou visam a implementação de estratégias de DR. Os modelos apresentados aqui servem como base para composição e análise de diferentes tipos e configurações de infraestruturas computacionais e estratégias de DR. Especificamente, os modelos DSPNs são aplicáveis para analisar infraestruturas computacionais adotando o *backup*, a replicação de BD e a replicação de ambiente como estratégias de DR. Este conjunto de modelos pode auxiliar empresas, profissionais ou pessoas interessadas no processo de avaliação dessas estratégias e no processo de tomada de decisão.

6 ESTUDOS DE CASO

Neste capítulo, são apresentados estudos de caso realizados para demonstrar o uso da metodologia e do conjunto de modelos propostos. Os modelos propostos no Capítulo 5 são empregados para compor e analisar diferentes estratégias de DR aplicadas à infraestruturas computacionais. No primeiro estudo de caso, é analisado uma infraestrutura computacional sem nenhuma estratégia de DR. Depois, no segundo estudo de caso, é analisado o *backup* como uma estratégia de DR. Em seguida, a estratégia de replicação de banco de dados é avaliada. Posteriormente, é considerado o uso da estratégia de replicação de ambientes. Por fim, é demonstrado o uso da análise multicritério para auxiliar na seleção e no ranqueamento das estratégias de DR.

6.1 AVALIAÇÃO DE UMA INFRAESTRUTURA COMPUTACIONAL SEM O USO DE ESTRATÉGIAS DE DR

Neste primeiro estudo de caso, é utilizado uma infraestrutura computacional que não adota nenhuma estratégia de DR. Foi escolhido um ambiente comumente usado em pequenas e médias empresas (ANDRADE et al., 2017). A seguir, são detalhados o cenário, os modelos DSPNs, processo de avaliação dos modelos e a execução dos experimentos deste cenário, de acordo com a metodologia proposta neste trabalho.

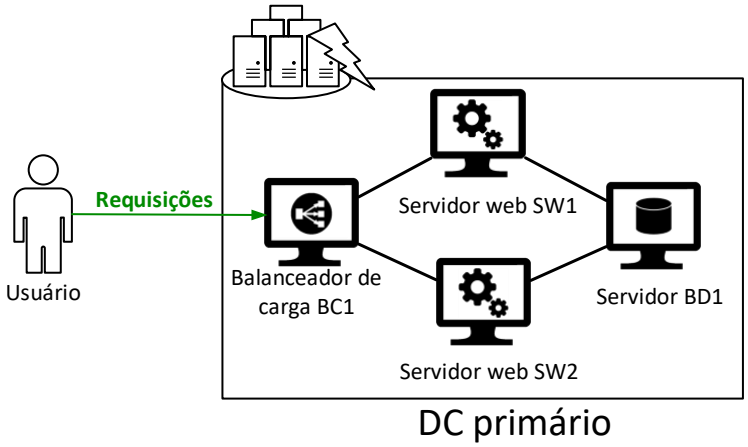
6.1.1 Cenário adotado

A infraestrutura computacional adotada é composta por um DC primário que oferece um serviço *web* através de uma aplicação web (ex.: ferramenta de comércio eletrônico). Para oferecer tal serviço, considera-se que o DC primário hospeda quatro VMs. Duas VMs são usadas como servidores web, onde ambas hospedam a mesma aplicação (redundância dos servidores web). A aplicação hospedada no servidor web usa um servidor de BD localizado em outra VM para armazenar e consultar dados. Há também uma outra VM que é um balanceador de carga. Este balanceador de carga é responsável por distribuir as requisições dos usuários entre os dois servidores web, a fim de evitar sobrecarga de uma única VM. Vale ainda ressaltar que a infraestrutura computacional do DC primário está suscetível à ocorrência de desastres.

A Figura 17 representa a infraestrutura computacional do DC primário. Nela são representados os elementos de cada uma das VMs: balanceador de carga BC1, servidores web SW1 e SW2 e o servidor do BD BD1. As linhas conectando as VMs representam os elementos de rede (ex.: roteadores, *switches*) necessários para conexão de rede entre as VMs e também para a conexão com a Internet. Caso alguma falha ou evento de desastre ocorra no DC primário, os elementos nele hospedados ficam indisponíveis, impossibilitando

o processamento das requisições dos usuários. Assim, a infraestrutura adotada está apta a processar as requisições dos usuários quando: o balanceador de carga BC1, ao menos um dos servidores de aplicação SW1 ou SW2, o servidor BD1 e os elementos de rede estiverem operando normalmente.

Figura 17 – Infraestrutura adotada para o DC primário.



Fonte: Criado pelo autor.

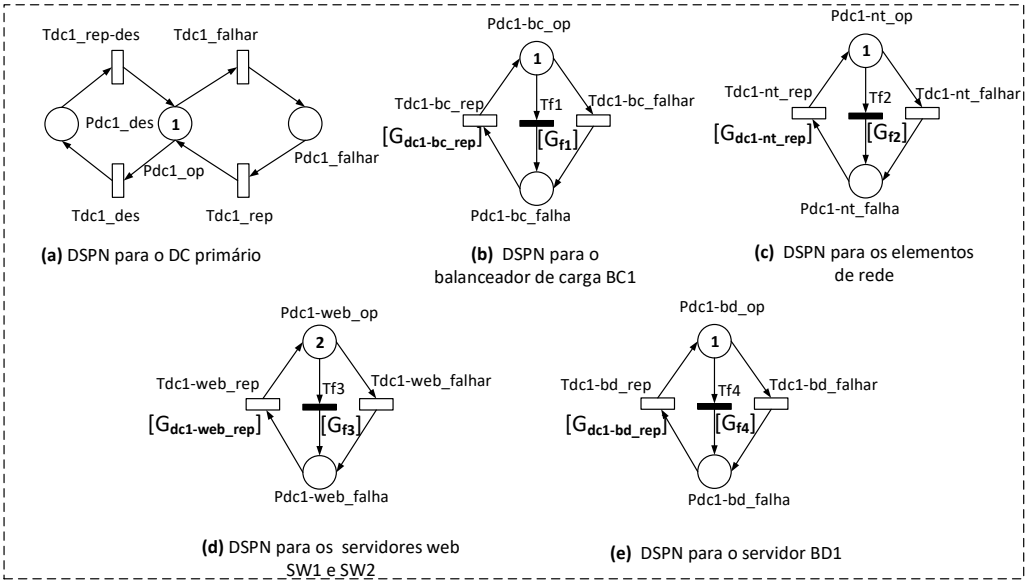
6.1.2 Modelos DSPNs

A Figura 18 apresenta todas as DSPNs usadas para modelar e avaliar o cenário deste primeiro estudo de caso. As Figuras 18 (a) e (b) exibem as DSPNs para o DC primário e o balanceador de carga BC1, respectivamente. A DSPN para os elementos de rede do DC primário é apresentada na Figura 18 (c). Em seguida, a Figura 18 (d) apresenta a DSPN para os servidores web SW1 e SW2. Esse é um componente redundante, assim, os dois *tokens* presentes no lugar *Pdc1-web_op* representam o nível de redundância do servidor web. Por fim, a DSPN mostrada na Figura 18 (d) representa o servidor de BD BD1. Para representar a dependência das VMs em relação à infraestrutura física do DC primário são utilizadas transições imediatas e funções de guarda. Ou seja, quando ocorre uma falha transiente ou um evento de desastre na infraestrutura do DC primário, as funções de guarda G_{f1}, G_{f2}, G_{f3}, G_{f4}, G_{dc1-bc_rep}, G_{dc1-nt_rep}, $G_{dc1-web\_rep}$, G_{dc1-bd_rep} são habilitadas para disparar, deixando os componentes inoperantes. A Tabela 17 resume as funções de guardas utilizadas nos modelos, enquanto a Tabela 18 detalha as características das transições utilizadas nas DSPNs.

Tabela 17 – Funções de guarda utilizadas nos modelos DSPNs da Figura 18.

Guarda	**Função de habilitação**
G_{f1}, G_{f3}, G_{f4}	*(#Pdc1-op = 0)*
G_{f2}	*(#Pdc1_des = 1)*
G_{dc1-bc_rep}, G_{dc1-nt_rep}, $G_{dc1-web\_rep}$, G_{dc1-bd_rep}	*(#Pdc1-op = 1)*

Figura 18 – Modelos DSPNs correspondentes ao DC primário exibido na Figura 17.



Fonte: Criado pelo autor.

Tabela 18 – Descrição das transições dos modelos DSPNs da Figura 18.

Transições	**Tipo**	**Semântica**	**Peso**	**Prioridade**
Tdc1_des, Tdc1_falhar	Temporizada (exp)	*Single-server*	-	-
Tdc1_rep, Tdc1_rep-des	Temporizada (exp)	*Single-server*	-	-
Tdc1-web_rep, Tdc1-bd_rep	Temporizada (exp)	*Single-server*	-	-
Tdc1-bc_rep ,Tdc1-nt_rep	Temporizada (exp)	*Single-server*	-	-
Tdc1-bd_falhar, Tdc1-web_falhar	Temporizada (exp)	*Infinite-server*	-	-
Tdc1-bc_falhar, Tdc1-nt_falhar	Temporizada (exp)	*Infinite-server*	-	-
Tf1, Tf2, Tf3, Tf4	Imediata	-	1	1

6.1.3 Resultados dos modelos DSPNs

Nesta subseção são apresentados os resultados dos modelos DSPNs da Figura 18. Os modelos DSPNs foram analisados usando a ferramenta Mercury (SILVA et al., 2015) através da análise estacionária. A Tabela 19 apresenta os parâmetros de entrada utilizados para tempo de falhas (MTTF) e reparo (MTTR) dos componentes do DC primário, tendo seus valores baseados nos trabalhos de Kim, Machida e Trivedi (2009), Machida et al. (2011), Andrade et al. (2017). Já a Tabela 20 exibe os parâmetros de entrada para estimar o custo do DC primário, tendo seus valores baseados nos trabalhos de Dantas et al. (2015) e Andrade e Nogueira (2019).

Utilizando os parâmetros de entrada, foram calculadas as métricas de disponibilidade, indisponibilidade e custo para a infraestrutura adotada neste estudo de caso. As métricas de RTO e RPO não são computadas para este cenário, uma vez que nenhuma estratégia de DR foi utilizada. Assim, a métrica de disponibilidade é obtida pela probabilidade de se

Tabela 19 – Parâmetros de entrada para as DSPNs da Figura 18.

Parâmetro	**Descrição**	**Transição associada**	**Valor (h)**
DC_{tf}	MTTF DC (transiente)	*Tdc1_falhar*	8760
DC_{tr}	MTTR DC (transiente)	*Tdc1_rep*	4
DC_{d}	Tempo médio para desastre	*Tdc1_des*	17520
DC_{rd}	Tempo médio para reparar desastre	*Tdc1_rep-des*	24
DC_{lbf}	MTTF balanceador de carga	*Tdc1-bc_falhar*	8760
DC_{lbr}	MTTR balanceador de carga	*Tdc1-bc_rep*	0,5
DC_{vmf}	MTTF servidor Web/BD	*Tdc1-web_falhar*, *Tdc1-bd_falhar*	2654
DC_{vmr}	MTTR servidor Web/BD	*Tdc1-web_rep*, *Tdc1-bd_rep*	1,25
N_f	MTTF elementos de rede	*Tdc1-nt_falhar*	10000
N_r	MTTR elementos de rede	*Tdc1-nt_rep*	1

Tabela 20 – Parâmetros de custos utilizados.

Parâmetro	**Descrição**	**Valor**
COP		
-	Segmento da indústria	*E-commerce*
-	Receita anual	US$ 1.000.000,00
-	Número de funcionários	30
-	Dias trabalhados por ano	240 dias
-	Horas trabalhadas por dia	8 horas
-	% de funcionários afetados (pela indisponibilidade)	70 %
-	Área do DC	328,084 ft^2
-	Localização DC	Califórnia
LSR	Perda de receita em vendas	US$ 114
LEP	Perda em produtividade dos funcionários	US$ 365
COP	Custo de oportunidade por hora	US$ 479
PoI		
C_{hw}	Servidor (unidade)	US$ 1.424,88
C_{sw}	Software	US$ 0
C_{en}	Custo médio do kWh	US$ 0,165
PI	Potência de entrada da máquina	0,134 (kWh)
α	Fator do consumo de energia	0,2
T	Tempo de observação	8760 horas
CM		
C_{sp}	Custo de suporte por hora	US$ 150

ter o balanceador de carga BC1, ao menos um servidor web (SW1 ou SW2), o servidor BD1 e os elementos de rede do DC primário ativos. Por outro lado, a indisponibilidade é calculada utilizando a Equação 2.6. Por fim, o custo do ambiente considerado é obtido utilizando o modelo de custo apresentado na Seção 5.2 (ver Equação 5.20). A Tabela 21

Tabela 21 – Expressões utilizadas para cálculo das métricas dos modelos DSPNs da Figura 18 no *Mercury*.

Métrica	**Expressão**
Disponibilidade	P{ *(#Pdc1-bc_op = 1) AND (#Pdc1-web_op > 0) AND (#Pdc1-nt_op > 0) AND (#Pdc1-bd_op = 1)* }
Indisponibilidade	1 - (P{ *(#Pdc1-bc_op = 1) AND (#Pdc1-web_op > 0) AND (#Pdc1-nt_op > 0) AND (#Pdc1-bd_op = 1)* })

exibe as expressões adotadas para calcular a disponibilidade e a indisponibilidade dos modelos DSPNs da Figura 18.

A Tabela 22 exibe os resultados obtidos através da realização de análise estacionária dos modelos DSPNs. Na tabela são apresentados os resultados para as métricas de disponibilidade, indisponibilidade e custo. O resultado da indisponibilidade é em horas por ano, o qual é obtido utilizando a Equação 2.7. Para a infraestrutura adotada, a disponibilidade alcançada foi de 0,99720455, enquanto a indisponibilidade por ano foi de 24,488142 horas. O valor de indisponibilidade obtido pode ser explicado devido à presença de componentes não redundantes no ambiente. Por exemplo, a infraestrutura conta apenas com um balanceador de carga e um servidor de BD. Assim, caso um desses dois componentes falhe, a infraestrutura não conseguirá processar as requisições dos usuários.

Tabela 22 – Resultados das métricas nas DSPNs da Figura 18.

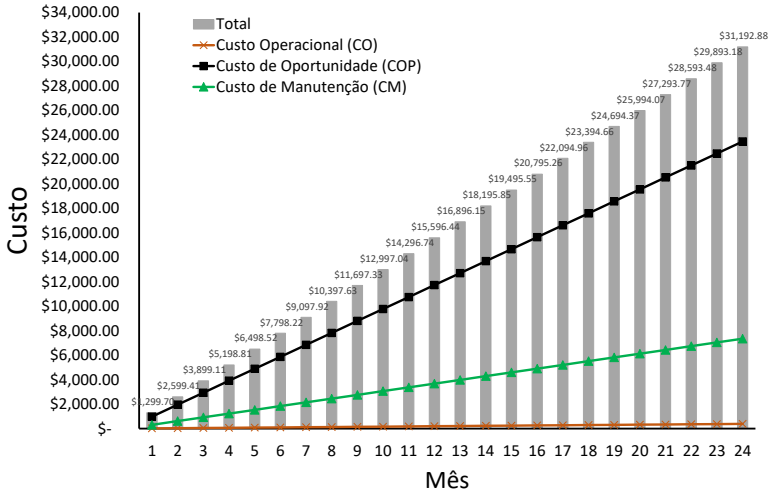
Métrica	**Resultado**
Disponibilidade	0,99720455
Indisponibilidade/ano	24,488142 horas
Custo primeiro ano	US$ 15.596,44

Baseado nos modelos de custo descritos na Seção 5.2, o custo estimado para o primeiro ano do DC primário foi de US$ 15.596,44. Esta estimativa de custos não levou em conta os CA, uma vez que consideramos que a infraestrutura do DC primário já tinha sido adquirida. Além disso, para o parâmetro de disponibilidade A_i da Equação 5.25, foi considerada apenas a disponibilidade do ambiente físico, visto que essa equação refere-se aos CO de consumo de energia elétrica. Assim, foi adotada a disponibilidade da máquina física do DC primário. Isso significa que foi calculado a probabilidade do modelo DSPN da Figura 18 (a) estar em estado operacional ($P\{\#Pdc1\_op > 0\}$). Dessa forma, o resultado para p parâmetro A_i foi de 0,99817684. A Equação B.1 detalha o cálculo do custo para o primeiro ano do DC primário.

Para um melhor detalhamento do custo, analisamos ainda os custos para dois anos de operação da infraestrutura adotada. A Figura 19 apresenta um gráfico com o custo estimado para os dois primeiros anos de operação. O gráfico exibe os COP, CM, CO e o

custo total do DC primário. Considerando esses tipos de custos, o custo total de dois anos de operação chega a US$ 31.192,88. Devido ao alto tempo de indisponibilidade, os COP são elevados (US$ 23.459,64 ao final de dois anos), representando uma perda significativa em vendas e produtividade. Esse tipo de custo é o maior responsável por este alto custo operacional. Ele representa cerca de 75% do custo total, enquanto os custos de manutenção e operacionais representam cerca de 23% e 2%, respectivamente.

Figura 19 – Custo acumulado para dois anos de operação do DC primário sem estratégias de DR.



Fonte: Criado pelo autor.

6.1.4 Experimentos para a infraestrutura sem estratégias de DR

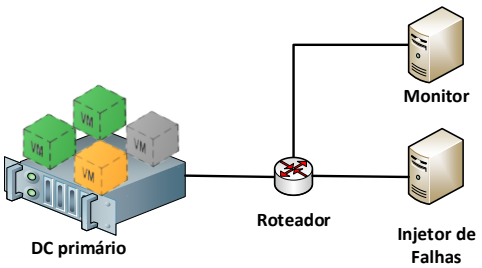
Com o objetivo de verificar os resultados obtidos pelas DSPNs, foram realizados experimentos de injeção de falhas em um ambiente real. Este tipo de experimento consiste na injeção de falhas e reparos artificiais na infraestrutura computacional adotada a fim de analisar as métricas de interesse (SILVA, 2016; MATOS; ANDRADE; MACIEL, 2014). Dessa forma, experimentos foram realizados em uma infraestrutura computacional sem DR, que foi configurada em laboratório para verificar a acurácia dos modelos apresentados anteriormente. Nas seções subsequentes são detalhadas a arquitetura utilizada, a configuração dos experimentos e os resultados.

6.1.4.1 Arquitetura de testes

Como apresentado anteriormente, o DC primário é composto por quatro VMs (ver Figura 17). Assim, no ambiente de testes, também foram adotadas quatro VMs. A Figura 20 ilustra o ambiente de testes utilizado. A VM do balanceador de carga utilizou o software HAProxy (HAPROXY, 2017) para distribuir as requisições entre os servidores SW1 e SW2. Para os servidores web, foram utilizadas duas VMs executando o Apache v2 (APACHE, 2017) hospedando uma aplicação web de *e-commerce*. Para o servidor BD1, foi utilizado o MySQL versão 5.6 (ORACLE, 2017) que armazenava dados da aplicação. É importante ressaltar que todas as VMs foram configuradas com memória *Random-access Memory*

(RAM) de 512 *Megabytes* (MBs), um v-CPU, 20 *Gigabytes* (GBs) de HD virtual e CentOS 7 (64 bits) como SO. Além disso, essas VMs foram hospedadas em uma máquina física com memória RAM de 4 GB, processador Intel Core i7-3770 3.40GHz, HD de 500 GB e CentOS 7 (64 bits) como SO, representando a infraestrutura física do DC primário.

Figura 20 – Ambiente de testes utilizado para experimentos no DC primário sem estratégias de DR.



Fonte: Criado pelo autor.

Uma ferramenta baseada no Eucabomber (SOUZA et al., 2013) e implementada em Matos, Andrade e Maciel (2014) foi usada para gerar falhas e reparos artificiais na infraestrutura adotada. Essa ferramenta foi instalada em uma máquina física (que aqui denominamos *injetor de falhas*), porém no mesmo segmento de rede. A máquina física do injetor de falhas foi configurada com memória RAM de 4 GB, processador Intel Core i5 650 3.20GHz, HD de 500 GB, e CentOS 7 (64 bits) como SO. Para representar eventos de falha e reparo nos componentes (VMs e máquina física) foram utilizados *scripts* escritos na linguagem de programação *shell script* (FLYNT; LAKSHMAN; TUSHAR, 2017). As falhas foram injetadas nos componentes em diferentes níveis (ex.: rede, SO, VM, e aplicação), de modo a simular falhas reais. Os scripts utilizados são apresentados no Apêndice D.

Para verificar se o DC primário era capaz de responder às requisições dos usuários, foi utilizado um *monitor*. O monitor foi configurado em uma máquina física e possuía um processador Intel Core i5-650, 4 GB de RAM, HD de 500 GB, e CentOS 7 (64 bits) como SO. Ele verificava a acessibilidade de cada VM do DC primário usando o comando *ping* (MANPAGEZ, 2019). Vale ressaltar que o *timeout* do *ping* foi ajustado para um segundo. Para cada componente, o monitor gravava "operacional" em um arquivo de *log* se a solicitação do *ping* fosse respondida; caso contrário, o monitor registrava o estado "falha" para o componente correspondente. Para conectar os componentes no mesmo segmento de rede, foi utilizado um roteador genérico com interfaces de 10/100 *Megabits* por segundo (Mbps). Por fim, o tempo de simulação dos experimentos representou dois anos de operação da infraestrutura adotada. Durante esse período, eventos de falha e reparo foram gerados com intervalos de tempo exponencialmente distribuídos, utilizando os mesmos valores adotados para análise das DSPNs (ver Tabela 19). No entanto, foi usado um fator de redução no valor 100 para os parâmetros de entrada dos componentes. O uso do fator de redução é uma técnica utilizada para que os experimentos sejam executados de forma mais rápida (MATOS; ANDRADE; MACIEL, 2014; ANDRADE, 2014).

6.1.4.2 Resultados dos experimentos

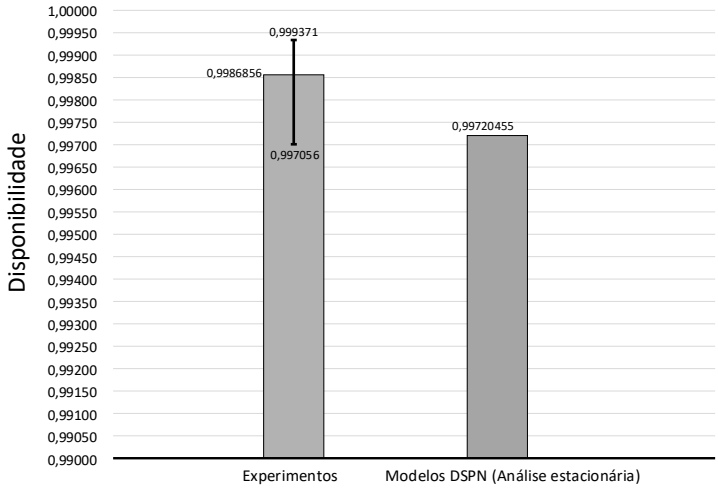
Para calcular a disponibilidade do ambiente obtida através dos experimentos, foram utilizados os arquivos de *log* de monitoramento. A Tabela 23 apresenta os resultados dos experimentos, onde a disponibilidade alcançada foi de 0,9986856. Considerando um Intervalo de Confiança (IC) de 95%, o valor da disponibilidade alcançada varia entre 0,997056 e 0,999371. O IC da disponibilidade foi calculado usando o método proposto por Keesee (1965). Além disso, a indisponibilidade total da infraestrutura durante os experimentos foi de 0,2302 horas. Como os valores dos parâmetros dos componentes foram reduzidos em um fator de 100, os resultados indicam que em 2 anos de operação a infraestrutura mostrou uma indisponibilidade de 23,02 horas, ou seja, uma média de 11,51 horas/ano.

Tabela 23 – Resultado dos experimentos de injeção de falhas.

Métrica	**Resultado**
Disponibilidade	99,86856%
IC para disponibilidade	[0,997056 - 0,999371]
Indisponibilidade/ano	11,51 horas

Por último, a Figura 21 apresenta uma comparação dos resultados obtidos através dos experimentos de injeção de falhas e dos modelos DSPNs. No gráfico, é possível notar que o resultado da análise estacionária dos modelos DSPNs está dentro do IC dos resultados dos experimentos. Dessa forma, não pode-se rejeitar a hipótese de que o modelo representa o comportamento do sistema real.

Figura 21 – Comparação entre os resultados obtidos nos experimentos e modelos DSPNs.



Fonte: Criado pelo autor.

6.2 AVALIAÇÃO DE UMA INFRAESTRUTURA COMPUTACIONAL UTILIZANDO *BACKUP* COMO ESTRATÉGIA DE DR

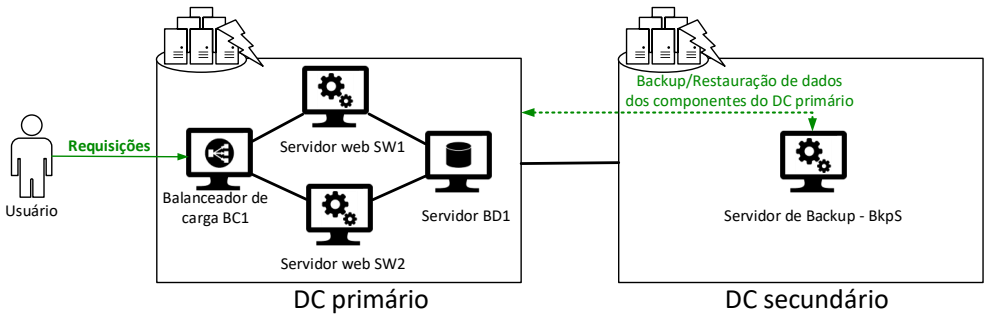
Nesta seção, iremos apresentar um estudo de caso onde será analisada uma infraestrutura computacional que adota a estratégia de *backup* para fins de DR. Primeiro, é descrito o cenário adotado no estudo de caso. Em seguida, são apresentados os modelos DSPNs criados para avaliar o cenário descrito. Posteriormente, são apresentados os resultados da avaliação dos modelos, bem como os experimentos conduzidos na infraestrutura adotada.

6.2.1 Cenário adotado

A infraestrutura adotada neste segundo estudo de caso consiste basicamente de uma extensão da infraestrutura apresentada anteriormente (ver Seção 6.1), de modo que o *backup* é usado como estratégia de DR. Dessa forma, a configuração e funcionamento do DC primário permaneceram as mesmas. Isso significa que para processar as requisições dos usuários é necessário ter o balanceador de carga BC1, ao menos um dos servidores web (SW1 ou SW2), o servidor BD1 e os elementos de rede operando normalmente. Adicionalmente, neste estudo de caso, o DC primário utiliza um serviço de *backup* e de restauração de dados. O uso de tal serviço tem como objetivo manter os dados seguros, atualizados e organizados; assim, os mesmos podem ser recuperados caso desastres afetem o DC primário.

A infraestrutura que provê o serviço de *backup* está geograficamente distante do DC primário, onde aqui denominaremos de DC secundário. O serviço de *backup* executado no DC secundário realiza periodicamente os *backups* de dados de todas as VMs hospedadas no DC primário. Assim, em casos de desastres no DC primário, é realizada a restauração do último *backup* válido, restaurando os dados para as respectivas VMs, quando o DC primário voltar à operacionalidade. Dessa forma, após a restauração das VMs do DC primário, o sistema pode voltar a processar as requisições dos usuários normalmente. Para coordenar os serviços de *backup* e de restauração dos dados, uma VM é usada no DC secundário como servidor de *backup* (BkpS). A Figura 22 mostra os elementos do cenário adotado neste estudo de caso e suas relações.

Figura 22 – Infraestrutura adotada utilizando o *backup* como estratégia de DR.

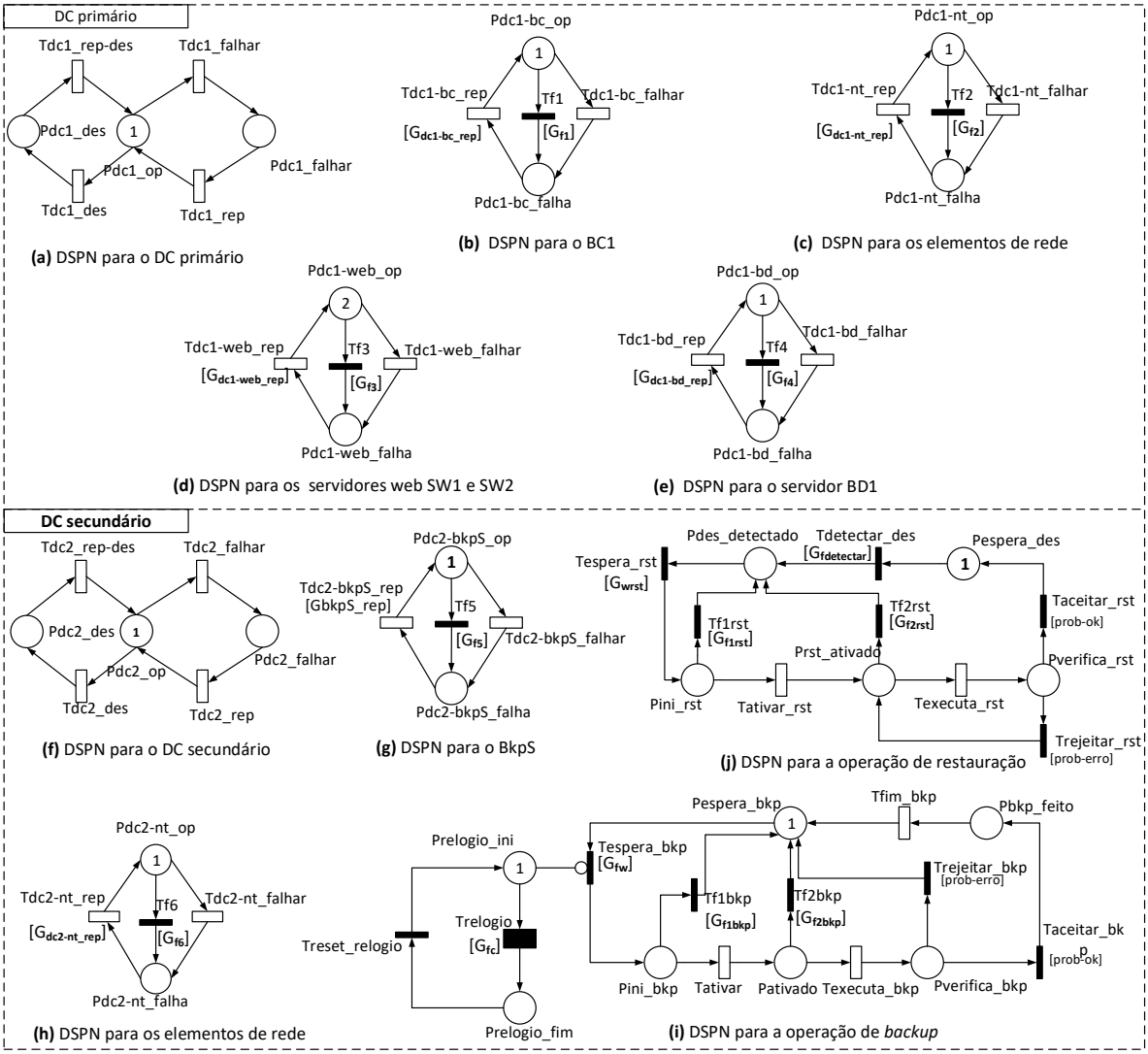


Fonte: Criado pelo autor.

6.2.2 Modelos DSPNs

Esta subseção descreve os modelos DSPNs utilizados para o cenário adotado. A Figura 23 exibe todas as DSPNs desenvolvidas para representação e avaliação do cenário. As DSPNs representam cada componente do DC primário e do DC secundário. Na parte superior da Figura 23 estão os modelos DSPNs para os componentes do DC primário. Esses modelos DSPNs permaneceram inalterados em relação ao primeiro estudo de caso (ver Subseção 6.1.2). Já na parte inferior, estão localizados os modelos DSPNs para os componentes do DC secundário e as operações de *backup* e restauração de dados. A Figura 23 (f) exibe a DSPN para o DC secundário. As Figuras 23 (g) e (h) apresentam as DSPNs para o servidor de *backup* BkpS e elementos de rede, respectivamente. Por fim, as Figuras 23 (i) e (j) mostram os modelos que representam, respectivamente, as operações de *backup* e restauração de dados.

Figura 23 – Modelos DSPNs que representam a infraestrutura apresentada na Figura 22.



Fonte: Criado pelo autor.

Tabela 24 – Funções de guarda utilizadas nos modelos DSPNs da Figura 23.

Guarda	**Função de Habilitação**
DC Primário	
G_{f1}, G_{f3}, G_{f4}	*(#Pdc1-op = 0)*
G_{f2}	*(#Pdc1_des = 1)*
G_{dc1-bc_rep}, G_{dc1-nt_rep}, $G_{dc1-web\_rep}$, G_{dc1-bd_rep}	*(#Pdc1-op = 1)*
DC secundário	
G_{fc}	(*#Pdc2-bkpS_op* = 1)
$G_{fdetectar}$	(*#Pdc1_des = 1*)
G_{f5}	(*#Pdc2_op = 0*)
G_{f6}	(*#Pdc2_des = 1*)
G_{bkpS_rep}, G_{dc2-nt_rep}	(*#Pdc2_op = 1*)
G_{fw}, G_{wrst}	(*#Pdc2-bkpS_op=1*) AND (*#Pdc2-nt_op=1*) AND (*#Pdc1-nt_op=1*) AND (*#Pdc1_op=1*) AND (*#Pdc1-bc_op >0*) AND (*#Pdc1-web_op >0*) AND (*#Pdc1-bd_op > 0)*
G_{f1bkp}, G_{f2bkp}, G_{f1rst}, G_{f2rst}	(*#Pdc2-bkpS_op=0*) OR (*#Pdc2-nt_op=0*) OR (*#Pdc1-nt_op=0*) OR (*#Pdc1_op = 0*) OR (*#Pdc1-bc_op=0*) OR (*#Pdc1-web_op=0*) OR (*#Pdc1-bd_op = 0*)

A periodicidade da operação de *backup* é definida pelo *delay* associado à transição *Trelogio*. Quando a guarda G_{fc} é satisfeita e *Trelogio* dispara, a transição *Tespera_bkp* pode ser disparada caso a guarda G_{fw} seja satisfeita, dando início à operação de *backup*. Caso nenhum dos componentes do DC primário falhe, habilitando as funções de G_{f1bkp} e G_{f2bkp}, o *backup* é realizado (disparo da transição *Texecuta_bkp*). Por fim, o *backup* é aceito e finalizado com sucesso caso a transição *Taceitar_bkp* seja disparada. O disparo dessa transição é baseado na probabilidade do *backup* ser válido, a qual é fornecida como parâmetro de entrada para o peso da transição *Taceitar_bkp*.

A operação de restauração é realizada automaticamente apenas em casos de desastres no DC primário. Quando um desastre ocorre no DC primário, a função de guarda $G_{fdetectar}$ habilita a transição *Tdetectar_des*. No entanto, o processo de restauração apenas inicia quando a transição *Tespera_rst* é disparada. Isso ocorre quando a função de guarda G_{wrst} é satisfeita, o que ocorre apenas quando todos os componentes do DC primário voltam a estar operacionais. A partir desse momento, caso não ocorram mais falhas nos componentes do DC primário, habilitando as funções de guarda G_{f1rst} e G_{f2rst}, a restauração dos dados é realizada. Por fim, a restauração é aceita e finalizada com sucesso caso a transição *Taceitar_rst* seja disparada. Caso contrário, o processo de restauração é reiniciado.

Por fim, a Tabela 24 exibe as funções de guarda utilizadas nas DSPNs da Figura 23, enquanto a Tabela 25 lista as características das transições utilizadas nos modelos DSPNs da Figura 23. Vale destacar nessa última tabela, a adoção de uma transição do tipo

determinística (*Trelogio*), a qual é indicada pelo tipo *Temporizada (det)*. Além disso, os pesos associados às transições *Taceitar_bkp*, *Taceitar_rst*, *Trejeitar_bkp*, e *Trejeitar_rst* são representados por parâmetros que são definidos posteriormente na Tabela 26.

Tabela 25 – Descrição das transições das DSPNs da Figura 23.

Transições	**Tipo**	**Semântica**	**Peso**	**Prioridade**
DC Primário				
Tdc1_des, Tdc1_falhar	Temporizada (exp)	*Infinite-server*	-	-
Tdc1_rep, Tdc1_rep-des	Temporizada (exp)	*Single-server*	-	-
Tdc1-web_rep, Tdc1-bd_rep	Temporizada (exp)	*Single-server*	-	-
Tdc1-bc_rep , Tdc1-nt_rep	Temporizada (exp)	*Single-server*	-	-
Tdc1-bd_falhar, Tdc1-web_falhar	Temporizada (exp)	*Infinite-server*	-	-
Tdc1-bc_falhar, Tdc1-nt_falhar	Temporizada (exp)	*Infinite-server*	-	-
Tf1, Tf2, Tf3, Tf4	Imediata	-	1	1
DC secundário				
Tdc2_des, Tdc2_falhar	Temporizada (exp)	*Infinite-server*	-	-
Tdc2_rep, Tdc2_rep-des	Temporizada (exp)	*Single-server*	-	-
Tdc2-bkpS_rep, Tdc2-nt_rep	Temporizada (exp)	*Single-server*	-	-
Tativar, Texecuta_bkp, Tfim_bkp	Temporizada (exp)	*Single-server*	-	-
Tativar_rst, Texecuta_rst	Temporizada (exp)	*Single-server*	-	-
Trelogio	Temporizada (det)	*Single-server*	-	-
Tdc2-bkpS_falhar, Tdc2-nt_falhar	Temporizada (exp)	*Infinite-server*	-	-
Treset_relogio	Imediata	-	1	0
Tf5, Tf6, Tespera_bkp	Imediata	-	1	1
Tf1bkp, Tf2bkp, Tf1rst, Tf2rst	Imediata	-	1	1
Taceitar_bkp, *Taceitar_rst*	Imediata	-	B_{sp}	1
Trejeitar_bkp, *Trejeitar_rst*	Imediata	-	B_{ep}	1
Tdetectar_des, Tespera_rst	Imediata	-	1	1

6.2.3 Resultados dos modelos DSPNs

Nesta subseção, são apresentados os resultados obtidos através da análise dos modelos DSPNs da Figura 23. Os modelos DSPNs foram analisados usando o Mercury (SILVA et al., 2015). Diferentemente do primeiro estudo de caso, devido à necessidade de modelagem utilizando uma transição determinística, foi realizada a simulação estacionária dos modelos DSPNs ao invés da análise estacionária. A Tabela 26 apresenta os parâmetros de entrada utilizados para os componentes do DC primário e secundário. Esses valores foram baseados nos trabalhos de Kim, Machida e Trivedi (2009), Machida et al. (2011), Andrade et al. (2017), Mendonça et al. (2019b). Vale destacar ainda que os parâmetros B_{bt}, B_{rt}, B_{sp} e B_{ep} foram obtidos através de experimentos em um ambiente real (ver Subseção 6.2.5).

Conforme apresentado anteriormente, a Tabela 20 (ver Subseção 6.1.3) exibe os parâmetros de entrada para estimar os custos.

Tabela 26 – Parâmetros de entrada para as DSPNs da Figura 23.

Parâmetro	**Descrição**	**Transição associada**	**Valor (h)**
DC primário			
DC_{tf}	MTTF DC (transiente)	*Tdc1_falhar*	8760
DC_{tr}	MTTR DC (transiente)	*Tdc1_rep*	4
DC_{d}	Tempo médio para desastre	*Tdc1_des*	17520
DC_{rd}	Tempo médio para reparar desastre	*Tdc1_rep-des*	12
DC_{lbf}	MTTF balanceador de carga	*Tdc1-bc_falhar*	8760
DC_{lbr}	MTTR balanceador de carga	*Tdc1-bc_rep*	0,5
DC_{vmf}	MTTF servidor Web/BD	*Tdc1-web_falhar*, *Tdc1-bd_falhar*	2654
DC_{vmr}	MTTR servidor Web/BD	*Tdc1-web_rep*, *Tdc1-bd_rep*	1,25
Rede			
N_f	MTTF elementos de rede	*Tdc1-nt_falhar*, *Tdc2-nt_falhar*	10000
N_r	MTTR elementos de rede	*Tdc1-nt_rep*, *Tdc2-nt_rep*	1
DC secundário			
B_{tf}	MTTF DC (transiente)	*Tdc2_falhar*	8760
B_{tr}	MTTR DC (transiente)	*Tdc2_rep*	4
B_{d}	Tempo médio para desastre	*Tdc2_des*	17520
B_{rd}	Tempo médio para reparar desastre	*Tdc2_rep-dis*	12
B_{tfs}	MTTF servidor de *Backup*	*Tdc2-bkpS_falhar*	8760
B_{trs}	MTTR servidor de Backup	*Tdc2-bkpS_rep*	4
B_{abr}	(Des)ativar *backup*/restauração	*Tativar, Tativar_rst, Tfim_bkp*	0,015
B_{bt}	Tempo médio de *Backup* por GB	-	0,0304
B_{rt}	Tempo médio de restauração por GB	-	0,0253
B_{s}	Tamanho do *backup*	-	100 GB
B_{i}	Intervalo do *backup*	*Trelogio*	24
B_{bte}	Execução de operação de *backup*	*Texecuta_bkp*	$B_{bt} \times B_s$
B_{rte}	Execução de operação de restauração	*Texecuta_rst*	$B_{rt} \times B_s$
B_{ep}	Prob. de erro para *backup*/restauração	*Trejeitar_bkp*, *Trejeitar_rst*	0,015 (%)
B_{sp}	Prob. de sucesso para *backup*/restauração	*Taceitar_bkp*, *Taceitar_rst*	0,985 (%)

Para este segundo estudo de caso, as seguintes métricas foram calculadas: disponibilidade, indisponibilidade, RTO, RPO e custo. A Tabela 27 exibe as expressões usadas para obter as métricas. As métricas obtidas neste estudo de caso são baseadas nas métricas definidas na Subseção 5.1.2.1. A disponibilidade é obtida pela probabilidade de se ter o balanceador de carga BC1, ao menos um servidor web, o servidor BD1, e os elementos de rede do DC primário operacionais. Além disso, é necessário que a infraestrutura não esteja

realizando uma operação de recuperação de dados. O RTO é calculado pelo tempo médio que o sistema leva para voltar ao estado operacional (com os dados recuperados) após um desastre. O RPO é obtido pelo tempo médio decorrido desde o último *backup* até o evento de desastre. Já o custo é calculado usando-se o modelo de custo apresentado na Seção 5.2.

Tabela 27 – Expressões utilizadas nas DSPNs da Figura 23 para cálculo das métricas.

Métrica	**Expressão**
Disponibilidade	$P\{ (\#Pdc1\text{-}bc\_op > 0)\ AND\ (\#Pdc1\text{-}web\_op > 0)\ AND\ (\#Pdc1\text{-}bd\_op > 0)\ AND\ (\#Pdc1\text{-}nt\_op > 0)\ AND\ (\#Pespera\_des = 1)\ \}$
Indisponibilidade	$1 - (P\{ (\#Pdc1\text{-}bc\_op > 0)\ AND\ (\#Pdc1\text{-}web\_op > 0)\ AND\ (\#Pdc1\text{-}bd\_op > 0)\ AND\ (\#Pdc1\text{-}nt\_op > 0)\ AND\ (\#Pespera\_des = 1)\ \})$
RTO	$((E\{\#Pdes\_detectado\})+(E\{\#Pini\_rst\})+(E\{\#Prst\_ativado\}) + (E\{\#Pverifica\_rst\})) \ / \ (E\{\#Pini\_rst\} \times (1/B_{abr}))$
RPO	$((E\{\#Pini\_bkp\})+(E\{\#Pativado\})+(E\{\#Pverifica\_bkp\}) +(E\{\#Pbkp\_feito\})) \ / \ (E\{\#Pini\_bkp\} \times (1/B_{abr}))$

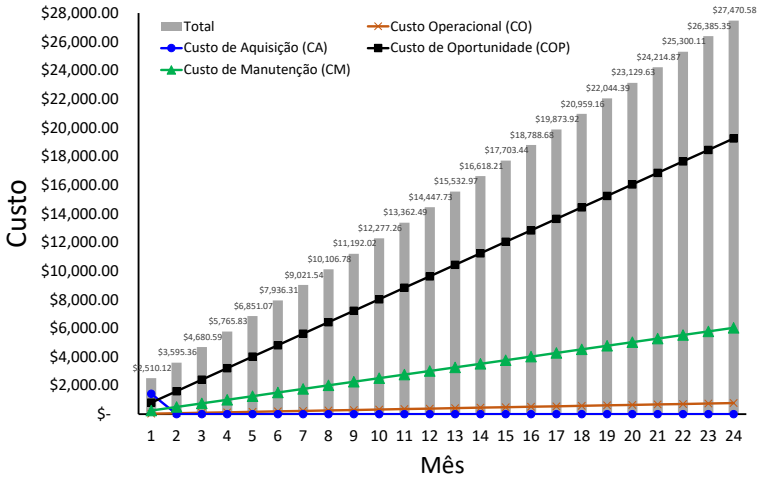
A Tabela 28 exibe os resultados obtidos através da simulação estacionária dos modelos DSPNs. Ela apresenta os resultados obtidos para as métricas de disponibilidade, indisponibilidade, indisponibilidade/ano, RPO, e RTO, considerando um IC de 95% e margem de erro de 5%. Além disso, também é apresentada a estimativa de custo para o primeiro ano de operação da infraestrutura utilizando a estratégia de *backup*. A disponibilidade média alcançada para a infraestrutura avaliada no cenário foi de 0,99770674, o que representa uma indisponibilidade média anual de 20,088957 horas. Considerando o IC, o valor da disponibilidade variou entre 0,99770638 e 0,99770710, de modo que representa apenas uma pequena variação de alguns segundos na indisponibilidade do DC primário. O RPO apresentou uma média de 24,107 horas, o que significa que, em média, podem ser perdidos 24,107 horas de dados inseridos nas VMs. Já para o RTO, foi obtida uma média de 16,928 horas, representando o tempo médio que levaria para o DC primário operar normalmente após um desastre.

Tabela 28 – Resultados das métricas avaliadas utilizando as DSPNs da Figura 23.

Métrica	**Resultado**
Disponibilidade	0,99770674 [0,99770638 - 0,99770710]
Indisponibilidade/ano	20,088957 horas [20,08580 - 20,092111]
RPO	24,107 horas [24,106 - 24.108]
RTO	16,928 horas [16,910 - 16,946]
Custo anual	US$ 14.447,73

O custo estimado para o primeiro ano de operação da infraestrutura utilizando a estratégia de *backup* foi de US$ 14.447,73. Essa estimativa contempla os CA de uma máquina física para infraestrutura do DC secundário. Além disso, também são considerados os COP, CM e CO. Vale destacar que para calcular o CO do DC secundário foi utilizado a disponibilidade da máquina física (0,99885974) para o parâmetro A_i da Equação 5.25. A Seção B.2 detalha o cálculo do custo para o primeiro ano operação considerando o DC primário e secundário.

Figura 24 – Custo acumulado para dois anos de operação do DC primário utilizando a estratégia de *backup*.



Fonte: Criado pelo autor.

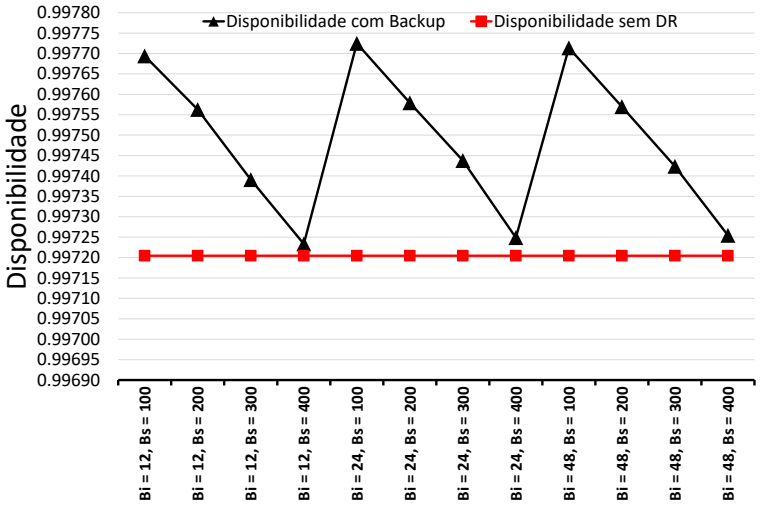
Também foi analisada a estimativa dos custos para dois anos de operação do DC primário utilizando a estratégia de *backup*. A Figura 24 apresenta um gráfico com o custo estimado de dois anos de operação, contemplando os COP, CA, CO, CM e o custo total. Considerando esses tipos de custos, o custo total para dois anos de operação chega ao máximo de US$ 27.470,58. Os custos de oportunidade, de aproximadamente US$ 19.245,20 ainda são os que mais impactam no custo total, cerca de 70%. Tal custo é seguido pelos custos de manutenção (US$ 6,026.69), representando cerca de 22% do custo total. Os custos de aquisição (US$ 1.424,88) e operacionais (US$ 773,82) são os que menos impactam, representando, respectivamente, cerca de 5% e 3% do custo total.

6.2.4 Análise multi-cenário utilizando os modelos DSPNs

Após analisar o cenário base do estudo de caso, cenários distintos foram analisados para demonstrar a flexibilidade dos modelos e encontrar configurações que poderiam apresentar os melhores resultados. Foram avaliados 12 cenários distintos variando o intervalo da operação de *backup* (B_i = 12, 24 e 48 horas) e o tamanho do *backup* (B_s = 100, 200, 300 e 400 GB). A Tabela 54 (ver Apêndice C) apresenta o resultado de cada um desses cenários, considerando as métricas de disponibilidade, indisponibilidade, RTO, RPO, e custo. A Figura 25 exibe um gráfico comparando os resultados da disponibilidade dos

12 cenários analisados. Os melhores resultados para a disponibilidade foram os cenários onde o tamanho dos dados para *backup* e restauração eram menores. Além disso, através do gráfico é possível notar que em todos os cenários testados, a disponibilidade foi mais alta quando a estratégia de *backup* foi utilizada. No entanto, à medida que a quantidade de dados à serem salvos e/ou restaurados crescem, a disponibilidade tende a diminuir. Dessa forma, um sistema precisa de uma grande quantidade de dados críticos para ser recuperada, a disponibilidade desse sistema tende a ser mais baixa. Mesmo assim, ainda vale ressaltar que com a adoção da estratégia de *backup*, os dados da organização estariam menos sujeitos a uma perda total.

Figura 25 – Análise da disponibilidade considerando diferentes cenários.

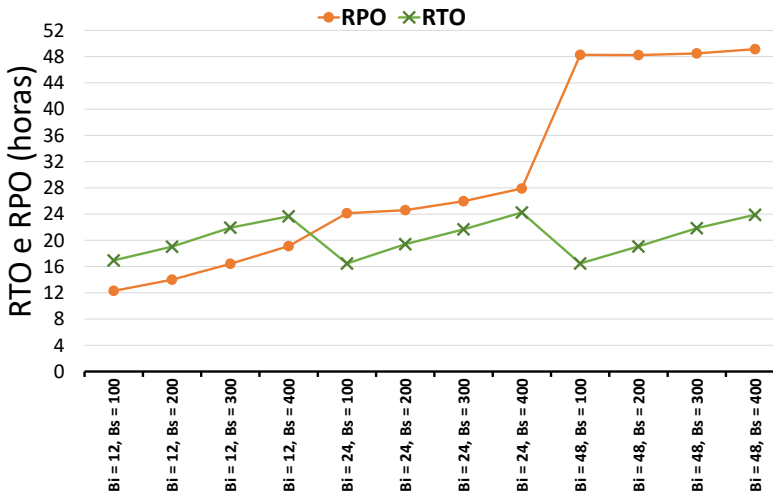


Fonte: Criado pelo autor.

A Figura 26 também apresenta uma análise de cenários distintos para as métricas de RTO e RPO. Os melhores resultados para o RTO acontecem quando o tamanho do *backup* (B_s) é menor (ex.: 100 GB). Por outro lado, os melhores valores para o RPO ocorrem quando o intervalo de *backup* (B_i) é curto (12 horas), mas também quando o sistema tem uma pequena quantidade de dados para *backup*/restauração. Em resumo, o RTO degrada à medida que a quantidade de dados para *backup*/restauração aumenta, enquanto o RPO degrada quando o intervalo de *backup* e a quantidade de dados para *backup*/restauração aumentam.

Por fim, foram analisados os custos dos diferentes cenários considerados. A Figura 27 mostra um gráfico com as estimativas de custo para o primeiro ano de operação de cada cenário analisado. Analisando o gráfico, é possível notar que os custos para os cenários considerados são praticamente inversos ao gráfico referente à disponibilidade (ver Figura 25). Isso ocorre, porque os COP (tipo de custo que têm mais impacto sobre o custo total para essa estratégia) estão diretamente relacionados à disponibilidade do cenário. Dessa forma, uma maior disponibilidade significará um menor custo total, bem como uma menor disponibilidade significará maior custo total. O cenário 4 ($B_i = 12$ e $B_s = 400$) foi o que

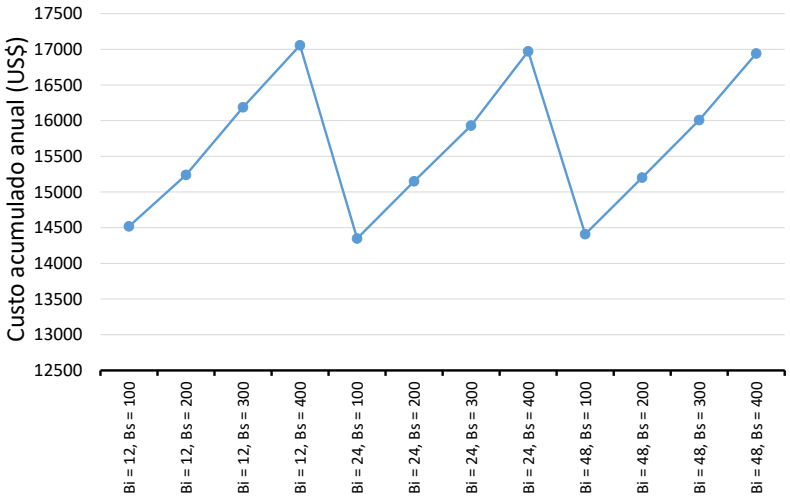
Figura 26 – Análise do RPO e do RTO considerando diferentes cenários.



Fonte: Criado pelo autor.

apresentou o maior custo dentre os cenários avaliados, com um valor de US$ 17.055,65. Por outro lado, o cenário 9 ($B_i = 48$ e $B_s = 100$) teve o menor custo dentre os cenários avaliados, com um valor de US$ 14.408,11.

Figura 27 – Análise do custo acumulado do primeiro ano de operação considerando diferentes cenários da estratégia de *backup*.



Fonte: Criado pelo autor.

6.2.5 Experimentos para o *backup* como estratégia de DR

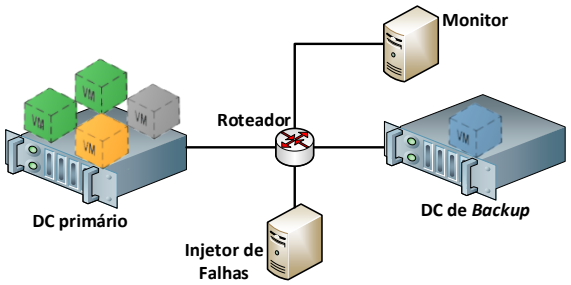
Os experimentos para este estudo de caso foram conduzidos em duas fases. Na *fase 1*, os experimentos tinham como principal objetivo observar o tempo médio necessário para completar tarefas de *backup* e restauração de dados. Os resultados obtidos a partir desses experimentos foram utilizados como parâmetros de entrada para os modelos DSPNs. Já os experimentos conduzidos na *fase 2* tinham como principal objetivo comparar os resultados

obtidos através dos experimentos com os resultados obtidos através dos modelos DSPNs. Para tal, foram executados experimentos de injeção de falhas no ambiente. A seguir, é descrito a arquitetura de testes utilizada para execução dos experimentos, bem como os resultados obtidos.

6.2.5.1 Arquitetura de testes

A Figura 28 apresenta a infraestrutura utilizada nos experimentos. A configuração da máquina física e dos componentes do DC primário é a mesma utilizada no primeiro estudo de caso (ver Subseção 6.1.4.1). A configuração das máquinas físicas do monitor e do injetor de falhas também permaneceu a mesma. A máquina física do DC secundário foi configurada com memória RAM de 4 GB, processador Intel Core i7-3770 3.40GHz, HD de 500 GB e CentOS 7 (64 bits) como SO. A VM do servidor de *backup* hospedada nessa infraestrutura, foi configurada com RAM de 1 GB, dois v-CPU, 600 GB de HD e CentOS 7 (64 bits) como SO. Para o *backup* e restauração de dados foi utilizado o software *Bacula Community* na versão 9.0.7 (Bacula Systems, 2018).

Figura 28 – Ambiente de testes utilizado para análise do *backup* como estratégia de DR.



Fonte: Criado pelo autor.

Na *fase 1*, com o objetivo de coletar o tempo médio para operações de *backup* e restauração de dados, foram executadas operações de *backup* completo e operações de restauração de dados usando conjuntos de dados de 5, 10 e 15 GB. O conjunto de arquivos utilizados para tais operações possuíam tamanhos distintos, a fim de simular um ambiente mais próximo da realidade. Assim, cada conjunto de arquivos (5, 10 e 15 GB) foi dividido da seguinte forma: 20% com arquivos de 1 MB, 20% com arquivos de 3 MB, 20% com arquivos de 10 MB, 20% com arquivos de 50 MB e 20% com arquivos de 500 MB. Esses arquivos foram gerados de forma aleatória por meio da ferramenta *data duplicator* (*dd*) (Unix Tutorial, 2019).

Na *fase 2* dos experimentos, com o objetivo de verificar os resultados obtidos através dos modelos DSPNs, foi utilizado o injetor de falhas baseada no Eucabomber (SOUZA et al., 2013) e implementada em Matos, Andrade e Maciel (2014) para gerar falhas e reparos artificiais na infraestrutura montada. Para gerar as falhas e reparos nos componentes (VMs

e máquina física) foram utilizados *scripts* escritos na linguagem de programação *shell script* (FLYNT; LAKSHMAN; TUSHAR, 2017). As falhas foram injetadas nos componentes em diferentes níveis (ex.: rede, SO, VM, e aplicação), de modo a simular falhas reais. Assim, falhas e reparos artificias foram gerados nos componentes das infraestruturas considerando os mesmos parâmetros definidos para os modelos DSPNs. Vale ressaltar que o monitor foi responsável por verificar a acessibilidade de cada VM do DC primário e de *backup*, usando o comando *ping* (MANPAGEZ, 2019). O tempo de simulação do experimento representou 5,4 anos de operação da infraestrutura adotada, visto que foi usado um fator de redução de 100 nos parâmetros de entrada dos componentes da infraestrutura. *Backups* de dados das VMs do DC primário foram realizados periodicamente, de acordo com o intervalo e quantidade de dados definidos. Além disso, quando um desastre ocorria no DC primário, era realizado uma restauração dos dados de cada VM após elas retornarem ao estado operacional.

6.2.5.2 Resultado dos experimentos

Nesta subseção, são apresentados os resultados dos experimentos realizados na arquitetura de testes. Primeiramente, são apresentados os resultados da *fase 1*. Em seguida, são discutidos os resultados dos experimentos de injeção de falhas (*fase 2*). Na *fase 1*, foram realizados experimentos para obter o tempo médio de execução das operações de *backup* e restauração de dados. A Tabela 29 exibe estes resultados para cada conjunto de dados utilizados nos experimentos. Entre parênteses é apresentado o desvio padrão de cada amostra.

Tabela 29 – Resultados da fase *1* dos experimentos para cada conjunto de dados.

Quantidade de dados	**Tempo de *Backup* (desv. pad.)**	**Tempo de restauração (desv. pad.)**
5 GB	551.231 s (31.28894)	457.184 s (2.84563)
10 GB	1107.103 s (16.17975)	913.171 s (1.84368)
15 GB	1623.111 s (35.41482)	1370.054 s (2.80873)

Depois de coletar o tempo médio das operações de *backup* e restauração, calculamos o tempo médio por GB para executar o *backup* e o tempo médio por GB para restaurar os dados. Foi escolhido adotar o tempo médio por GB para flexibilizar e tornar mais adaptável os modelos DSPNs. A Tabela 30 exibe os resultados médios. Esses resultados foram usados como parâmetros de entrada para os modelos DSPNs (ver os parâmetros B_{bt} e B_{rt} descritos na Tabela 26). Além dos valores mostrados nessa tabela, nesta fase dos experimentos foram computadas as taxas de erro e sucesso das operações de *backup* e restauração. O valor observado com os experimentos para as taxas de erro e sucesso dessas operações foi de 1,5% e 98,5%, respectivamente. Essas taxas também são utilizadas

como parâmetros de entrada para os modelos DSPNs através dos pesos B_{ep} e B_{sp} (ver Tabela 26).

Tabela 30 – Resultados consolidados dos experimentos da *fase 1*.

Descrição	**Parâmetro**	**Valor (*h*)**
Média de tempo por *GB* para *backup* de dados	B_{bt}	0,0304
Média de tempo por *GB* para restauração de dados	B_{rt}	0,0253
Porcentagem de sucesso de operações de *backup* e restauração de dados	B_{sp}	98,5%
Porcentagem de erro de operações de *backup* e restauração de dados	B_{ep}	1,5%

Na *fase 2*, escolhemos um dos cenários considerados pela análise multi-cenário (ver Subseção 6.2.4) para comparar os resultados dos modelos com resultados de um ambiente real. O cenário escolhido foi aquele onde o intervalo de *backup* foi de 24 horas ($B_i = 24$) e o tamanho dos dados de *backup* foi de 400 GB ($B_s = 400$). Os demais parâmetros utilizados para configurar o *script* de injeção de falhas dos componentes da infraestrutura foram os mesmos apresentados anteriormente na Tabela 26.

A Tabela 31 exibe os resultados obtidos através dos experimentos e modelos DSPNs. Para calcular a métrica de disponibilidade do cenário nos experimentos, foi utilizado o *log* de monitoramento. A disponibilidade média para os experimentos foi de 0,9965243. Considerando o IC de 95%, o resultado pode flutuar entre 0,9949956 e 0,9975440. Esse resultado significa uma média de indisponibilidade por ano de 30,447132 horas. Já para os modelos DSPNs, a disponibilidade média alcançada foi de 0,997248, variando entre 0,997247 e 0,997249 para um IC de 95%. Tal resultado indica uma média de indisponibilidade por ano de 24,10752 horas. Por último, visualizando esses resultados, é possível perceber que os ICs da disponibilidade obtidos através dos experimentos e modelos se sobrepõem. Dessa forma, não é possível rejeitar a hipótese de que os resultados são iguais.

Tabela 31 – Resultado dos experimentos de injeção de falhas (*fase 2*) e modelos DSPNs com os parâmetros $B_i = 24$ e $B_s = 400$.

Métricas	**Experimentos**	**Modelos DSPNs**
Disponibilidade	99,65243%	99,7248%
IC para disponibilidade	[0,9949956 - 0,9975440[	[0,997247 - 0,997249]
Indisponibilidade/ano	30,447132 horas	24,10752 horas

6.3 AVALIAÇÃO DE UMA INFRAESTRUTURA COMPUTACIONAL UTILIZANDO REPLICAÇÃO DE BANCO DE DADOS COMO ESTRATÉGIA DE DR

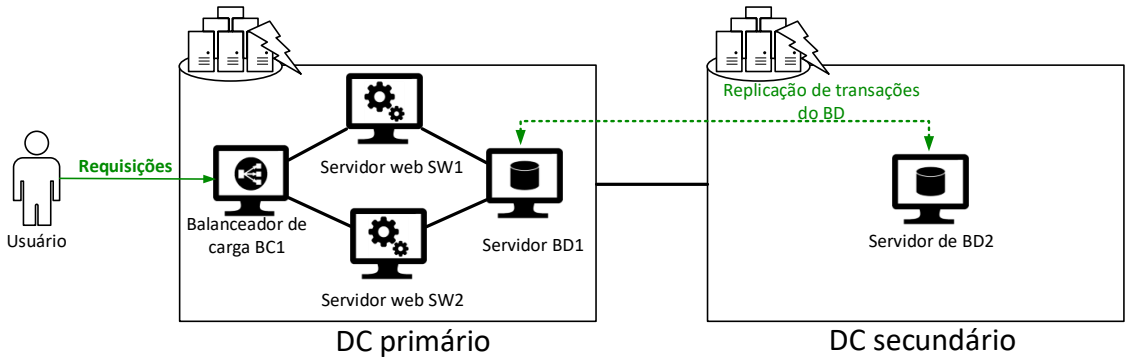
Esta seção apresenta um estudo de caso onde é analisado uma infraestrutura computacional que adota a replicação de BD como estratégia de DR. Sendo assim, essa estratégia é

aplicável especialmente a casos onde as empresas têm seus principais dados armazenados em BD. Tal solução é geralmente escolhida por pequenas ou médias empresas que almejam um baixo investimento inicial em estratégias de DR, uma vez que apenas o BD é replicado em uma infraestrutura secundária. Neste estudo de caso é considerado o uso de BD relacionais, onde são utilizados dois tipos de configurações para o processo de replicação dos dados: (*1*) Assíncrona e (*2*) Semi-síncrona. A seguir, são descritos o cenário adotado e os modelos DSPNs criados. Posteriormente, são discutidos os resultados obtidos através dos modelos DSPNs, bem como os experimentos realizados.

6.3.1 Cenário adotado

A Figura 29 apresenta o ambiente adotado. Para o DC primário, é utilizada a mesma configuração definida no primeiro estudo de caso (ver Seção 6.1). No entanto, neste estudo de caso, o DC primário utiliza um serviço de replicação de dados para o servidor BD1. Assim, no DC secundário, o servidor BD2 que é configurado em modo *hot-standby* funciona como *replica* do BD1, recebendo atualizações de dados que são inseridos no servidor BD1. Dessa forma, vale salientar que o servidor BD1 funciona como um servidor de BD *master*.

Figura 29 – Ambiente adotado para a estratégia de replicação de BDs.



Fonte: Criado pelo autor.

Nesse cenário, os componentes são configurados para funcionar da seguinte forma: quando o usuário envia requisições ao sistema, a requisição é enviada ao balanceador de carga BC1. Esse, por sua vez, decide para qual servidor Web a requisição deve ser encaminhada. Em seguida, a aplicação presente no servidor Web acessa o servidor BD1 para persistir/consultar dados no BD. Quando o servidor BD1 recebe alguma transação de persistência de dados, a mesma é replicada no servidor BD2 que, por sua vez, pode seguir duas configurações distintas: a replicação assíncrona ou a replicação semisíncrona. Vale ressaltar que essas configurações têm comportamentos diferentes (ver Subseção 2.4.2.2).

Caso alguma falha aconteça no servidor BD1, que é executado no DC primário, sua *replica*, o servidor BD2, pode assumir o lugar dele e manter o sistema operando normalmente. Geralmente, os SGBDs permitem que tal configuração seja feita de modo automático. É importante destacar que, em caso de desastres no DC primário, as requisições dos

usuários não conseguem ser processadas mesmo que o servidor BD2 ainda esteja operando normalmente no DC secundário. Isso acontece porque é necessário ter o balanceador de carga, ao menos um dos servidores web e um dos servidores de BD operacionais para que os usuários tenham suas requisições processadas corretamente. No entanto, vale ser ressaltado que a utilização de uma réplica ajuda a manter uma cópia das informações que são salvas no servidor BD1 em um local distinto, provendo redundância dos dados do sistema.

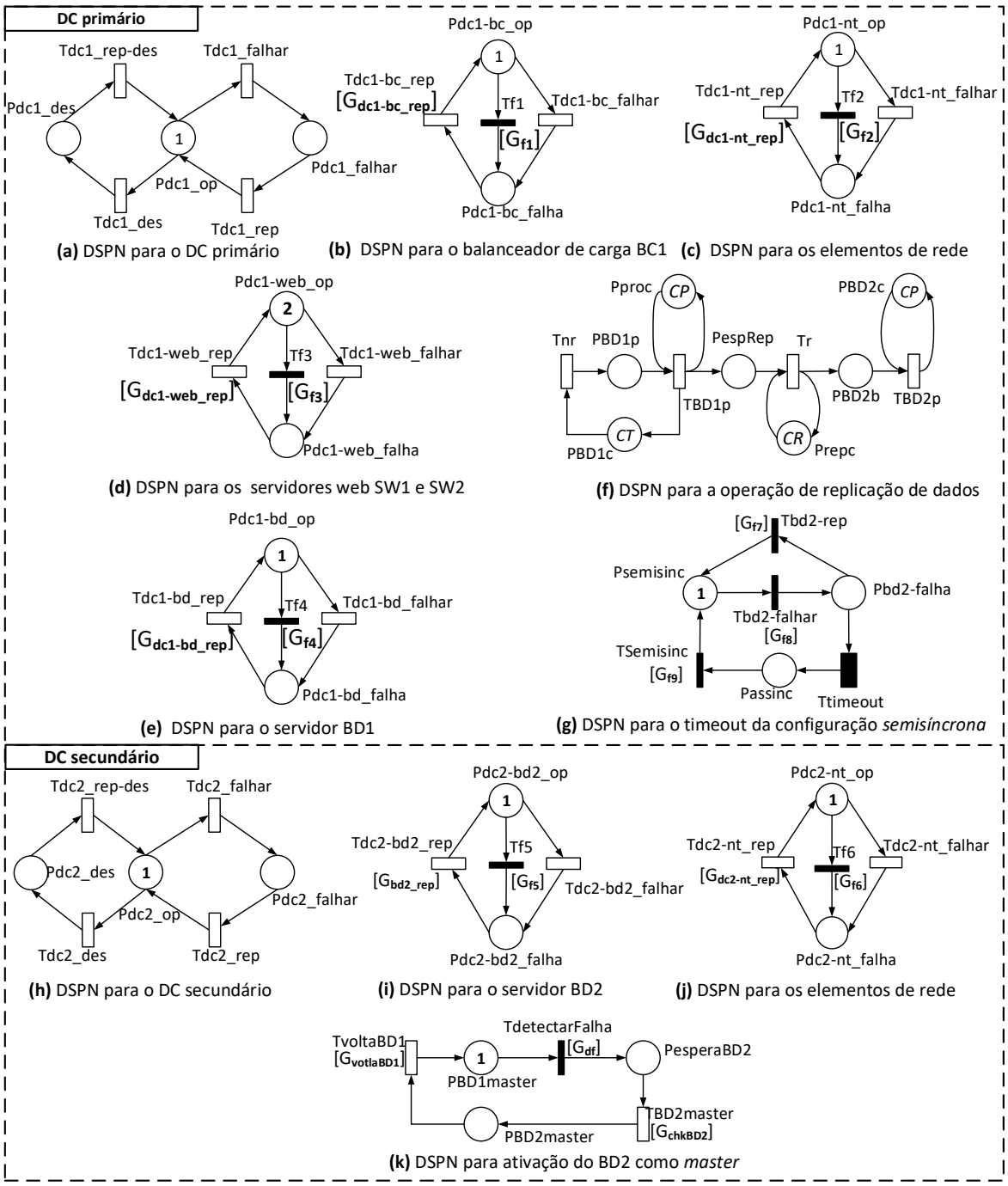
6.3.2 Modelos DSPNs

Essa subseção apresenta os modelos DSPNs criados para analisar o cenário adotado neste estudo de caso. A Figura 30 apresenta os modelos DSPNs criados para o cenário da Figura 29. Esses modelos DSPNs representam cada componente do DC primário e secundário, além dos processos de replicação para a configuração assíncrona e o *timeout* da configuração semisíncrona. Na parte superior da Figura 30 estão os modelos DSPNs para os componentes do DC primário. Os modelos DSPNs das Figuras 30 (a - e) permaneceram inalterados em relação ao primeiro estudo de caso (ver Subseção 6.1.2). No entanto, no DC primário foram incluídos os modelos DSPNs para o processo de replicação assíncrona (ver Figura 30 (f)) e para o *timeout* da configuração semisíncrona (ver Figura 30 (g)). Vale ressaltar que o modelo DSPN da Figura 30 (g) só é utilizado quando se deseja considerar a infraestrutura com a replicação semisíncrona.

Já na parte inferior da Figura 30, estão localizados os modelos DSPNs para os componentes do DC secundário. A Figura 30 (h) apresenta a DSPN para o DC secundário. As Figuras 30 (i) e (j) apresentam as DSPNs para o servidor BD2 e os elementos de rede do DC secundário, respectivamente. Já a Figura 30 (k) mostra a DSPN que representa a ativação do servidor BD2 como BD *master*, quando o servidor BD1 falhar. Quando ocorre uma falha do servidor BD1 a função de guarda G_{df} habilita o disparo da transição *TdetectarFalha*. Em seguida, para o servidor BD2 ser definido como *master* a guarda G_{chkBD2} precisa ser satisfeita para habilitar o disparo da transição *TBD2master*. Essa função de guarda verifica se os elementos de rede do DC secundário e o servidor BD2 estão operacionais. No disparo da transição *TBD2master* um *token* é gerado no lugar *PBD2master*, indicando que o servidor BD2 é o BD *master*. O servidor BD2 é mantido como *master* até o reparo do servidor BD1, onde a função de guarda $G_{voltaBD1}$ realiza essa verificação. Quando $G_{voltaBD1}$ é satisfeita, a transição *TvoltaBD1* pode ser disparada, tornando o servidor BD1 o BD *master* e servidor BD2 a *replica*.

Quando a replicação semisíncrona é considerada, o sistema necessita dos dois servidores de BD (BD1 e BD2) operacionais para processar as transações do sistema. Caso uma falha ocorra no servidor BD2, o servidor BD1 pode ser reconfigurado automaticamente para utilizar a replicação assíncrona. Isso garante que o sistema continue recebendo requisições dos usuários, pois o servidor BD1 está operacional. A DSPN da Figura 30 (g) modela

Figura 30 – Modelos DSPNs considerando a estratégia de replicação de BD.



Fonte: Criado pelo autor.

esse comportamento. Quando o servidor BD2 falha, a guarda G_{f8} habilita o disparo da transição *Tbd2-falhar*. Em seguida, a transição *Ttimeout* é habilitada. O *delay* associado à essa transição representa o tempo de *timeout* o qual o servidor BD1 aguarda o servidor BD2 voltar ao estado operacional (ex.: 60 segundos). Esse tempo pode ser configurado no SGBD. Caso o servidor BD2 volte ao estado operacional antes do disparo de *Ttimeout*, a função de guarda G_{f7} habilita o disparo da transição imediata *Tbd2-rep*, indicando que sistema ainda utiliza a replicação semisíncrona. No entanto, caso *Ttimeout* seja disparada,

o servidor BD1 passa a operar utilizando a replicação semisíncrona. A Tabela 32 exibe as funções de guarda utilizadas nas DSPNs da Figura 30, enquanto a Tabela 33 lista as características das transições utilizadas nessas DSPNs.

Tabela 32 – Funções de guarda utilizadas nos modelos DSPNs da Figura 30.

Guarda	**Função de Habilitação**
DC Primário	
G_{f1}, G_{f3}, G_{f4}	*(#Pdc1-op = 0)*
G_{f2}	*(#Pdc1_des = 1)*
G_{dc1-bc_rep}, G_{dc1-nt_rep}, $G_{dc1-web\_rep}$, G_{dc1-bd_rep}	*(#Pdc1-op = 1)*
G_{f7}, G_{f9}	(*#Pdc1-bd_op > 0*) AND ((*#Pdc2-bd2_op > 0*) AND (*#Pdc2-nt_op > 0*))
G_{f8}	(*#Pdc1-bd_op > 0*) AND ((*#Pdc2-bd2_op = 0*) OR (*#Pdc2-nt_op = 0*))
DC Secundário	
G_{f5}	(*#Pdc2_op = 0*)
G_{f6}	(*#Pdc2_des = 1*)
G_{bd2_rep}, $G_{dc2\text{-}nt\_rep}$	(*#Pdc2_op = 1*)
G_{df}	(*#Pdc1-bd_op = 0*)
G_{chkBD2}	(*#Pdc2-bd2_op > 0*) *AND* (*#Pdc2-nt_op > 0*)
$G_{voltaBD1}$	(*#Pdc1-bd_op > 0*) *AND* (*#Pdc1-nt_op > 0*)

Tabela 33 – Descrição das transições dos modelos DSPNs da Figura 30.

Transições	**Tipo**	**Semântica**	**Peso**	**Prioridade**
DC Primário				
Tdc1_des, Tdc1_falhar	Temporizada (exp)	*Infinite-server*	-	-
Tdc1_rep, Tdc1_rep-des	Temporizada (exp)	*Single-server*	-	-
Tdc1-web_rep, Tdc1-bd_rep	Temporizada (exp)	*Single-server*	-	-
Tdc1-bc_rep ,Tdc1-nt_rep, Tnr	Temporizada (exp)	*Single-server*	-	-
Tdc1-bd_falhar, Tdc1-web_falhar	Temporizada (exp)	*Infinite-server*	-	-
Tdc1-bc_falhar, Tdc1-nt_falhar	Temporizada (exp)	*Infinite-server*	-	-
TBD1p, Tr,TBD2p	Temporizada (exp)	*Infinite-server*	-	-
Ttimeout	Temporizada (det)	*Single-server*	-	-
Tf1, Tf2, Tf3, Tf4	Imediata	-	1	1
Tbd2-rep, *Tsemisinc*, *Tbd2-falhar*	Imediata	-	1	1
DC secundário				
Tdc2_des, *Tdc2_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tdc2_rep, *Tdc2_rep-des*	Temporizada (exp)	*Single-server*	-	-
Tdc2-bd2_rep, *Tdc2-nt_rep*	Temporizada (exp)	*Single-server*	-	-
TBD2master, *TvoltaBD1*	Temporizada (exp)	*Single-server*	-	-
Tdc2-bd2_falhar, *Tdc2-nt_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tf5, *Tf6*	Imediata	-	1	1
TdetectarFalha	Imediata	-	1	1

6.3.3 Resultados dos modelos DSPNs

Nesta subseção, são apresentados os resultados obtidos através dos modelos DSPNs apresentados na subseção anterior. Primeiro, a Tabela 34 apresenta os parâmetros de entrada utilizados para os componentes do DC primário (DC Prim.) e do DC secundário (DC Sec.), os quais foram baseados nos trabalhos de Kim, Machida e Trivedi (2009), Machida et al. (2011) e Andrade et al. (2017). Vale destacar ainda que os parâmetros $\boldsymbol{DCT_{p\_assinc}}$, $\boldsymbol{DCT_{p\_semisinc}}$, $\boldsymbol{DCT_{r\_assinc}}$ e $\boldsymbol{DCT_{r\_semisinc}}$ foram obtidos através de experimentos em um ambiente real, os quais são apresentados na Subseção 6.3.4. Já os parâmetros de entrada para estimar o custo são exibidos na Tabela 20, conforme apresentados anteriormente.

Tabela 34 – Parâmetros de entrada para os modelos DSPNs da Figura 30.

Parâmetro	**Descrição**	**Transição associada**	**Valor (h)**
DC Prim.			
DC_{tf}	MTTF DC (transiente)	*Tdc1_falhar*	8760
DC_{tr}	MTTR DC (transiente)	*Tdc1_rep*	4
DC_{d}	Tempo médio para desastre	*Tdc1_des*	17520
DC_{rd}	Tempo médio para reparar desastre	*Tdc1_rep-des*	12
DC_{lbf}	MTTF balanceador de carga	*Tdc1-bc_falhar*	8760
DC_{lbr}	MTTR balanceador de carga	*Tdc1-bc_rep*	0,5
DC_{vmf}	MTTF servidor Web/BD	*Tdc1-web_falhar*, *Tdc1-bd_falhar*	2654
DC_{vmr}	MTTR servidor Web/BD	*Tdc1-web_rep*, *Tdc1-bd_rep*	1,25
DCT_{nr}	Tempo médio de chegada das transações	*Tnr*	0,000138
DCT_{p_assinc}	Tempo médio p/ processar uma transação Assínc.	*TBD1p*	6,67116667E-5
DCT_{r_assinc}	Tempo médio p/ replicar uma transação Assínc.	*Tr*	7,52463889E-5
$DCT_{p\_semisinc}$	Tempo médio p/ processar uma transação Semisinc.	*TBD1p*	0,000134153
$DCT_{r\_semisinc}$	Tempo médio p/ replicar uma transação Semisinc.	*Tr*	6,6936111E-5
DC_{t}	*Timeout* da replicação semisíncrona	*Ttimeout*	0,01666
CT	Capacidade de transações no BD1	-	10
CP	Capacidade de processamento dos servidores de BD	-	10
CR	Capacidade de replicação do BD1	-	10
Rede			
N_f	MTTF elementos de rede	*Tdc1-nt_falhar*, *Tdc2-nt_falhar*	10000
N_r	MTTR elementos de rede	*Tdc1-nt_rep*, *Tdc2-nt_rep*	1
DC Sec.			
R_{tf}	MTTF DC (transiente)	*Tdc2_falhar*	8760
R_{tr}	MTTR DC (transiente)	*Tdc2_rep*	4
R_{d}	Tempo médio para desastre	*Tdc2_des*	17520
R_{rd}	Tempo médio para reparar desastre	*Tdc2_rep-dis*	12
R_{tfs}	MTTF servidor de BD (BD2)	*Tdc2-bd2_falhar*	2654
R_{trs}	MTTR servidor de BD (BD2)	*Tdc2-bd2_rep*	1,25
R_{m}	Tempo de ativação do servidor de BD como *master*	*TBD2master*, *TvoltaBD1*	0,00833

Utilizando os parâmetros de entrada, foram calculadas as métricas de disponibilidade, indisponibilidade, RTO e RPO através da análise e simulação estacionária. A análise estacionária foi utilizada para avaliar os modelos com a configuração assíncrona, enquanto a simulação estacionária foi usada para avaliar os modelos da configuração semisíncrona. A presença da transição determinística *Ttimeout* para a configuração semisíncrona, faz com que o modelo precise ser simulado.

Para esse estudo de caso foi utilizada a execução dos modelos de forma hierárquica, de modo a evitar problemas como *largness* e *stifness* (TRIVEDI et al., 2001). Dessa forma, a execução dos modelos é feita em duas etapas: (*1*) são executados os modelos DSPNs das Figuras 30 com exceção da DSPN *para a operação da replicação de dados* exibida na Figura 30 (*f*); posteriormente (*2*) a DSPN da Figura 30 (*f*) é executada separadamente. As métricas computadas neste estudo de caso são baseadas nas métricas definidas na Subseção 5.1.3.1.

Na primeira etapa, são calculadas as métricas de disponibilidade e indisponibilidade e métricas secundárias que servem como parâmetros de entrada para a segunda etapa de avaliação dos modelos DSPNs. As métricas secundárias calculadas nessa etapa são: $\mathbf{Prob_{entrada}}$, $\mathbf{Prob_{BD1op}}$, $\mathbf{Prob_{envio}}$, $\mathbf{Prob_{assinc}}$ e $\mathbf{Tmp_{ativacao}}$. Na segunda etapa da execução das DSPNs são obtidos os valores para o RPO e para a métrica secundária $\mathbf{Tmp_{aplicacao}}$ que faz parte do cálculo do RTO. Para facilitar o entendimento, a seguir é listado como as métricas são compostas para cada configuração avaliada:

- **Replicação assíncrona**:

  A disponibilidade é obtida pela probabilidade de se ter os elementos de rede do DC primário, o balanceador de carga BC1, ao menos um servidor web e o servidor BD1 operacionais; **OU** os elementos de rede do DC primário e secundário, o balanceador de carga BC1, ao menos um servidor web e o servidor BD2 operacionais, além do servidor BD2 ativado como *master*;

  O RTO é definido pelo *tempo máximo entre* o tempo decorrido desde a falha do servidor BD1 até a sua recuperação ou a ativação do servidor BD2 como *master* **E** o tempo necessário para o servidor de BD2 aplicar as transações já recebidas do servidor BD1;

  O RPO é definido pelo tempo médio que as transações levam para serem replicadas, visto que se um desastre acontecer neste intervalo de tempo, as transações que ainda não foram replicadas são desfeitas.

- **Replicação semisíncrona**:

  A disponibilidade é obtida pela probabilidade de se ter os elementos de rede dos DCs primário e secundário, o balanceador de carga BC1, ao menos um servidor web, o servidor BD1 e o servidor BD2 operacionais; **OU** os elementos de rede do

DC primário, o balanceador de carga BC1, ao menos um servidor web e o servidor BD2 operacionais, porém o servidor BD2 precisa estar ativado como *master*; **OU** os elementos de rede do DC primário, o balanceador de carga BC1, ao menos um servidor web e o servidor BD1 operacionais, além do método de replicação ter mudado para o assíncrono;

O RTO é definido da mesma forma que na configuração assíncrona;

O RPO é definido pelo tempo médio que as transações levam para serem replicadas do BD1 para o BD2 quando o método de replicação é alternado automaticamente para o assíncrono.

O custo estimado para cada cenário (replicação assíncrona e semissíncra) é calculado utilizando a Equação 5.20. Para os CA, é considerado a aquisição de uma máquina física (servidor) para representar a infraestrutura do DC secundário. A Tabela 35 apresenta as expressões usadas para obter as métricas de disponibilidade, indisponibilidade, RTO e RPO, bem como as métricas secundárias.

A Tabela 36 exibe os resultados obtidos a partir dessas métricas, além da estimativa de custo para o primeiro ano de operação da infraestrutura com a adoção da estratégia de replicação de BD. Vale ressaltar que os resultados para configuração semisíncrona foram obtidos através de simulação estacionária, seguindo um intervalo IC de 95% e margem de erro de 5%. Os valores para os ICs são apresentados entre parênteses na Tabela 36.

Os resultados mostram que as configurações para as replicações de BD são bastante similares. A disponibilidade média alcançada para a infraestrutura adotando a replicação assíncrona é de 0,99843347. Isso representa uma indisponibilidade anual média de 13,72280228 horas. Quando a configuração semisíncrona é adotada, a disponibilidade da infraestrutura é um pouco menor, de 0,99841683, o que representa uma indisponibilidade anual média de 13,8685692 horas.

Os resultados para o RPO e RTO também foram muito próximos à zero (ver Tabela 36). O RPO para a configuração semisíncrona foi de 0,00046767 segundos, enquanto para a configuração assíncrona o RPO foi maior, mas também próximo de zero com 0,27161950 segundos. Para o RTO, o valor médio obtido foi de 0,02117695 horas para a configuração assíncrona e 0,02190006 horas para a configuração semisíncrona. Os valores obtidos tanto para o RTO quanto para o RPO, em ambas as configurações analisadas, demonstram a eficácia da estratégia em manter a perda de dados dos sistemas próximo de zero.

O custo estimado para o primeiro ano de operação da infraestrutura utilizando a replicação de BD foi de US$ 10.443,54 para a configuração assíncrona e US$ 10.535,22 para a configuração semisíncrona. Vale ressaltar que essas estimativas contemplam aquisição de uma máquina física para a infraestrutura do DC secundário. Além disso, para calcular os custos de operacionais dos DCs foi utilizado valor da disponibilidade das máquinas físicas que compõem cada DCs para o parâmetro A_i da Equação 5.25. Para as duas configurações

Tabela 35 – Expressões usadas para calcular as métricas dos modelos DSPNs da Figura 30.

Métrica	Expressão
Executada na primeira etapa	
Disp (assínc.)	$P\{((\#Pdc1\text{-}nt\_op{>}0)$ AND $(\#Pdc1\text{-}bc\_op{>}0)$ AND $(\#Pdc1\text{-}web\_op{>}0)$ AND $(\#Pdc1\text{-}bd\_op{>}0)$ AND $(\#PBD1master{=}1))$ OR $((\#Pdc1\text{-}nt\_op{>}0)$ AND $(\#Pdc1\text{-}bc\_op{>}0)$ AND $(\#Pdc1\text{-}web\_op{>}0)$ AND $(\#Pdc2\text{-}nt\_op{>}0)$ AND $(\#Pdc2\text{-}bd2\_op{>}0)$ AND $(\#PBD2master{=}1))\}$
Disp (semisínc.)	$P\{((\#Pdc1\text{-}nt\_op{>}0)$ AND $(\#Pdc1\text{-}bc\_op{>}0)$ AND $(\#Pdc1\text{-}web\_op{>}0)$ AND $(\#Pdc1\text{-}bd\_op{>}0)$ AND $(\#PBD1master{=}1)$ AND $(\#Pdc2\text{-}nt\_op{>}0)$ AND $(\#Pdc2\text{-}bd2\_op{>}0)$ AND $(\#Psemisinc{=}1))$ OR $((\#Pdc1\text{-}nt\_op{>}0)$ AND $(\#Pdc1\text{-}bc\_op{>}0)$ AND $(\#Pdc1\text{-}web\_op{>}0)$ AND $(\#Pdc1\text{-}bd\_op{>}0)$ AND $(\#PBD1master{=}1)$ AND $(\#Passinc{=}1))$ OR $((\#Pdc1\text{-}nt\_op{>}0)$ AND $(\#Pdc1\text{-}bc\_op{>}0)$ AND $(\#Pdc1\text{-}web\_op{>}0)$ AND $(\#Pdc2\text{-}nt\_op{>}0)$ AND $(\#Pdc2\text{-}bd2\_op{>}0)$ AND $(\#PBD2master{=}1))\}$
Indisp (assínc.)	*1 -* ***Disp (assínc.)***
Indisp (semisínc.)	*1 -* ***Disp (semisínc.)***
$\mathbf{Prob_{assinc}}$ **(semisínc.)**	$P\{\#Passinc{>}0\}$
$\mathbf{Prob_{entrada}}$	$P\{(\#Pdc1\text{-}nt\_op{>}0)$ AND $(\#Pdc1\text{-}bc\_op{>}0)$ AND $(\#Pdc1\text{-}web\_op{>}0)$ AND $(\#Pdc1\text{-}bd\_op{>}0)$ AND $(\#PBD1master{=}1)\}$
$\mathbf{Prob_{BD1op}}$	$P\{\#Pdc1\text{-}bd\_op{>}0\}$
$\mathbf{Prob_{envio}}$	$P\{(\#Pdc1\text{-}nt\_op{>}0)$ AND $(\#Pdc2\text{-}nt\_op{>}0)$ AND $\#Pdc1\text{-}bd\_op{>}0)$ AND $(\#Pdc2\text{-}bd\_op{>}0)$ AND $(\#Pdc1\text{-}des{=}0)\}$
$\mathbf{Tmp_{ativacao}}$	$\mathrm{P}\{\#PesperaBD2{>}0\}$ / $(\mathrm{P}\{(\#Pdc2\text{-}nt\_op{>}0)$ AND $(\#Pdc2\text{-}bd\_op{>}0)$ AND $(\#Pdc1\text{-}bd\_op{=}0)$ AND $(\#PesperaBD2{>}0)\}$ / $R_m)$
Executada na segunda etapa	
$\mathbf{Tmp_{aplicacao}}$	$\mathrm{E}\{\#PBD2b\} \times (1$ / $((1$ / $\mathrm{DCT_{nr}}) \times \mathbf{Prob_{entrada}} \times \mathrm{P}\{\#PBD1c{>}0\} \times \mathbf{Prob_{BD1op}} \times \mathrm{P}\{\#Pproc{>}0\} \times \mathbf{Prob_{envio}} \times \mathrm{P}\{\#Prepc{>}0\}))$
RPO (assínc.)	$\mathrm{E}\{\#PespRep\} \times (1$ / $((1$ / $DCT_{nr}) \times \mathbf{Prob_{entrada}} \times \mathrm{P}\{\#PBD1c{>}0\} \times \mathbf{Prob_{BD1op}} \times \mathrm{P}\{\#Pproc{>}0\}))$
RPO (semisínc.)	$\mathrm{E}\{\#PespRep\} \times (1$ / $((1$ / $DCT_{nr}) \times \mathbf{Prob_{entrada}} \times \mathrm{P}\{\#PBD1c{>}0\} \times \mathbf{Prob_{BD1op}} \times \mathrm{P}\{\#Pproc{>}0\})) \times \mathbf{Prob_{assinc}}$ **(semisínc.)**

analisadas (assíncrona e semisíncrona) os valores desse parâmetro foram $A_i = 0{,}998859749$ para o DC primário e $A_i = 0{,}998859739$ para o DC secundário. A Seção B.3 detalha os cálculos das estimativas dos custos para este estudo de caso.

Por fim, foi analisada a estimativa dos custos para dois anos considerando as estratégias de replicação de BD. As Figuras 31 (a) e (b) apresentam gráficos com estimativa de custos para dois anos de operação, utilizando as configurações assíncrona e semisíncrona, respectivamente. A análise contempla os COP, CA, CO, CM e o custo total. Considerando esses tipos de custos, o custo total máximo para dois anos de operação chega a US$ 19.462,19 para a configuração assíncrona e US$ 19.645,57 para a configuração semisíncrona. Semelhante aos outros estudos de casos, os COP são os que mais impactam

Tabela 36 – Resultados da análise e simulação estacionária para as métricas das DSPNs da Figura 30.

Métrica	Resultados config. assíncrona	Resultados config. semisíncrona
$\mathbf{Prob_{entrada}}$	0,99787779	0,99787242 [0,99786958 - 0,99787525]
$\mathbf{Prob_{BD1op}}$	0,99817597	0,99817169 [0.99816904 - 0.99817435]
$\mathbf{Prob_{envio}}$	0,99610535	0,99609799 [0,99609636 - 0,99609962]
$\mathbf{Prob_{assinc}}$	-	0,00194116 [0,00193868 - 0,00194365]
$\mathbf{Tmp_{ativacao}}$	0,02117695 horas	0,02190006 horas [0,02168123 - 0,02211889]
$\mathbf{Tmp_{aplicacao}}$	0,24205644 segundos	0,48567508 segundos
Disponibilidade	0,99843347	0,99841683 [0,99841418, 0,99841947]
Indisponibilidade/ano	13,7228028 horas	13,8685692 horas [13,8454428 - 13,8917832]
RPO	0,27161950 segundos	0,00046767 segundos
RTO	0,02117695 horas	0,02190006 horas
Custo para o 1º ano	US$ 10.443,54	US$ 10.535,22

no custo total, representando aproximadamente 67% do custo total para ambas as configurações assíncrona e semisíncrona. Os demais custos representam cerca de 21% para os CM, aproximadamente 7% para os CA e aproximadamente 4% para os CO.

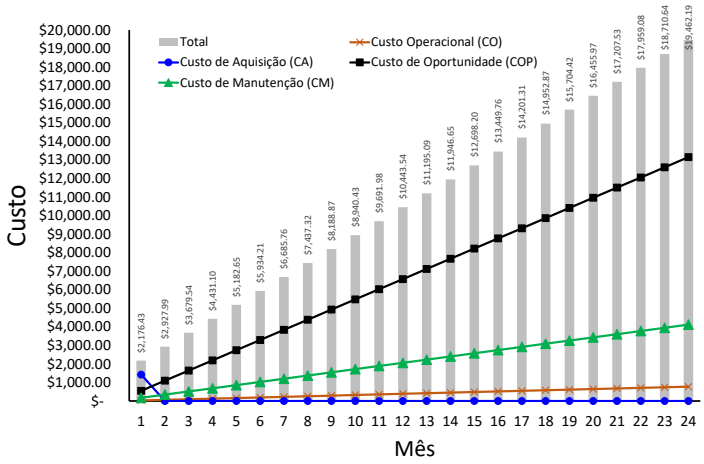
6.3.4 Experimentos para a estratégia de replicação de BDs

Os experimentos deste estudo de caso foram realizados para atender a dois objetivos: (1) obter valores referentes ao tempo de processamento de uma transação pelo servidor de BD e (2) obter o valor médio referente ao tempo de replicação de uma transação do servidor *master* para sua *replica*. Os resultados obtidos a partir desses experimentos foram utilizados como parâmetros de entrada para os modelos DSPNs apresentados anteriormente na Subseção 6.3.2. A seguir, são apresentados a arquitetura dos experimentos e os resultados dos mesmos.

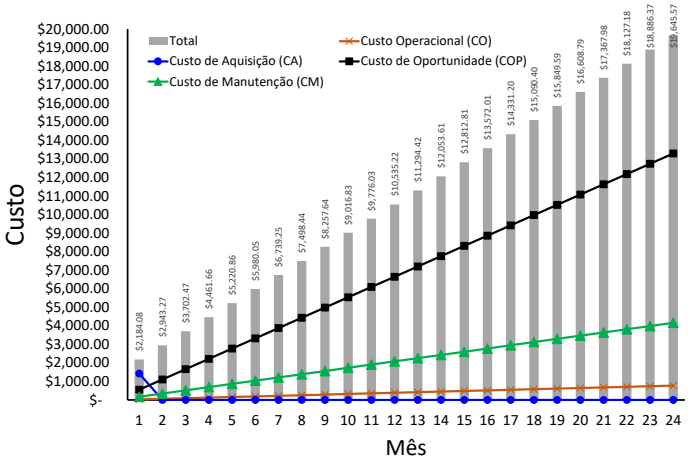
6.3.4.1 Arquitetura de testes

Com o intuito de representar um ambiente mais próximo da realidade, no qual o uso da replicação de BD está inserido, foi necessário usar um ambiente de testes distribuído geograficamente. A adoção desse ambiente de testes distribuído geograficamente ao invés de um ambiente de testes local (montado em laboratório) foi motivada principalmente pela necessidade de se obter resultados mais realísticos referentes aos tempos de replicação das transações. Em um ambiente distribuído geograficamente, os servidores de BD estão

Figura 31 – Custo acumulado para dois anos de operação da infraestrutura adotada usando a estratégia de replicação de BD com configuração assíncrona (a) e semisíncrona (b).



(a) Custo considerando a configuração assíncrona



(b) Custo considerando a configuração semisíncrona

Fonte: Criado pelo autor.

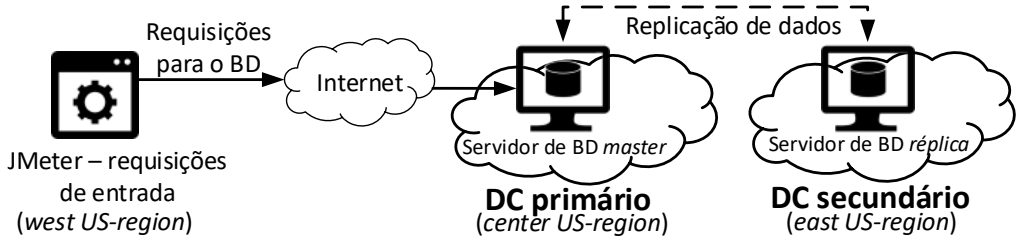
propensos a diferentes fatores que podem influenciar na replicação dos dados, os quais não estariam presentes em um ambiente local como, por exemplo, a latência de rede devido à distância geográfica entre os servidores.

Dessa forma, o ambiente de testes foi configurado em uma nuvem pública, onde foi escolhido o *Google Cloud* (GOOGLE, 2020) por apresentar as funcionalidades necessárias para realização do experimento. Assim, todo o ambiente de testes foi virtualizado. No total, foram utilizadas três VMs, onde uma foi usada para gerar as cargas dos testes, representando as requisições de usuários (ou sistema). Vale ser ressaltado que essa VM foi configurada com o Apache JMeter (HALILI, 2008). Uma outra foi utilizada para realizar o papel do servidor de BD *master*. Por fim, a terceira VM foi usada para representar o servidor de BD *replica*. Todas as VMs configuradas como servidor de BD utilizaram o MySQL 8 como SGBD. O SGBD utilizado fornecia configurações próprias que eram necessárias para realizar as sincronizações entre o BD *master* e sua réplica.

As VMs foram colocadas em diferentes DCs do *Google Cloud*, com localizações geo-

gráficas distintas. A Figura 32 apresenta como o ambiente de testes foi organizado. A localização das VMs foi definida da seguinte maneira: a VM configurada com o *JMeter*, a qual representa o envio de requisições de usuários (ou sistema), ficou em um DC na *west US-region*; o servidor de BD *master* ficou em um DC na *center US-region*; e a *réplica* ficou localizada em um DC na *east US-region*. Vale ainda mencionar que todas as VMs utilizaram como sistema operacional o Ubuntu 18.04 LTS, 2 v-CPUs, memória RAM de 7,5GB e *Solid-State Drive* (SSD) de 80GB.

Figura 32 – Organização do ambiente de testes utilizado para avaliar o processo de replicação de dados entre BDs.



Fonte: Criado pelo autor.

6.3.4.2 Resultados dos experimentos

Utilizando a arquitetura descrita, foram conduzidos experimentos para obter os tempos médios de processamento e de replicação das transações no BD. A execução dos experimentos foi realizada conforme descrevemos a seguir. A carga de trabalho foi gerada por *threads* do Apache JMeter, representando as transações dos usuários. As transações continham operações *Structured Query Language* (SQL) de *insert* e *update*, a fim de persistir dados no BD. Várias operações foram realizadas pelas mesmas *threads*, de forma que na primeira operação ela realizava uma transação de *insert*, e nas demais operações uma transação de *update* no registro já gerado. Vale ser ressaltado que essas operações de *insert* e *update* inseriam e modificavam um campo do tipo *blob* (dado binário) (ORACLE, 2020) em uma única tabela no BD *master*, o qual possuía um tamanho fixo de 500 KB.

Os valores para os tempos médios de processamento e de replicação das transações das *threads* foram coletados, respectivamente, através de um *parsing* nos registros do *binary log* dos BDs e de consultas ao BD *performance_schema* disponível no MySQL 8 (Oracle, 2020a). A Tabela 37 exibe a média dos resultados, de acordo com o tipo de configuração adotada. Além disso, também são apresentados entre parênteses os desvios padrões de cada métrica obtida nos experimentos.

Os tempos de processamento apresentaram mais variação de acordo com o tipo da replicação utilizada. O resultado foi de 240,162 ms para processar transações quando a replicação assíncrona foi adotada. Já para a replicação semisíncrona, o resultado foi

Tabela 37 – Resultados dos experimentos para replicação de BD utilizando as configurações assíncrona e semisíncrona.

Parâmetro associado	Descrição da Métrica Obtida	Resultado (desv. pad.)
DCT_{p_assinc}	Tempo de processamento para as transações assíncronas	240,162 (109,440) ms
$DCT_{p\_semisinc}$	Tempo de processamento para as transações semisíncronas	482,951 (161,344) ms
DCT_{r_assinc}	Tempo de replicação das transações assíncronas	270,887 (112,217) ms
$DCT_{p\_semisinc}$	Tempo de replicação das transações semisíncronas	240,970 (109,352) ms

de 482,951 ms. Essa diferença está relacionada ao modo de operação dos dois tipos de replicação, uma vez que na replicação semisíncrona o BD *master* só faz o *commit* da transação quando a *replica* confirma o recebimento da mesma. Por outro lado, o tempo médio de replicação das transações foram semelhantes entre os tipos de replicação utilizada. Utilizando a replicação assíncrona, o tempo médio de replicação observado foi de 270,887ms, enquanto para a replicação semisíncrona, o valor foi de 240,940 ms.

6.4 AVALIAÇÃO DE UMA INFRAESTRUTURA COMPUTACIONAL UTILIZANDO REPLICAÇÃO DE AMBIENTES COMO ESTRATÉGIA DE DR

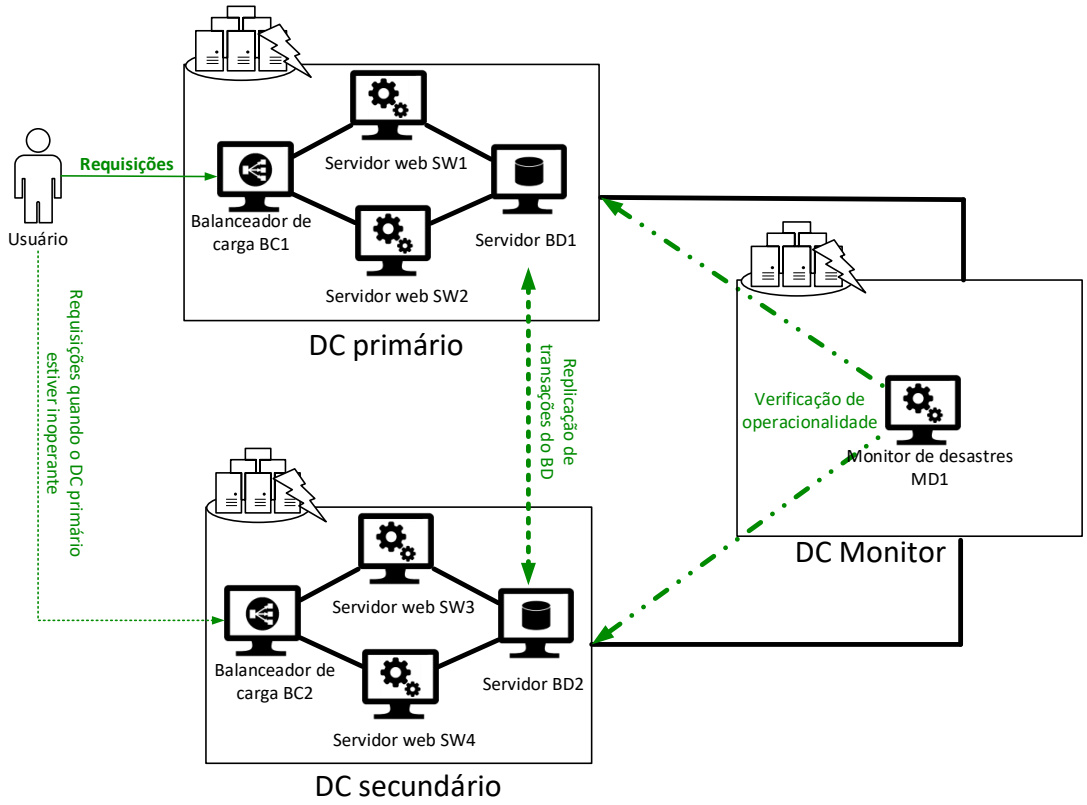
Neste estudo de caso, é analisada uma infraestrutura computacional que adota a replicação de ambiente como estratégia de DR. Nessa estratégia, toda a infraestrutura computacional presente no DC primário é replicada em um DC secundário. Geralmente, essa estratégia é adotada quando as empresas necessitam manter seus sistemas acessíveis para seus usuários pelo máximo período de tempo possível. O uso da replicação de ambiente tende a aumentar a disponibilidade dos sistemas uma vez que as réplicas dos sistemas presentes no DC secundário podem assumir as operações do DC primário, caso o mesmo falhe. Neste estudo de caso, é considerado o uso da replicação de ambiente utilizando dois tipos de configurações: (*1*) *Hot-standby* e (*2*) *warm-standby* (ver diferenças das configurações na Subseção 2.4.2.3). A seguir, é descrito o ambiente adotado no estudo de caso. Depois, são apresentados os modelos DSPNs criados e, por fim, são discutidos os resultados obtidos através dos modelos DSPNs.

6.4.1 Ambiente adotado

A Figura 33 apresenta o ambiente adotado neste estudo de caso. O DC primário utiliza a mesma configuração dos estudos de caso anteriores. No DC secundário, por sua vez, é criada uma réplica de todos os elementos contidos no DC primário. Isto é, um balanceador de carga, dois servidores de aplicação e um servidor de BD. Para garantir a sincronização dos dados, também é considerada a replicação de BD. Assim, o servidor BD2 funciona como uma *réplica* do servidor BD1, de modo que recebe atualizações de dados que são inseridos no servidor BD1. Além dos dois DCs, o ambiente ainda conta com um monitor de

desastres MD1, o qual está localizado em um outro local físico. O monitor de desastres é responsável por verificar a operacionalidade dos elementos dos dois DCs. Ao detectar uma falha no DC primário, ele é o responsável por mudar as configurações de encaminhamento das requisições dos usuários, redirecionando-as para o DC secundário. Vale ressaltar que o monitor de desastres pode ser configurado de diferentes formas e não necessariamente em um terceiro DC. Apesar disso, seu uso em uma terceira infraestrutura é mais comum, a fim de mitigar falhas ou desastres de componentes em uma única localidade.

Figura 33 – Ambiente adotado que utiliza a estratégia de replicação de ambiente.



Fonte: Criado pelo autor.

No ambiente adotado, por padrão, as requisições dos usuários são enviadas para o DC primário. Assim, uma requisição chega primeiro ao balanceador de carga BC1. Este, por sua vez, decide para qual dos servidores web deve encaminhar a requisição. Por fim, a aplicação acessa o servidor BD1 para persistir ou consultar dados no BD. Para garantir que os dados dos usuários estejam presentes nos dois DCs, é adotada a replicação **assíncrona** entre os servidores de BD. Dessa forma, o servidor BD1 replica as transações recebidas para o servidor BD2, localizado no DC secundário.

A implementação da configuração *hot-standby* ou *warm-standby* afetam diretamente o processo de replicação de dados entre os servidores de BD. Na configuração *hot-standby* os componentes do DC secundário (balanceador de carga, servidores de aplicação e servidor de BD) estão totalmente ativados, implicando que o servidor BD2 estará apto a receber as

transações, caso não sofra falhas. Já na configuração *warm-standby*, os componentes do DC secundário encontram-se em estado de espera, ou seja, não estão ativos e necessitam de um tempo para isso. Tal comportamento impossibilita o servidor BD2 de receber as transações enviadas pelo servidor BD1. Assim, o servidor de BD necessita ser ativado e desativado em intervalos de tempo regulares para receber as requisições já inseridas no servidor BD1. Para esse caso, assumimos que esse processo de ativação e desativação periódica do servidor BD2 é gerenciado pelo monitor de desastres MD1.

O monitor de desastres executa operações importantes nesta estratégia de DR. Além de ser o responsável pela ativação/desativação do servidor BD2, ele também é o responsável por verificar a operacionalidade dos DCs primário e secundário. Assim, quando o MD1 identifica que os componentes do DC primário não estão operacionais, ele é encarregado de configurar o redirecionamento das requisições dos usuários para o DC secundário. Neste estudo de caso, assumimos que o redirecionamento acontece através da reconfiguração do IP dos servidores através de *scripts* previamente criados e configurados. Dentre os eventos que podem deixar o DC primário inoperante estão: (*1*) a ocorrência de algum desastre no DC; ou (*2*) a ocorrência de falhas nos componentes que não permitam o processamento correto das requisições dos usuários (ex.: falha dos elementos de rede).

Quando o monitor de desastres identifica a inoperacionalidade do DC primário, então ocorre o início do processo de ativação do DC secundário para processar as requisições. Inevitavelmente, esse processo leva mais tempo quando a configuração *warm-standby* é adotada, uma vez que, além da configuração de encaminhamento das requisições dos usuários para o DC secundário, os componentes do DC também precisam ser ativados. Seja na configuração, *hot-standby* ou *warm-standby*, o DC secundário apenas estará apto a receber as requisições dos usuários quando o servidor BD2 finalizar a persistência dos dados já recebidos do servidor BD1 que ainda não foram finalizadas, de modo a garantir a integridade dos dados. Portanto, resumidamente, há três operações para que o DC secundário passe a processar as requisições dos usuários: o redirecionamento do IP dos servidores, a ativação dos componentes do DC secundário (na configuração *warm-standby*) e a finalização da persistência dos dados do servidor BD2.

6.4.2 Modelos DSPNs

Esta subseção apresenta os modelos DSPNs para analisar o ambiente adotado quando as configurações *hot-standby* e *warm-standby* são implementadas. Primeiro, são apresentados os modelos para a configuração *hot-standby* e, em seguida, os modelos para a configuração *warm-standby*.

6.4.2.1 Modelos para a configuração *hot-standby*

A Figura 34 exibe os modelos DSPNs desenvolvidos para analisar a configuração *hot-standby* adotada neste estudo de caso. Esses modelos representam cada componente do DC

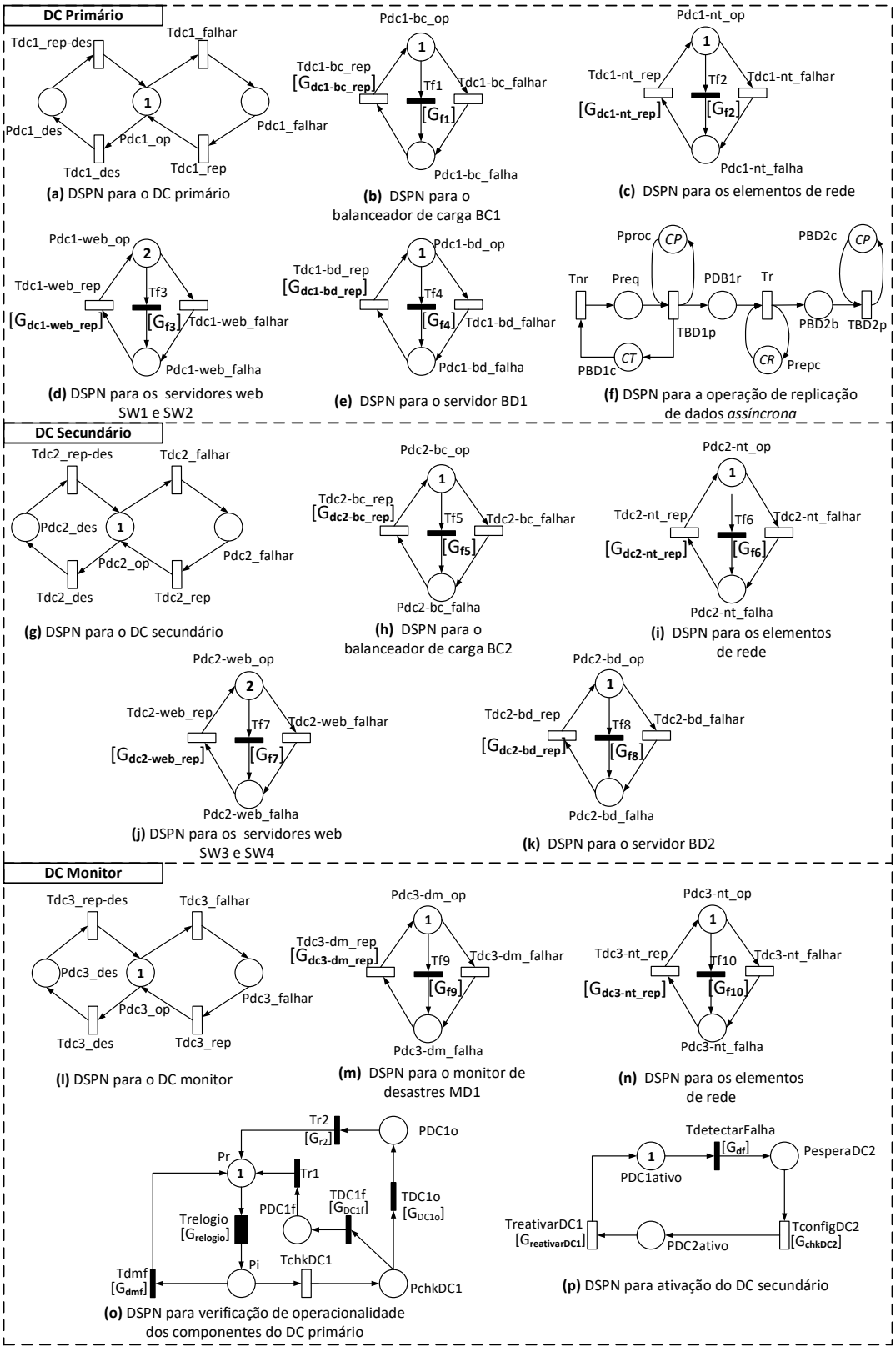
primário, do secundário e do DC monitor. Além disso, também são exibidos os modelos que representam os processos de replicação de dados, bem como o mecanismo de verificação de operacionalidade realizado pelo monitor de desastres. Na parte superior da Figura 34 (a - f) é possível observar os modelos DSPNs para os componentes do DC primário. Na parte central (ver Figuras 34 (*g* - *k*)), estão os modelos DSPNs para os componentes do DC secundário, que representam os mesmos componentes presentes no DC primário. Já na parte inferior (ver Figuras 34 (*l* - *p*)), são exibidos os modelos para o DC monitor, seus componentes e operações.

As Figuras 34 (a - e) e (*g* - *k*) apresentam os modelos DSPNs para os componentes dos DCs primário e secundário. Como o DC secundário é uma replicação da infraestrutura do DC primário, seus componentes possuem o mesmo comportamento dos componentes do DC primário. Na configuração *hot-standby*, os componentes do DC secundário estão totalmente ativados, mas não dividem a carga de trabalho com os componentes do DC primário enquanto esses ainda conseguirem processar as requisições dos usuários.

As Figuras 34 (*l* - *n*) representam, respectivamente, o DC monitor, o servidor MD1 e os elementos de rede do DC. Já as Figuras 34 (*o*) e (*p*) apresentam as DSPNs que representam o processo de verificação de operacionalidade do DC primário e a ativação do DC secundário, respectivamente. A verificação da operacionalidade é executada pelo MD1 periodicamente em um intervalo de tempo definido pela transição *Trelogio* (ver Figura 34 (*o*)). Quando o monitor verifica os estados de todos os componentes do DC primário (*TchkDC1*), ele pode encontrar uma falha (*TDC1f*) ou não (*TDC1o*). É importante ser ressaltado que todas as verificações são apoiadas por funções de guarda, as quais verificam as marcações alcançadas pelos modelos DSPNs e, consequentemente, os estados dos componentes. Essa operação é realizada periodicamente desde que o MD1 esteja operacional (através da função de guarda $G_{relogio}$).

A DSPN para a ativação do DC secundário (ver Figura 34 (*p*)) também faz o uso de funções de guardas para ativar o DC secundário (*TconfigDC2*) quando uma falha é detectada no DC primário (*TdetectarFalha*). Nesse modelo, a presença de um *token* no lugar *PDC2ativo* indica que o DC secundário está ativo e processando requisições dos usuários. Quando isso ocorre, o DC primário só retorna a processar as requisições dos usuários (*TreativarDC1*) quando o MD1 verificar que todos os seus componentes estão operacionais ($G_{reativarDC1}$). Para um melhor detalhamento dos modelos criados, a Tabela 38 exibe todas as funções de guarda utilizadas nos modelos DSPNs da Figura 34, enquanto a Tabela 39 apresenta as características das transições.

Figura 34 – Modelos DSPNs criados para o ambiente exibido na Figura 33 considerando a configuração *hot-standby*.



Fonte: Criado pelo autor.

Tabela 38 – Funções de guarda utilizadas nos modelos DSPNs da Figura 34.

Guarda	**Função de Habilitação**
DC Primário	
G_{f1}, G_{f3}, G_{f4}	*(#Pdc1-op = 0)*
G_{f2}	*(#Pdc1_des = 1)*
G_{dc1-bc_rep}, G_{dc1-nt_rep}, $G_{dc1-web\_rep}$, G_{dc1-bd_rep}	*(#Pdc1-op = 1)*
DC Secundário	
G_{f5}, G_{f7}, G_{f8}	$(\#Pdc2\_op = 0)$
G_{f6}	$(\#Pdc2\_des = 1)$
G_{dc2-bc_rep}, G_{dc2-nt_rep}, $G_{dc2-web\_rep}$, G_{dc2-bd_rep}	*(#Pdc2-op = 1)*
DC Monitor	
G_{f9}	$(\#Pdc3\_op = 0)$
G_{f10}	$(\#Pdc3\_des = 1)$
G_{df}	$(\#PDC1f = 1)$
$G_{reativarDC1}$	$(\#PDC1o = 1)$
G_{chkDC2}	$(\#Pdc2\text{-}nt\_op = 1)$ AND $(\#Pdc2\text{-}bc\_op > 0)$ AND $(\#Pdc2\text{-}web\_op > 0)$ AND $(\#Pdc2\text{-}bd\_op > 0)$
$G_{relogio}$	$(\#Pdc3\text{-}dm\_op > 0)$
G_{dmf}	$(\#Pdc3\text{-}dm\_op = 0)$
G_{DC1f}	$(\#Pdc1\text{-}nt\_op = 0)$ OR $(\#Pdc1\text{-}bc\_op = 0)$ OR $(\#Pdc1\text{-}web\_op = 0)$ OR $(\#Pdc1\text{-}bd\_op = 0)$
G_{DC1o}	$(\#Pdc1\text{-}nt\_op = 1)$ AND $(\#Pdc1\text{-}bc\_op > 0)$ AND $(\#Pdc1\text{-}web\_op > 0)$ AND $(\#Pdc1\text{-}bd\_op > 0)$
G_{r2}	$(\#PDC2ativo = 0)$

Tabela 39 – Descrição das transições dos modelos DSPNs da Figura 34.

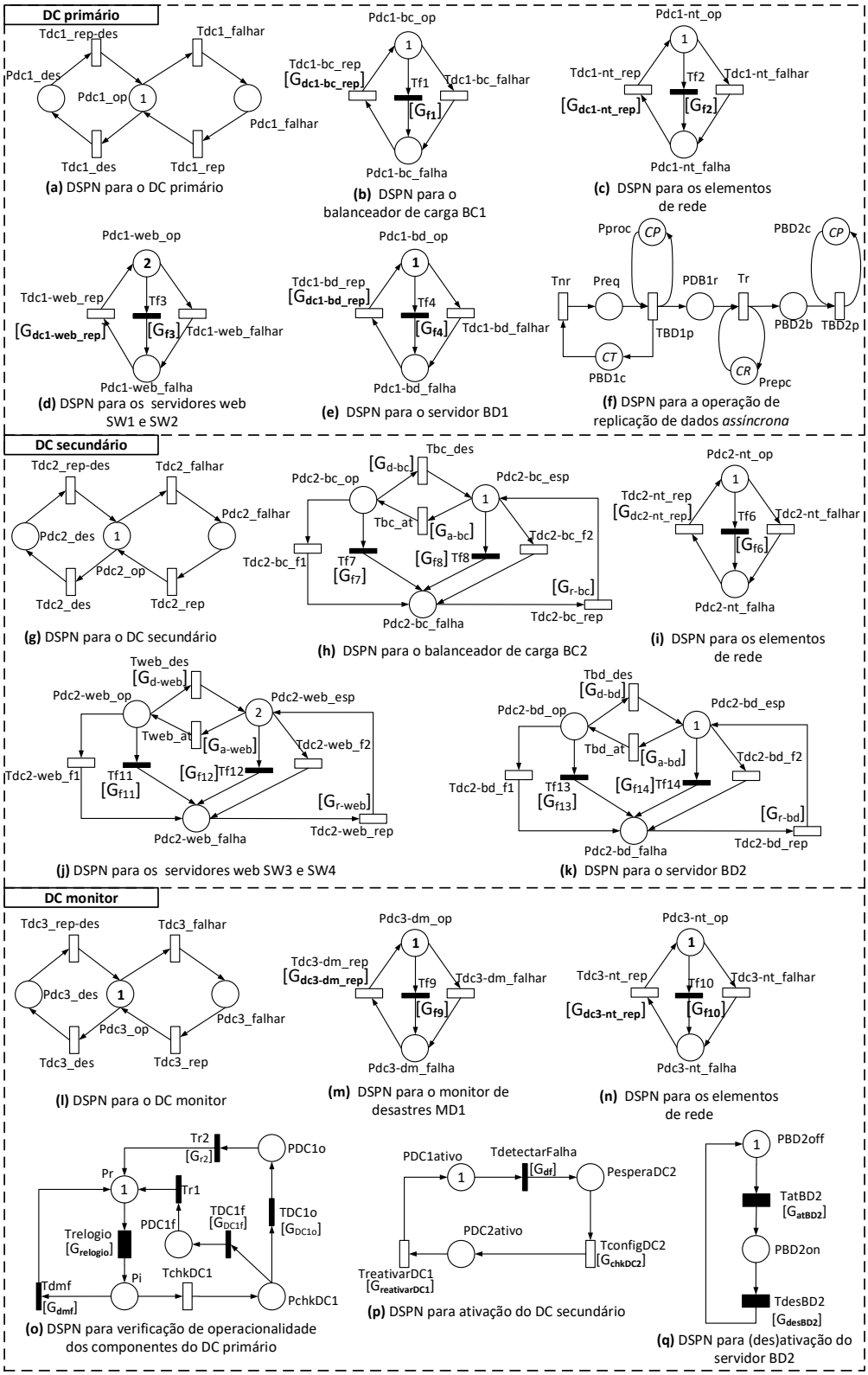
Transições	**Tipo**	**Semântica**	**Peso**	**Prioridade**
DC Primário				
Tdc1_des, *Tdc1_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tdc1_rep, *Tdc1_rep-des*	Temporizada (exp)	*Single-server*	-	-
Tdc1-web_rep, *Tdc1-bd_rep*	Temporizada (exp)	*Single-server*	-	-
Tdc1-bc_rep ,*Tdc1-nt_rep*, *Tnr*	Temporizada (exp)	*Single-server*	-	-
Tdc1-bd_falhar, *Tdc1-web_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tdc1-bc_falhar, *Tdc1-nt_falhar*	Temporizada (exp)	*Infinite-server*	-	-
TBD1p, *Tr*, *TBD2p*	Temporizada (exp)	*Infinite-server*	-	-
Tf1, *Tf2*, *Tf3*, *Tf4*	Imediata	-	1	1
DC secundário				
Tdc2_rep, *Tdc2_falhar*	Temporizada (exp)	*Single-server*	-	-
Tdc2_des, *Tdc2_rep-des*	Temporizada (exp)	*Single-server*	-	-
Tdc2-web_rep, *Tdc2-bd_rep*	Temporizada (exp)	*Single-server*	-	-
Tdc2-bc_rep, *Tdc2-nt_rep*	Temporizada (exp)	*Single-server*	-	-
Tdc2-web_falhar, *Tdc2-bd_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tdc2-bc_falhar, *Tdc2-nt_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tf5, *Tf6*, *Tf7*, *Tf8*	Imediata	-	1	1
DC Monitor				
Tdc3_rep, *Tdc3_falhar*	Temporizada (exp)	*Single-server*	-	-
Tdc3_des, *Tdc3_rep-des*	Temporizada (exp)	*Single-server*	-	-
Tdc3-dm_rep, *Tdc3-nt_rep*	Temporizada (exp)	*Single-server*	-	-
Tdc3-dm_falhar, *Tdc3-nt_falhar*	Temporizada (exp)	*Infinite-server*	-	-
TchkDC1, *TconfigDC2*	Temporizada (exp)	*Single-server*	-	-
TreativarDC1	Temporizada (exp)	*Single-server*	-	-
Trelogio	Temporizada (det)	*Single-server*	-	-
Tf9, *Tf10*, *Tdmf*, *TDC1f,TDC1o*	Imediata	-	1	1
Tr1, *Tr2*	Imediata	-	1	0
TdetectarFalha	Imediata	-	1	1

6.4.2.2 Modelos para a configuração *warm-standby*

A Figura 35 apresenta os modelos DSPNs para o ambiente adotado seguindo a configuração *warm-standby*. Esses modelos são semelhantes aos da configuração *hot-standby*, porém modificações foram feitas nos modelos que representam os componentes do DC secundário. Mais especificamente, foram adotados os modelos DSPNs para representação de componentes em modo *warm-standby* (ver Subseção 5.1.1) para o balanceador de carga BC2, os servidores web SW3 e SW4 e o servidor BD2 (ver Figuras 35 (h), (j) e (k). Além disso, também foi utilizado um modelo DSPN para controlar a ativação periódica do servidor BD2 (ver Figura 35 (q)). Através da ativação periódica do servidor BD2, o mesmo pode receber as transações salvas no servidor BD1 periodicamente. Esse modelo é baseado no modelo apresentado anteriormente na Figura 16.

Adaptações nas funções de guarda bem como a criação de novas funções também foram realizadas, a fim de tornar a execução do modelo condizente com a configuração *warm-standby*. A Tabela 40 detalha todas as funções de guarda utilizadas nos modelos DSPNs da Figura 35. Dentre as novas funções criadas pode-se destacar $G_{a\text{-}bc}$, $G_{a\text{-}web}$ e $G_{a\text{-}bd}$ que são responsáveis por verificar a possibilidade de ativação do balanceador de carga BC2, dos servidores SW3 e SW4 e do servidor BD2 no DC secundário, respectivamente. Nas DSPNs desses mesmos componentes, as funções $G_{d\text{-}bc}$, $G_{d\text{-}web}$ e $G_{d\text{-}bd}$ são responsáveis por verificar a possibilidade de desativação dos mesmos. Por fim, a Tabela 41 detalha as transições utilizadas nos modelos DSPNs apresentados na Figura 35.

Figura 35 – Modelos DSPNs criados para o ambiente exibido na Figura 33 considerando a configuração *warm-standby*.



Fonte: Criado pelo autor.

Tabela 40 – Funções de guarda utilizadas nos modelos DSPNs da Figura 35.

Guarda	**Função de Habilitação**
DC Primário	
G_{f1}, G_{f3}, G_{f4}	*(#Pdc1-op = 0)*
G_{f2}	*(#Pdc1_des = 1)*
G_{dc1-bc_rep}, G_{dc1-nt_rep}, $G_{dc1-web\_rep}$, G_{dc1-bd_rep}	*(#Pdc1-op = 1)*
DC Secundário	
G_{f6}	$(\#Pdc2\_des = 1)$
G_{f7}, G_{f8}, G_{f11}, G_{f12}, G_{f13},G_{f14}	$(\#Pdc2\_op = 0)$
$G_{a\text{-}bc}$, $G_{a\text{-}web}$	$(\#PesperaDC2{=}1)$ OR $(\#PDC2ativo{=}1)$
$G_{a\text{-}bd}$	$(\#PesperaDC2{=}1)$ OR $(\#PDC2ativo{=}1)$ OR $(\#PBD2on{=}1)$
$G_{d\text{-}bc}$, $G_{d\text{-}web}$	$(\#PDC2ativo{=}0)$
$G_{d\text{-}bd}$	$(\#PDC2ativo{=}0)$ AND $(\#PBD2off{=}1)$
G_{dc2-bc_rep}, G_{dc2-nt_rep}, $G_{dc2-web\_rep}$, G_{dc2-bd_rep}	*(#Pdc2-op = 1)*
DC Monitor	
G_{f9}	$(\#Pdc3\_op = 0)$
G_{f10}	$(\#Pdc3\_des = 1)$
G_{df}	$(\#PDC1f = 1)$
$G_{reativarDC1}$	$(\#PDC1o = 1)$
G_{chkDC2}	$(\#Pdc2\text{-}nt\_op = 1)$ AND $(\#Pdc2\text{-}bc\_op > 0)$ AND $(\#Pdc2\text{-}web\_op > 0)$ AND $(\#Pdc2\text{-}bd\_op > 0)$
$G_{relogio}$	$(\#Pdc3\text{-}dm\_op > 0)$
G_{dmf}	$(\#Pdc3\text{-}dm\_op = 0)$
G_{DC1f}	$(\#Pdc1\text{-}nt\_op = 0)$ OR $(\#Pdc1\text{-}bc\_op = 0)$ OR $(\#Pdc1\text{-}web\_op = 0)$ OR $(\#Pdc1\text{-}bd\_op = 0)$
G_{DC1o}	$(\#Pdc1\text{-}nt\_op = 1)$ AND $(\#Pdc1\text{-}bc\_op > 0)$ AND $(\#Pdc1\text{-}web\_op > 0)$ AND $(\#Pdc1\text{-}bd\_op > 0)$
G_{r2}	$(\#PDC2ativo = 0)$
G_{atBD2}	$(\#Pdc3\text{-}dm\_op = 1)$ AND $(\#Pdc3\text{-}nt\_op = 1)$ AND $(\#Pdc2\text{-}nt\_op = 1)$
G_{desBD2}	$(\#Pdc3\text{-}dm\_op = 1)$ AND $(\#Pdc3\text{-}nt\_op = 1)$ AND $(\#Pdc2\text{-}nt\_op = 1)$ AND $(\#PDC1ativo = 1)$

Tabela 41 – Descrição das transições dos modelos DSPNs da Figura 35.

Transições	**Tipo**	**Semântica**	**Peso**	**Prioridade**
DC Primário				
Tdc1_des, *Tdc1_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tdc1_rep, *Tdc1_rep-des*	Temporizada (exp)	*Single-server*	-	-
Tdc1-web_rep, *Tdc1-bd_rep*	Temporizada (exp)	*Single-server*	-	-
Tdc1-bc_rep ,*Tdc1-nt_rep*, *Tnr*	Temporizada (exp)	*Single-server*	-	-
Tdc1-bd_falhar, *Tdc1-web_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tdc1-bc_falhar, *Tdc1-nt_falhar*	Temporizada (exp)	*Infinite-server*	-	-
TBD1p, *Tr*, *TBD2p*	Temporizada (exp)	*Infinite-server*	-	-
Tf1, *Tf2*, *Tf3*, *Tf4*	Imediata	-	1	1
DC secundário				
Tdc2_falhar, *Tdc2_des*	Temporizada (exp)	*Infinite-server*	-	-
Tdc2_rep-des, *Tdc2-web_rep*, *Tdc2-bd_rep*	Temporizada (exp)	*Single-server*	-	-
Tdc2-bc_rep, *Tdc2-nt_rep*, *Tbc-at*	Temporizada (exp)	*Single-server*	-	-
Tdc2_rep, *Tweb-at*, *Tbd-at*	Temporizada (exp)	*Single-server*	-	-
Tbc-des, *Tweb-des*, *Tbd-des*	Temporizada (exp)	*Infinite-server*	-	-
Tdc2-web_f1, *Tdc2-web_f2*	Temporizada (exp)	*Infinite-server*	-	-
Tdc2-bd_f1, *Tdc2-bd_f2*	Temporizada (exp)	*Infinite-server*	-	-
Tdc2-bc_f1, *Tdc2-bc_f12*, *Tdc2-nt_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tf6, *Tf7*, *Tf8*, *Tf11*, *Tf12*, *Tf13*, *Tf14*	Imediata	-	1	1
DC Monitor				
Tdc3_des, *Tdc3_falhar*	Temporizada (exp)	*Infinite-server*	-	-
Tdc3_rep, *Tdc3_rep-des*	Temporizada (exp)	*Single-server*	-	-
Tdc3-dm_rep, *Tdc3-nt_rep*	Temporizada (exp)	*Single-server*	-	-
Tdc3-dm_falhar, *Tdc3-nt_falhar*	Temporizada (exp)	*Infinite-server*	-	-
TchkDC1, *TconfigDC2*, *TreativarDC1*	Temporizada (exp)	*Single-server*	-	-
Trelogio, *TatBD2*, *TdesBD2*	Temporizada (det)	*Single-server*	-	-
Tf9, *Tf10*, *Tdmf*, *TDC1f*,*TDC1o*	Imediata	-	1	1
Tr1, *Tr2*	Imediata	-	1	0
TdetectarFalha	Imediata	-	1	1

6.4.3 Resultados dos modelos DSPNs

Nesta subseção, são apresentados os resultados obtidos através da simulação dos modelos DSPNs apresentados. Os parâmetros de entrada para os modelos são baseados nos trabalhos de Kim, Machida e Trivedi (2009), Machida et al. (2011), Andrade et al. (2017) e são apresentados pela Tabela 42. Assim como nos estudos de caso anteriores, os parâmetros de entrada para estimar o custo tiveram seus valores baseados nos trabalhos de Dantas et al. (2015), Andrade e Nogueira (2019) e são apresentados na Tabela 20 (ver Subseção 6.1.3).

Tabela 42 – Parâmetros de entrada para os modelos DSPNs da Figura 34 e 35.

Parâmetro	Descrição	Transição associada	Valor (h)
Todos os DCs			
DC_{tf}	MTTF DC (transiente)	*Tdc1_falhar*, *Tdc2_falhar*, *Tdc3_falhar*	8760
DC_{tr}	MTTR DC (transiente)	*Tdc1_rep*, *Tdc2_rep*, *Tdc3_rep*	4
DC_{d}	Tempo médio para desastre	*Tdc1_des*, *Tdc2_des*, *Tdc3_des*	17520
DC_{rd}	Tempo médio para reparar desastre	*Tdc1_rep-des*, *Tdc2_rep-des*, *Tdc3_rep-des*	12
DCs Primário e Secundário			
DC_{lbr}	MTTR balanceador de carga	*Tdc1-bc_rep*,*Tdc2-bc_rep*	0,5
DC_{lbf}	MTTF balanceador de carga	*Tdc1-bc_falhar*, *Tdc2-bc_falhar*, *Tdc2-bc_f1*, *Tdc2-bc_f2*	8760
DC_{vmf}	MTTF servidor Web/BD	*Tdc1-web_falhar*, *Tdc1-bd_falhar*, *Tdc2-web_falhar*, *Tdc2-bd_falhar*, *Tdc2-web_f1*, *Tdc2-bd_f1*, *Tdc2-web_f2*, *Tdc2-bd_f2*	2654
DC_{vmr}	MTTR servidor Web/BD	*Tdc1-web_rep*, *Tdc1-bd_rep*, *Tdc2-web_rep*, *Tdc2-bd_rep*	1,25
DC Primário			
DCT_{nr}	Tempo médio de chegada das transações	*Tnr*	0,000138
DCT_{p_assinc}	Tempo médio de processar uma transação Assínc.	*TBD1p*, *TBD2p*	6,67116667E-5
DCT_{r_assinc}	Tempo médio de replicar uma transação Assínc.	*Tr*	7,52463889E-5
CT	Capacidade de transações no servidor	-	10
CP	Capacidade de processamento de transações	-	10
CR	Capacidade de replicação	-	10
DC Secundário			
DC_{ca}	Tempo médio de ativação dos componentes	*Tbc_at*, *Tweb_at*,*Tbd_at*	0,0333
DC_{cd}	Tempo médio de desativação dos componentes	*Tbc_des*, *Tweb_des*,*Tbd_des*	0,0333
Rede			
N_f	MTTF elementos de rede	*Tdc1-nt_falhar*,*Tdc2-nt_falhar*, *Tdc3-nt_falhar*	10000
N_r	MTTR elementos de rede	*Tdc1-nt_rep*,*Tdc2-nt_rep*, *Tdc3-nt_rep*	1
DC Monitor			
DM_f	MTTF Monitor de desastres	*Tdc3-dm_falhar*,	2654
DM_r	MTTR Monitor de desastres	*Tdc3-dm_rep*	1,25
DM_{re}	Intervalo de verificação de operacionalidade	*Trelogio*	0,0166667
DM_{hc}	Tempo médio de verificação de operacionalidade	*TchkDC1*	2,7777E-4
DM_a	Tempo médio para ativar DCs	*TconfigDC2*, *TreativarDC1*	0,0166667
DM_{aBD2}	Intervalo de ativação do servidor BD2	*TatBD2*, *TdesBD2*	1

Utilizando os parâmetros de entrada, foram calculadas as métricas de disponibilidade, indisponibilidade, RTO e RPO através da análise (configuração *hot-standby*) e simulação estacionária (configuração *warm-standby*). As métricas computadas neste estudo de caso são baseadas nas métricas definidas na Subseção 5.1.4.1. O custo da estratégia de replicação de ambiente adotada é calculado utilizando a Equação 5.20, considerando a aquisição de duas máquinas físicas (servidores) que representam a infraestrutura do DC secundário e do DC monitor, respectivamente. A seguir, são detalhadas as métricas adotadas tanto para a configuração *hot-standby* quanto para a configuração *warm-standby*:

- A disponibilidade é obtida pela probabilidade de se ter os elementos de rede, o balanceador de carga BC1, ao menos um servidor web (SW1 ou SW2) e o servidor BD1 operacionais no DC primário, estando este DC também ativado para receber as requisições dos usuários; **OU** os elementos de rede, o balanceador de carga BC2, ao menos um servidor web (SW3 ou SW4) e o servidor BD2 operacionais no DC secundário, estando este DC também ativado para receber as requisições dos usuários;

- O RTO é definido pelo *tempo máximo entre:* (*1*) o tempo decorrido desde a identificação de uma falha em um componente do DC primário ou evento de desastre no mesmo, até o DC secundário estar ativado e capaz de processar as requisições dos usuários **e** (*2*) o tempo necessário para o servidor BD2 aplicar as transações já recebidas BD1;

- O RPO é definido pelo tempo médio que as transições levam para serem replicadas, pois se alguma falha ou evento de desastre acontecer nesse intervalo de tempo, as transações que ainda não foram replicadas serão perdidas;

- O custo é definido pelo valor de aquisição de dois servidores, mais o custo de operação, manutenção e oportunidade do ambiente adotado.

Vale destacar que os modelos precisam ser executados de forma hierárquica, devido ao uso da replicação de BD. Assim, a execução dos modelos é feita em duas etapas: (*1*) são executados os modelos DSPNs das Figuras 34 e 35 com exceção da DSPN exibida em (*f*); posteriormente (*2*) é executada a DSPN *para a operação da replicação de dados assíncrona* (*f*) separadamente. Na primeira etapa da execução dos modelos são calculadas as métricas de Disponibilidade e Indisponibilidade. Nessa etapa, também são computadas as métricas secundárias $\mathbf{Prob_{entrada}}$, $\mathbf{Prob_{BD1op}}$, $\mathbf{Prob_{envio}}$ e $\mathbf{Tmp_{ativacao}}$, que possuem as mesmas características definidas no estudo de caso anterior. Os resultados dessas métricas secundárias servem como parâmetros de entrada para a execução do modelo DSPN da segunda etapa, com exceção da métrica $\mathbf{Tmp_{ativacao}}$, que faz parte do cálculo do RTO.

Na segunda etapa da execução dos modelos, são obtidos os valores para o RPO e para a métrica secundária $\mathbf{Tmp_{aplicacao}}$, que também faz parte do cálculo para obtenção do

RTO. Sendo que $\mathbf{Tmp_{aplicacao}}$ consiste do tempo médio necessário para o servidor BD2 aplicar as transações já recebidas do servidor BD1 antes de se tornar *master*. Assim, o RTO é dado pelo valor máximo entre as métricas secundárias $\mathbf{Tmp_{ativacao}}$ e $\mathbf{Tmp_{aplicacao}}$, onde $RTO = max(T\_ativacao, T\_aplicacao)$. A Tabela 43 detalha as fórmulas para obter as métricas de disponibilidade, indisponibilidade, RPO e as métricas secundárias para ambas as configurações (*hot-standby* e *warm-standby*).

Tabela 43 – Expressões usadas para calcular as métricas adotadas neste estudo de caso.

Métrica	**Expressão**
Executada na primeira etapa	
Disponibilidade	$P\{((\#Pdc1\text{-}nt\_op{>}0)$ AND $(\#Pdc1\text{-}bc\_op{>}0)$ AND $(\#Pdc1\text{-}web\_op{>}0)$ AND $(\#Pdc1\text{-}bd\_op{>}0)$ AND $(\#PDC1ativo{=}1))$ OR $((\#Pdc2\text{-}nt\_op{>}0)$ AND $(\#Pdc2\text{-}bc\_op{>}0)$ AND $(\#Pdc2\text{-}web\_op{>}0)$ AND $(\#Pdc2\text{-}bd2\_op{>}0)$ AND $(\#PDC2ativo{=}1))\}$
Indisponibilidade	*1 -* ***Disponibilidade***
$\mathbf{Prob_{entrada}}$	$P\{(\#Pdc1\text{-}nt\_op{>}0)$ AND $(\#Pdc1\text{-}bc\_op{>}0)$ AND $(\#Pdc1\text{-}web\_op{>}0)$ AND $(\#Pdc1\text{-}bd\_op{>}0)$ AND $(\#PDC1ativo{=}1)\}$
$\mathbf{Prob_{BD1op}}$	$P\{\#Pdc1\text{-}bd\_op{>}0\}$
$\mathbf{Prob_{envio}}$	$P\{(\#Pdc1\text{-}nt\_op{>}0)$ AND $(\#Pdc2\text{-}nt\_op{>}0)$ AND $\#Pdc1\text{-}bc\_op{>}0)$ AND $(\#Pdc2\text{-}bd\_op{>}0)$ AND $(\#Pdc1\text{-}des{=}0)\}$
$\mathbf{Tmp_{ativacao}}$	$\mathrm{P}\{\#PesperaDC2{>}0\}$ / $(\mathrm{P}\{(\#Pdc2\text{-}nt\_op{>}0)$ AND $(\#Pdc2\text{-}bc\_op{>}0)$ AND $(\#Pdc2\text{-}web\_op{>}0)$ AND $(\#Pdc2\text{-}bd2\_op{>}0)\}$ / $DM_a)$
Executada na segunda etapa	
$\mathbf{Tmp_{aplicacao}}$	$\mathrm{E}\{\#PBD2b\} \times (1$ / $((1$ / $\mathrm{R_{nr}}) \times \mathbf{Prob_{entrada}} \times \mathrm{P}\{\#PBD1c{>}0\} \times \mathbf{Prob_{BD1op}} \times \mathrm{P}\{\#Pproc{>}0\} \times \mathbf{Prob_{envio}} \times \mathrm{P}\{\#Prepc{>}0\}))$
RPO	$\mathrm{E}\{\#PBD1r\} \times (1$ / $((1$ / $R_{nr}) \times \mathbf{Prob_{entrada}} \times \mathrm{P}\{\#PBD1c{>}0\} \times \mathbf{Prob_{BD1op}} \times \mathrm{P}\{\#Pproc{>}0\}))$

Os resultados das métricas obtidos através da execução dos modelos DSPNs são apresentados na Tabela 44, tanto para a configuração *hot-standby* quanto para a configuração *warm-standby*. A tabela também inclui a estimativa do custo para o primeiro ano de funcionamento do ambiente adotado. Os resultados ressaltam as diferenças e as similaridades entre as configurações *hot-standby* e *warm-standby* para a estratégia de replicação de ambiente. O valor obtido para disponibilidade média do ambiente possui uma diferença razoável entre as configurações analisadas. Enquanto a configuração *hot-standby* teve uma disponibilidade média de 0,99996158, o valor obtido quando a configuração *warm-standby* foi de 0,99986994. A diferença fica mais evidente quando é observada a indisponibilidade anual média, que foi de 0,3365592 horas para configuração *hot-standby* e 1,1393256 horas para a *warm-standby*.

De forma semelhante, os resultados para o RTO também mostram uma diferença razoá-

vel entre as duas configurações analisadas. Enquanto o valor do RTO para a configuração *hot-standby* foi de 0,01666669 horas, para a *warm-standby* esse valor foi quase 100 vezes maior de 0,16231441 horas. Por outro lado, os resultados para o RPO foram bastante similares para ambas as configurações, apresentando valores próximos de zero. Quando a configuração *hot-standby* é adotada, o RPO é de 0,27104497 segundos. Na configuração *warm-standby*, o resultado foi bastante similar de 0,27138210 segundos.

O custo estimado para o primeiro ano de execução do ambiente considerado foi de US$ 3.641,87 e US$ 4.069,51 para as configurações *hot-standby* e *warm-standby*, respectivamente. A estimativa do custo contempla a aquisição de dois servidores para a infraestrutura do DC secundário e do DC monitor. Para calcular os custos operacionais dos DCs, foram utilizados os valores da disponibilidade das máquinas físicas que compõem cada DC para o parâmetro A_i da Equação 5.25. Para a configuração *hot-standby*, os valores dos parâmetros foram: 0,998600657, 0,998991221 e 0,998538312 para o DC primário, secundário e monitor, respectivamente. Já na configuração *warm-standby*, os valores dos parâmetros foram: 0,998964134, 0,499571278, 0,998802029 para o DC primário, secundário e monitor, respectivamente. Vale destacar que o valor assumido do parâmetro A_i para o DC secundário na configuração *warm-standby* é mais divergente entre os demais. Isso se dá, pois, nessa configuração, o parâmetro representa a probabilidade do DC secundário possuir algum de seus componentes em estado operacional. Os demais valores apresentam uma pequena variação devido ao uso da análise ou simulação estacionária dos modelos realizada para cada configuração. A Seção B.4 detalha o cálculo dos custos para o ambiente adotado.

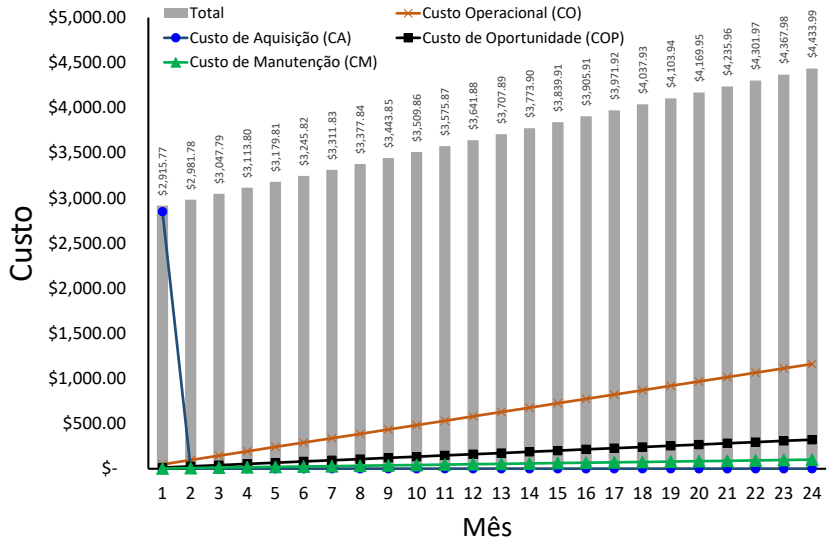
Tabela 44 – Resultados das métricas para os ambientes adotados.

Métrica	**Resultado *hot-standby***	**Resultado *warm-standby***
$\mathbf{Prob_{entrada}}$	0,99759276	0,99794713
$\mathbf{Prob_{BD1op}}$	0,99792744	0,99828793
$\mathbf{Prob_{envio}}$	0,99605337	0,49824623
$\mathbf{Tmp_{ativacao}}$	0,01666669 horas	0,16231441 horas
$\mathbf{Tmp_{aplicacao}}$	0,24172339 segundos	4,83519153 segundos
Disponibilidade	0,99996158	0,99986994
Indisponibilidade/ano	0,3365592 horas	1,1393256 horas
RPO	0,27104497 segundos	0,27138210 segundos
RTO	0,01666669 horas	0,16231441 horas
Custo para o 1º ano	US$ 3.641,87	US$ 4.069,51

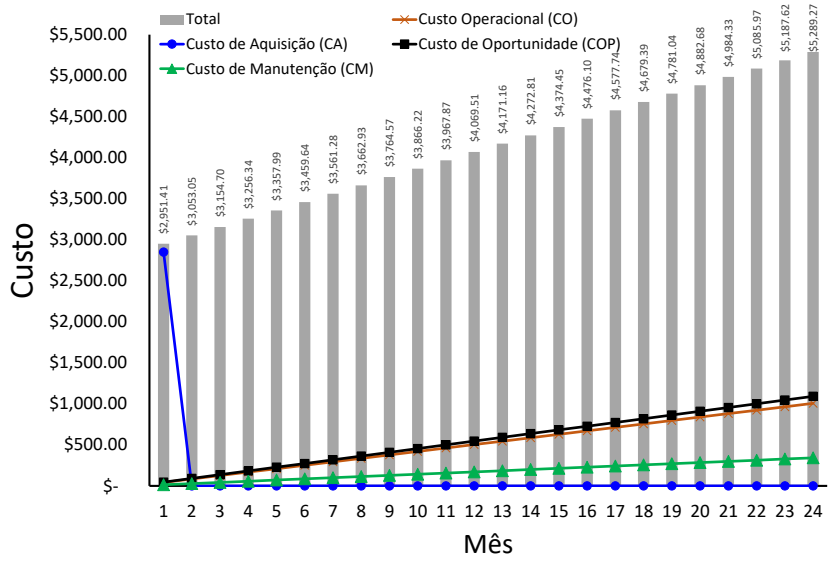
Por último, foi analisada a estimativa dos custos para dois anos de operação do ambiente adotado. Os gráficos das Figuras 36 (a) e (b) apresentam a estimativa de custos para dois anos de operação, contemplando os COP, CM, CO, CA e o custo total. Nos dois primeiros anos, considerando esses tipos de custos, o total chega a US$ 4.433,99 se implementada a configuração *hot-standby*, ou US$ 5.289,27 se adotada a configuração *warm-standby*. Isso ocorre especialmente pela maior disponibilidade da infraestrutura

alcançada com a configuração *hot-standby*. Mais disponibilidade, neste caso, implica em um menor CM e COP. Diferentemente de outras estratégias de DR analisadas, na replicação de ambiente, o CA é o custo que mais impacta no custo total. Esse tipo de custo representa aproximadamente 64% e 53% do custo total para as configurações *hot-standby* e *warm-standby*, respectivamente (ao final dos dois primeiros anos). Além disso, os gráficos revelam que quando a configuração *hot-standby* é adotada, o segundo maior custo é o CO. Já quando a configuração *warm-standby* é adotada, o segundo maior custo é o COP.

Figura 36 – Custo acumulado para dois anos de operação do DC primário utilizando a estratégia de replicação de ambiente com configuração *hot-standby* (a) e *warm-standby* (b).



(a) Custo com a configuração *hot-standby*



(b) Custo com a configuração *warm-standby*

Fonte: Criado pelo autor.

6.5 AVALIAÇÃO MULTICRITÉRIO PARA SELEÇÃO DE ESTRATÉGIAS DE DR

Neste estudo de caso é demonstrado o uso e aplicabilidade do método de ranqueamento adotado neste trabalho. São utilizados os resultados das estratégias de DR analisados nos estudos de caso anteriores. Assim, é considerado o uso do método de ranqueamento algumas situações, uma vez que é possível ter diferentes combinações possíveis definir a função multicritério.

6.5.1 Resultados das estratégias de DR

Para facilitar a visualização dos dados, a Tabela 45 apresenta todos os resultados das estratégias de DR analisadas nos estudos de caso anteriores. Nessa tabela, as estratégias de DR são representadas por identificadores onde: #1 indica a estratégia de *backup*; #2 corresponde à estratégia de replicação de BD assíncrona; #3 equivale à estratégia de replicação de BD semisíncrona; #4 é a estratégia de replicação de ambiente *hot-standby*; e #5 representa a estratégia de replicação de ambiente *warm-standby*.

Tabela 45 – Resultados dos modelos DSPNs de todos os cenários analisados nos estudos de casos anteriores.

Estratégia	**Disponibilidade**	**Indisponibilidade (h/ano)**	**RTO (h)**	**RPO (h)**	**Custo (US$)**
#1	0,99770674	20,088957	16,92800000	24,1070000	14.447,74
#2	0,99843347	13,7228028	0,02117695	7,5449861E-05	10.443,54
#3	0,99841683	13,8685692	0,02190006	1,2990833E-07	10.535,22
#4	0,99996158	0,3365592	0,01666669	7,52902694E-05	3.641,87
#5	0,99986994	1,1393256	0,16231441	7,53839166E-05	4.069,51

6.5.2 Ranqueamento I

A primeira análise de ranqueamento considera uma situação na qual um(a) projetista (ou uma pessoa interessada na adoção de estratégias de DR) deseja identificar a estratégia que apresenta maior disponibilidade, menor RTO, menor RPO, e menor custo para a infraestrutura computacional. Assumindo que as infraestruturas e sistemas computacionais que se desejam avaliar sigam as especificações apresentadas nos estudos de casos anteriores, os resultados apresentados na Tabela 45 são usados como matriz de entrada no método de ranqueamento descrito na Seção 2.5.3. Para encontrar o *ranking* das estratégias seguindo os critérios definidos, é necessário definir no método de ranqueamento que quer: maximizar a disponibilidade, minimizar o RTO, RPO e custo e omitir a utilização da indisponibilidade. A Tabela 46 mostra o *ranking* das estratégias de DR seguindo os critérios definidos. Na tabela, também é apresentado o coeficiente de aproximação (CC_i) para cada estratégia de DR. O CC_i é um coeficiente que mede a relação entre as distâncias para a solução ideal positiva (melhor alternativa) e para a solução ideal negativa (pior alternativa) de

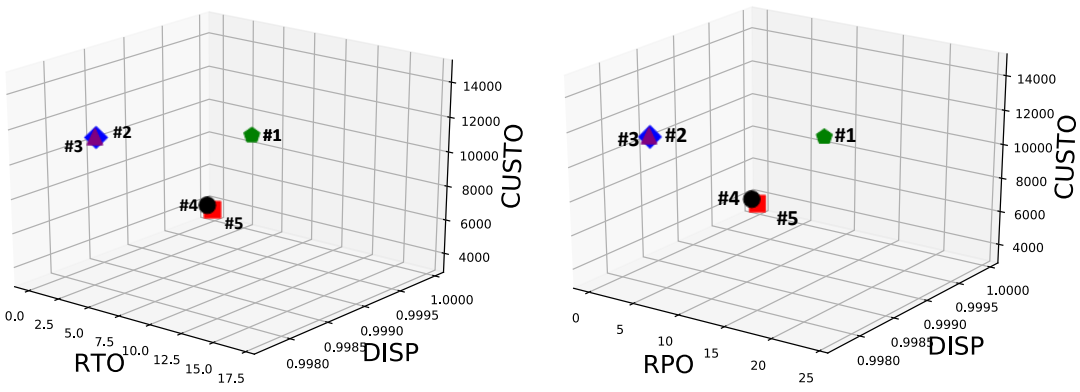
cada estratégia avaliada (ver Subseção 2.5.3). Quanto mais próximo de **1**, melhor é o desempenho global da alternativa.

Tabela 46 – *Ranking* das estratégias de DR considerando maximização da disponibilidade e minimização do RTO, do RPO e do custo.

Ranking	**Estratégia de DR**	**CC_i**
1	#4 - Replicação de ambiente (*hot*)	0.9999984411104058
2	#5 - Replicação de ambiente (*warm*)	0.9715012213092116
3	#2 - Replicação de BD (assínc.)	0.6180856037477563
4	#3 - Replicação de BD (semisínc.)	0.6146623956258632
5	#1 - Backup	0.0

Considerando os critérios definidos, as estratégias de replicação de ambiente apresentaram melhores resultados, onde a configuração *hot-standby* é a mais próxima à solução ideal, e a *warm-standby* é a segunda melhor. Em seguida, as estratégias de replicação de BD aparecem em terceiro e quarto lugar, respectivamente, para a configuração assíncrona e semisíncrona. Por fim, a estratégia de *backup* é a última colocada no *ranking*, tendo o CC_i com valor 0, sendo a estratégia que possui os piores critérios avaliados. O valor CC_i mostrado para cada estratégia deixa evidente que a estratégia de replicação de ambiente é a que melhor atende os critérios definidos. A Figura 37 (a) apresenta os resultados obtidos com as estratégias de DR de acordo com seus valores de RTO, disponibilidade e custo, enquanto a Figura 37 (b) mostra as mesmas estratégias de acordo com seus valores de RPO, disponibilidade e custo. Para atender aos critérios estabelecidos do ranqueamento considerado, as melhores alternativas devem se encontrar no canto inferior esquerdo do gráfico, uma vez que o local representa o valor máximo para a disponibilidade e os valores mínimos para o RTO, RPO e custo.

Figura 37 – Visão das estratégias de DR analisadas considerando RTO, disponibilidade e custo (a) e considerando RPO, disponibilidade e custo (b).



(a) RTO (h), disponibilidade e custo (US$) (b) RPO (h), disponibilidade e custo (US$)

Fonte: Criado pelo autor.

6.5.3 Ranqueamento II

O ranqueamento nesta subseção considera como critérios para seleção das estratégias de DR identificar a estratégia que apresenta menor RTO e menor RPO. Conforme o exemplo anterior, os resultados dos estudos de casos anteriores apresentados na Tabela 45 são usados como matriz de entrada no método de ranqueamento descrito na Seção 2.5.3. Dessa forma, os critérios definidos no método de ranqueamento devem ser: minimizar o RTO e RPO. Além disso, é necessário definir que as métricas de disponibilidade, indisponibilidade e custo serão omitidas.

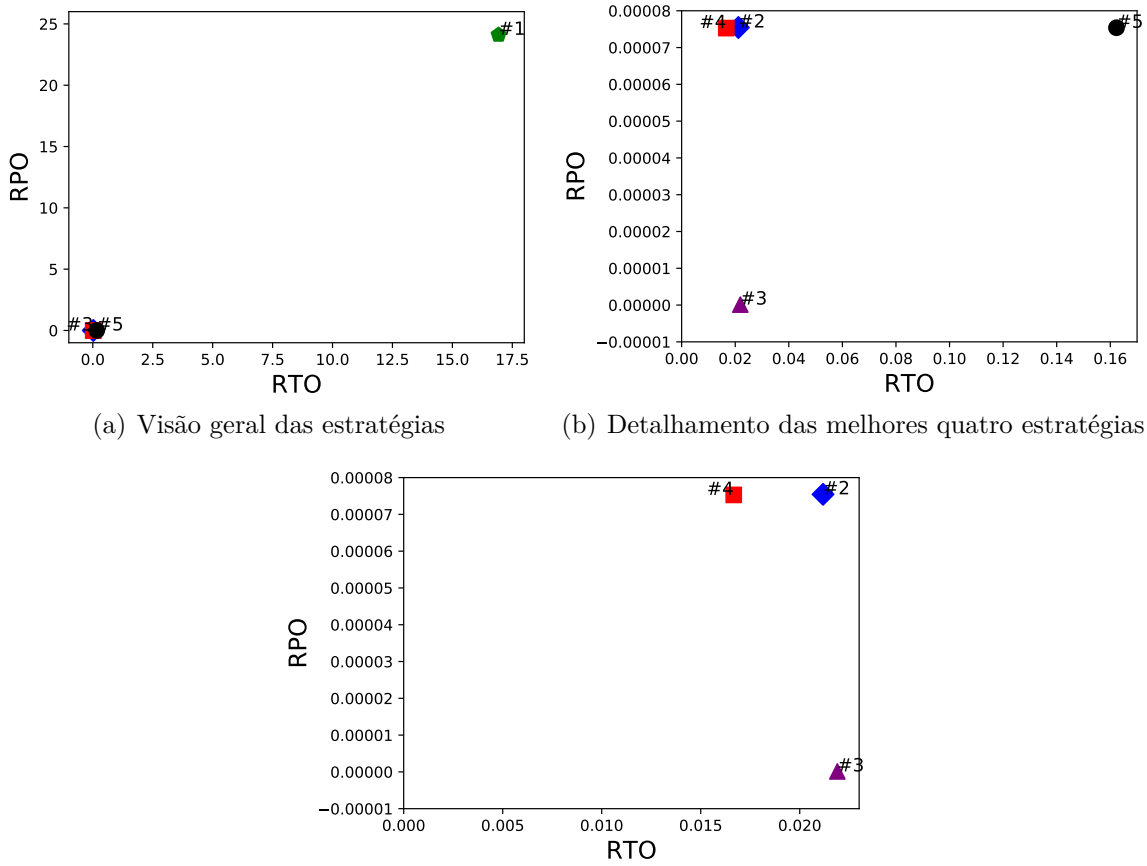
A Tabela 47 apresenta o *ranking* das estratégias de DR seguindo os critérios de definidos. Considerando esses critérios, a estratégia de replicação de ambiente utilizando a configuração *hot-standby* apresentou o melhor resultador. Em seguida, em segundo e terceiro lugar ficaram as estratégias de replicação de BD assíncrona e semisíncrona, respectivamente. Nas últimas posições ficaram as estratégias de replicação de ambiente utilizando a configuração *warm-standby* e o *backup*. Diferentemente do ranqueamento anterior (ver Subseção 6.5.2), é possível notar que pelo valor do CC_i das alternativas, com exceção da estratégia de *backup*, todas estão muito próximas. Esse tipo de situação explicita o benefício do uso do método de ranqueamento quando várias alternativas têm valores similares. Por exemplo, considerando a adoção de sistemas críticos, mínimas diferenças são cruciais. Dessa forma, o uso de um método de ranqueamento ajuda a perceber quantitativamente qual cenário apresenta vantagens sobre outros.

Tabela 47 – *Ranking* das estratégias de DR considerando minimização do RTO e do RPO.

Ranking	**Estratégia de DR**	$\mathbf{CC_i}$
1	#4 - Replicação de ambiente (*hot*)	0.9999977953968989
2	#2 - Replicação de BD (assínc.)	0.9998114114975516
3	#3 - Replicação de BD (semisínc.)	0.999781193230243
4	#5 - Replicação de ambiente (*warm*)	0.9939209802213982
5	#1 - Backup	0.0

Por último, as Figuras 38 (a), (b) e (c) exibem gráficos com as estratégias de DR de acordo com seus valores de RTO e RPO. Na Figura 38 (a), é exibida uma visão geral de todas as cinco estratégias. Nas Figuras 38 (b) e (c) é apresentada uma visão mais detalhada, contemplando as quatro e três melhores estratégias, respectivamente. Para atender aos critérios estabelecidos no ranqueamento, as melhores alternativas devem estar localizadas no canto inferior esquerdo do gráfico, uma vez que o local representa os valores mínimos para o RTO e para o RPO.

Figura 38 – Visão das estratégias de DR analisadas de acordo com RTO e RPO (em horas).



(a) Visão geral das estratégias

(b) Detalhamento das melhores quatro estratégias

(c) Detalhamento das melhores três estratégias

Fonte: Criado pelo autor.

6.5.4 Ranqueamento III

O ranqueamento realizado nesta subseção difere dos anteriores apresentados nas subseções 6.5.2 e 6.5.3. Aqui, o objetivo é identificar a melhor configuração para se aplicar a estratégia de *backup* de acordo com os critérios estabelecidos. Dessa forma, para realizar tal ranqueamento, são utilizados os resultados da análise multi-cenário da estratégia de *backup* (ver Subseção 6.2.4) apresentados na Tabela 54. Vale ressaltar que a análise multi-cenário da estratégia de *backup* considerou 12 cenários, onde foram variados os parâmetros de intervalo de *backup* (B_i) e quantidade de dados do *backup* (B_s). Os cenários foram definidos como apresentado na Tabela 48.

Para este ranqueamento é considerada a situação a qual se deseja selecionar a configuração da estratégia de *backup* que apresente maior disponibilidade, menor RTO e menor RPO. Os resultados apresentados na Tabela 54 são usados como matriz de entrada no método de ranqueamento descrito na Seção 2.5.3. Para selecionar os cenários de acordo com os critérios definidos, no método de ranqueamento é definido maximizar a disponibilidade, minimizar o RTO e o RPO. Além disso, é definida a omissão das métricas de indisponibilidade e custo. A Tabela 49 exibe o *ranking* dos cenários seguindo os critérios definidos.

Tabela 48 – Definição dos cenários na análise multi-cenário da estratégia de *backup*.

Cenário	B_i **(h)**	B_s **(GB)**
B1	12	100
B2	12	200
B3	12	300
B4	12	400
B5	24	100
B6	24	200
B7	24	300
B8	24	400
B9	48	100
B10	48	200
B11	48	300
B12	48	400

O cenário *B1* apresentou o melhor resultado. Nesse cenário, as operações de *backup* são realizadas a cada 12 horas ($B_i = 12$) e realizam o backup de 100GB de dados ($B_s = 100$). Em segundo e terceiro lugares ficaram os cenários *B5* e *B2*, respectivamente. É possível observar que mesmo entre os três primeiros cenários o coeficiente de aproximação (CC_i) apresenta valores com uma diferença significativa, mostrando que para os critérios definidos, o cenário *B1* apresenta uma vantagem considerável. Nas últimas posições, ficaram os cenários *B11* e *B12*, sendo o coeficiente de aproximação do cenário *B12* muito inferior ao penúltimo colocado *B11*.

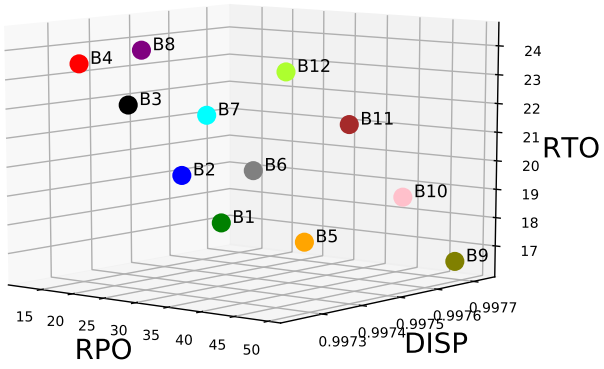
Tabela 49 – *Ranking* das estratégias de *backup* considerando maximização da disponibilidade, e minimização do RTO e do RPO.

Ranking	**Cenário**	$\mathbf{CC_i}$
1	B1	0.9501445351713852
2	B5	0.8300151135914209
3	B2	0.7419635077674627
4	B6	0.6629079420417158
5	B9	0.5887983171945038
6	B3	0.5017239204612373
7	B10	0.469667926124974
8	B7	0.46098384982588564
9	B4	0.3731486858299343
10	B8	0.28394542881275875
11	B11	0.26856078036474584
12	B12	0.035761831324592465

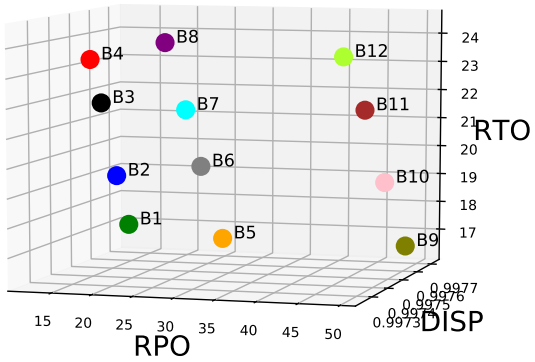
Para uma melhor visualização da disposição dos cenários quanto aos critérios utilizados

para esse ranqueamento, as Figuras 39 (a), (b) e (c) apresentam resultados dos cenários de acordo com os valores de disponibilidade, RTO e RPO. Todas as figuras apresentam os mesmos pontos, a partir de ângulos diferentes. Nesses gráficos, para atender aos critérios estabelecidos do ranqueamento, as melhores alternativas se encontram no canto inferior esquerdo da parte de trás do gráfico, sendo o local onde se encontram os valores máximos para a disponibilidade, e os valores mínimos para o RTO e para o RPO.

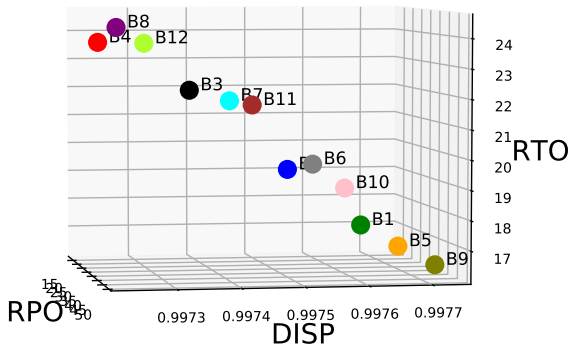
Figura 39 – Visão geral dos cenários analisados da estratégia de *backup* de acordo com o RTO (em horas), a disponibilidade e o RPO (em horas).



(a) Visão geral dos cenários 1



(b) Visão geral dos cenários 2



(c) Visão geral dos cenários 3

Fonte: Criado pelo autor.

6.6 CONSIDERAÇÕES FINAIS

Este capítulo apresentou possíveis aplicações para os modelos e método propostos nesta tese. Foram apresentados cinco estudos de caso que empregam os modelos DSPNs, modelo de custo e o método de seleção e ranqueamento das estratégias propostas. Cada estudo de caso abordou a avaliação de diferentes infraestruturas computacionais. Foi apresentado um estudo de caso onde nenhuma estratégia de DR foi adotada e estudos de caso onde foram consideradas as estratégias de *backup*, replicação de BD e replicação de ambiente. Através das avaliações conduzidas utilizando os modelos DSPNs propostos, foi possível observar os benefícios da adoção de estratégias de DR com relação a importantes métricas de DR, como a disponibilidade, a indisponibilidade, o RTO, o RPO e o custo. Em todas

as situações avaliadas, as infraestruturas computacionais que possuíam alguma estratégia de DR atrelada apresentaram melhores resultados se comparado à mesma infraestrutura sem uma estratégia de DR.

Também foi mostrado que, o uso das estratégias de DR apresenta um custo mais elevado quanto à operação, à manutenção e ao suporte da infraestrutura computacional. No entanto, quando também são considerados os custos de oportunidade (custo pela perda produção e/ou vendas pela infraestrutura computacional estar *offline*), até a estratégia de DR que possui um maior custo operacional se torna mais vantajosa. Além disso, alguns estudos de casos apresentaram os experimentos conduzidos em ambientes reais para avaliar as infraestruturas computacionais adotadas. Através desses experimentos foram calculados parâmetros de entrada mais próximos da realidade para os modelos DSPNs propostos. Também a partir dos experimentos foram verificadas as correspondências entre os resultados dos modelos propostos e os resultados que ocorrem em ambientes reais.

Por fim, no último estudo de caso, foi apresentado o uso do método de seleção e ranqueamento baseado em múltiplos critérios. Foram apresentados diferentes cenários nos quais o método de seleção e ranqueamento pode ser empregado para auxiliar na escolha de estratégias ou configurações de DR mais adequadas aos cenários em particular.

7 CONCLUSÕES E TRABALHOS FUTUROS

Falhas ou desastres podem comprometer a operação de empresas e o funcionamento de sistemas de TIC que não estão preparados para lidar com essas situações. A indisponibilidade dos sistemas ou a perda de dados decorrentes de tais falhas ou desastres podem ter consequências graves, como por exemplo, a insatisfação de clientes e perdas financeiras. Estratégias de DR têm sido empregadas em infraestruturas computacionais para mitigar os efeitos causados por falhas ou desastres inesperados. No entanto, um dos grandes desafios para a adoção de estratégias de DR, é a identificação de estratégias mais adequadas à realidade das empresas. Não há uma estratégia única que seja aplicável à todas as organizações e situações, pois cada organização apresenta diferentes características (ex.: mercado de atuação e faturamento) e necessidades. Além disso, cada estratégia de DR possui características próprias, sendo algumas dessas características mais apropriadas para algumas empresas e menos adequadas para outras.

Como apresentado no Capítulo 1, esta tese tem, como principal propósito, auxiliar empresas, projetistas ou pessoas interessadas na adoção de estratégias de DR a selecionar as melhores estratégias considerando diferentes critérios. Dessa forma, visando colaborar na resolução do problema estabelecido, esta tese apresentou, como sua principal contribuição, um conjunto de modelos e um método de decisão multicritério para avaliação e seleção de estratégias de DR. Foram propostos modelos para avaliar diferentes estratégias considerando métricas de DR relevantes como disponibilidade, indisponibilidade, RTO, RPO e custos. Além destes modelos, foi proposta a utilização de um método de decisão multicritério para identificar e selecionar as estratégias que mais se adéquam a critérios específicos. Estes critérios são definidos de acordo com a necessidade que cada empresa ou projetista tem em relação às métricas avaliadas das estratégias de DR. Além disso, esta tese também propôs uma metodologia para guiar o processo de avaliação e seleção das estratégias de DR utilizando os modelos e método propostos.

Com o intuito de demonstrar a aplicabilidade dos modelos, método e metodologia propostos, foram conduzidos um conjunto de estudos de casos. Inicialmente, foi avaliada uma infraestrutura computacional sem nenhuma estratégia de DR. Posteriormente, foi avaliada a mesma infraestrutura computacional adotando estratégias de DR distintas. Foram avaliadas as estratégias de *backup*, replicação de BD e replicação de ambiente. Além disso, também foram conduzidos experimentos em ambientes de testes a fim de obter parâmetros realísticos para os modelos propostos, bem como para verificar a corretude de tais modelos. Por fim, foi demonstrada a aplicação do método de decisão multicritério para auxiliar empresas e projetistas na seleção de estratégias ou cenários considerando diferentes situações e requisitos.

7.1 CONTRIBUIÇÕES

Podemos destacar, como principais contribuições desta tese:

- modelos probabilísticos foram propostos para representar e avaliar diferentes infraestruturas computacionais e estratégias de DR;
- utilizando os modelos probabilísticos propostos, foram realizadas análises em cenários com diferentes estratégias de DR considerando métricas de disponibilidade, indisponibilidade, RTO e RPO.
- de forma integrada, foram utilizados modelos de custos para estimar os valores de perda de produtividade e vendas, aquisição, operação e manutenção de infraestruturas computacionais, onde estratégias de DR são implementadas.
- um método de decisão multicritério foi utilizado de forma integrada com os resultados resultados dos modelos para auxiliar no processo decisório de seleção de estratégias de DR. O método considerou o uso de diferentes critérios, que podem ser definidos pelas empresas ou projetistas, e foram baseados nas métricas avaliadas pelos modelos probabilísticos e de custo.
- foi definida uma metodologia para descrever o fluxo das atividades necessárias para a execução das análises e seleção das estratégias de DR baseadas no uso dos modelos e método apresentados nesta tese.

Além das principais contribuições que foram citadas acima, durante o período de construção da pesquisa foram produzidos artigos científicos dentro do contexto e escopo da pesquisa. A Tabela 50 apresenta tais artigos, incluindo o capítulo desta tese a qual o artigo é relacionado. Já a Tabela 51 apresenta outros artigos produzidos durante o período de construção da pesquisa, mas que não estão diretamente relacionadas ao contexto desta tese.

Tabela 50 – Artigos científicos produzidos.

#	Artigo	Capítulo	Tipo
1	MENDONÇA, J.; ANDRADE, E.; ENDO, P. T.; LIMA, R. **Disaster recovery solutions for IT systems: A systematic mapping study**. Journal of Systems and Software, v. 149, p. 511–530, 2019.	3	Revista
2	MENDONÇA, J.; LIMA, R.; ANDRADE, E.; ARAÚJO, J.; & KIM, D. S. **Multiple-criteria Evaluation of Disaster Recovery Strategies Based on Stochastic Models**. In 16th International Conference on the Design of Reliable Communication Networks (DRCN), 2020.	4,5 e 6	Conferência
3	MENDONÇA, J.; LIMA, R.; MATOS, R.; FERREIRA, J.; & ANDRADE, E. **Availability analysis of a disaster recovery solution through stochastic models and fault injection experiments**. In IEEE 32nd International Conference on Advanced Information Networking and Applications (AINA), 2018.	6	Conferência
4	MENDONÇA, J.; LIMA, R.; QUEIROZ, E.; ANDRADE, E.; & KIM, D. S. **Evaluation of a backup-as-a-service environment for disaster recovery**. In 2019 IEEE Symposium on Computers and Communications (ISCC), 2019.	6	Conferência
5	QUEIROZ, E.; MENDONÇA, J.; CALLOU, G.; & ANDRADE, E. **Analisando o Backup-as-a-Service como uma Estratégia de Recuperação de Desastres**. In: IX Simpósio Brasileiro de Engenharia de Sistemas Computacionais (SBESC), 2019.	6	Conferência
6	MENDONÇA, J.; MEDEIROS, W.; ANDRADE, E; MACIEL, R; MACIEL, P; LIMA, R. **Evaluating Database Replication Mechanisms for Disaster Recovery in Cloud Environments**. In: IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2019.	6	Conferência
7	MEDEIROS, W.; MENDONÇA, J.; ALVES, G; & ANDRADE, E; **Avaliação Experimental de Replicação em Banco de Dados para Recuperação de Desastre**. In: Workshop em Desempenho de Sistemas Computacionais e de Comunicação (WPerformance), 2020.	6	Conferência
8	MENDONÇA, J.; ANDRADE, E; & LIMA, R. **Evaluating and Modelling Solutions for Disaster Recovery**. International Journal of Grid and Utility Computing, v.11, n.5, p. 683-704, 2020.	6	Revista

Tabela 51 – Outros artigos publicados.

#	Artigo	Tipo
1	MENDONÇA, J.; ANDRADE, E; LIMA, R. **Assessing mobile applications performance and energy consumption through experiments and Stochastic models**, Computing, v. 101, n. 12, p. 1789-1811, 2019.	Revista
2	MENDONÇA, J; CHO, J; J. MOORE, T; F. NELSON, F.; LIM, H; ZIMMERMANN, A; & KIM, D.S. **Performability analysis of services in a software-defined networking adopting time-based moving target defense mechanisms.** In: Proceedings of the 35th Annual ACM Symposium on Applied Computing (SAC '20), 2020.	Conferência

7.2 LIMITAÇÕES

As limitações desta tese são descritas a seguir:

- a principal limitação deste trabalho é que a geração dos modelos DSPNs propostos é feita de forma manual. Desta maneira, faz-se necessário que a(o) projetista que conduz o processo de avaliação e seleção das estratégias tenha conhecimento em modelagem analítica. Além disso, devido a tal aspecto, erros involuntários podem ser inseridos nos modelos, podendo acarretar em resultados inesperados ou não realísticos. A integração ou criação de uma ferramenta para gerar os modelos desenvolvidos a partir de modelos de alto nível poderia mitigar essa limitação e automatizar o processo de avaliação das infraestruturas e estratégias de DR.

- os estudos de casos apresentados neste trabalho focaram em infraestruturas computacionais para organizações de pequeno e médio porte. Organizações de grande porte chegam a utilizar milhares de componentes, como servidores, e máquinas virtuais em suas infraestruturas. Este tipo de cenário pode ser considerado em trabalhos futuros;

- caso os modelos propostos sejam empregados para avaliar infraestruturas de grande porte, com um número elevado de componentes, os modelos DSPNs propostos tendem a crescer o espaço de estados. Dessa forma, os modelos passam a não ser avaliados numericamente e precisam ser resolvidos somente por simulação, devido à explosão do espaço de estados. A utilização de modelagem hierárquica e modelos analíticos, como por exemplo o RBD, pode auxiliar na resolução dessa limitação em trabalhos futuros.

- este trabalho se limitou a estudar métricas relacionadas à dependabilidade de infraestruturas computacionais e estratégias de DR. Outras métricas relacionadas, como por exemplo a performabilidade, podem ser consideradas em trabalhos futuros.

- o modelo de custo considerou valores relativos à perda de produtividade e vendas, operação, manutenção e aquisição de componentes. Não foram consideradas outras abordagens existentes que consideram fatores como, por exemplo, tempo de contrato, composição heterogênea de componentes e infraestrutura compartilhada.

7.3 TRABALHOS FUTUROS

Como trabalhos futuros, pode-se incluir o desenvolvimento de modelos probabilísticos e/ou analíticos para outras estratégias de DR como, por exemplo, migração de VMs e deduplicação de dados. A inclusão de modelos para novas estratégias de DR pode ampliar a possibilidade de escolhas por parte das empresas. Junto a isso, também é importante a avaliação de cenários de larga escala, uma vez que esta tese focou em ambientes mais

modestos para pequenas e médias empresas. Podem ainda ser propostas alterações nos modelos (ou a criação de novos) para incluir a avaliação de métricas relacionadas à confiabilidade e performabilidade dos sistemas. A avaliação de tais métricas poderia fornecer mais detalhes acerca das características e das estratégias de DR. Contudo, a realização de análises para essas novas métricas podem requerer que novos parâmetros sejam informados ou coletados através de experimentos.

Outra possível extensão deste trabalho é a construção de uma ferramenta (ex.:um sistema web) que automatize a criação e avaliação dos modelos DSPNs. A automatização da criação poderia ser realizada a partir de modelos semiformais ou de alto nível, como por exemplo o trabalho realizado por Andrade (2014), que utilizou a SysML. A automatização da avaliação poderia ser realizada através do uso de *Application Programming Interface*s (APIs) de ferramentas já existentes e que são capazes de avaliar modelos DSPNs, como por exemplo a linguagem de *script* do Mercury (OLIVEIRA et al., 2017). A criação de tal ferramenta tornaria possível a realização do processo de avaliação das estratégias de DR por pessoas que não possuem (ou possuem pouco) conhecimento acerca de modelagem analítica.

Outra possível extensão para esta tese é o uso de técnicas de otimização, como algoritmos genéticos ou inteligência artificial para construir cenários considerando a adoção de diferentes estratégias de DR. Adotar algoritmos de otimização para este tipo de atividade é bastante factível, pois é possível trabalhar com uma grande gama de possibilidades e ainda ser possível encontrar o melhor cenário para as métricas de interesse estabelecidas. Um grande desafio para esta abordagem será diminuir o tempo de execução do algoritmo, uma vez que um grande número de cenários precisará ser analisado e alguns desses algoritmos requerem um tempo considerável de execução.

Por fim, uma possível extensão desta tese é a utilização e adaptação dos modelos para analisar ambientes em nuvem computacionais públicas e híbridas. Com a grande quantidade de recursos providos pelas nuvens públicas, muitas empresas têm adotado nuvens privadas ou híbridas para execução de seus sistemas e armazenamento dos dados. Dessa forma, é importante avaliar como desastres e falhas podem afetar sistemas que operam em tais ambientes. Uma análise quantitativa de métricas como disponibilidade, RTO, RPO e custos para comparação de estratégias de DR que operam em nuvens privadas, públicas ou híbridas pode ser de grande valor para as empresas.

REFERÊNCIAS

AALST, W. van der; HEE, K. van; REIJERS, H. Analysis of discrete-time stochastic petri nets. *Statistica Neerlandica*, v. 54, n. 2, p. 237–255, 2000.

ALEXANDER, P. *Information Security: A Manager's Guide to Thwarting Data Thieves and Hackers*. [S.l.]: ABC-CLIO, 2008. (PSI Business Security). ISBN 9780313345586.

ALHAZMI, O. H.; MALAIYA, Y. K. Assessing disaster recovery alternatives: On-site, colocation or cloud. *Proceedings - 23rd IEEE International Symposium on Software Reliability Engineering Workshops, ISSREW 2012*, p. 19–20, 2012.

ALMEIDA, A. de; CAVALCANTE, C.; ALENCAR, M.; FERREIRA, R.; ALMEIDA-FILHO, A. de; GARCEZ, T. *Multicriteria and Multiobjective Models for Risk, Reliability and Maintenance Decision Analysis*. [S.l.]: Springer International Publishing, 2015. (International Series in Operations Research & Management Science). ISBN 9783319179698.

ALSHAMMARI, M.; ALWAN, A.; ALSHAIKHLI, I. Data recovery and business continuity in Cloud computing: A Review of the Research Literature. *International Journal of Advancements in Computing Technology*, Vol.8, p. 80, 2016.

ANDRADE, E.; NOGUEIRA, B. Dependability evaluation of a disaster recovery solution for IoT infrastructures. *The Journal of Supercomputing*, mar 2018. ISSN 0920-8542.

ANDRADE, E.; NOGUEIRA, B. Performability Evaluation of a Cloud-Based Disaster Recovery Solution for IT Environments. *Journal of Grid Computing*, Springer, v. 17, n. 3, p. 603–621, sep 2019. ISSN 1570-7873.

ANDRADE, E.; NOGUEIRA, B.; MATOS, R.; CALLOU, G.; MACIEL, P. Availability modeling and analysis of a disaster-recovery-as-a-service solution. *Computing*, Springer Vienna, p. 1–26, feb 2017.

ANDRADE, E. C. *Modelagem e Análise de Mecanismos de Tratamento de Interrupções em Infraestruturas Computacionais dos Sistemas Distribuídos*. 138 p. Tese (Doutorado) — Universidade Federal de Pernambuco - UFPE, Recife, Pernambuco, 2014.

Anh Nguyen, T.; HAN, K.; MIN, D.; CHOI, E.; Duc Thang, T.; CHOI, Y.-J.; Tuan Anh Nguyen, B. A stochastic reward net-based assessment of reliability, availability and operational cost for a software-defined network infrastructure. *The Journal of Supercomputing*, v. 75, p. 4657–4683, 2019.

APACHE. *Apache HTTP Server 2.2.34 Released*. 2017. [Online]. Disponível em: <https://www.apache.org/dist/httpd/Announcement2.2.html>. Acessado em: 15/08/2017.

ARAUJO, C. J. M. *Tomada de Decisão Multicritério em Infraestruturas como Serviço em Nuvem: Uma Abordagem baseada em Modelos de Dependabilidade, Performabilidade e Custo*. 176 p. Tese (Doutorado) — Federal University of Pernambuco, 2019.

ARESE, M.; RANGEL, L. A.; HALL, J.; ZOTES, L.; BONINA, N.; MEIRIñO, M. Aplicação do método topsis na avaliação dos critérios utilizados na seleção de docentes em uma instituição de ensino superior. *Conhecimento & Diversidade*, v. 9, n. 19, p. 47–58, 2018. ISSN 2237-8049. Disponível em: <https://revistas.unilasalle.edu.br/index.php/conhecimento_diversidade/article/view/3906>.

ARLAT, J.; COSTES, A.; CROUZET, Y.; LAPRIE, J. C.; POWELL, D. Fault injection and dependability evaluation of fault-tolerant systems. *IEEE Transactions on Computers*, v. 42, n. 8, p. 913–923, 1993.

AVIZIENIS, A.; LAPRIE, J. C.; RANDELL, B.; LANDWEHR, C. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing*, v. 1, n. 1, p. 11–33, Jan 2004.

Bacula Systems. *Corporate Data Backup and Recovery Features in Bacula Enterprise Edition*. 2018. Disponível em: <https://www.baculasystems.com/architecture>.

BADGER, L.; GRANCE, T.; PATT-CORNER, R.; VOAS, J. *Cloud Computing Synopsis and Recommendations*. 2012. [Online]. Disponível em: <http://csrc.nist.gov/publications/nistpubs/800-146/sp800-146.pdf> Acessado em: 12/01/2019.

Bai, J.; Chang, X.; Machida, F.; Trivedi, K. S.; Han, Z. Analyzing software rejuvenation techniques in a virtualized system: Service provider and user views. *IEEE Access*, v. 8, p. 6448–6459, 2020. ISSN 2169-3536.

BAUER, E.; ADAMS, R.; EUSTACE, D. *Beyond Redundancy: How Geographic Redundancy Can Improve Service Availability and Reliability of Computer-Based Systems*. [S.l.]: Wiley, 2011.

BENZ, K.; BOHNERT, T. M. Impact of pacemaker failover configuration on mean time to recovery for small cloud clusters. In: *2014 IEEE 7th International Conference on Cloud Computing*. [S.l.: s.n.], 2014. p. 384–391.

BEZERRA, M. C.; MELO, R.; DANTAS, J.; MACIEL, P.; VIEIRA, F. Availability modeling and analysis of a vod service for eucalyptus platform. In: *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.: s.n.], 2014. p. 3779–3784.

CASSANDRAS, C.; LAFORTUNE, S. *Introduction to Discrete Event Systems*. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010. 800 p. ISBN 1441941193.

CHARETTE, R. N. *The Biggest IT Failures of 2018*. 2018. **IEEE Spectrum's risk analysis blog.** Disponível em: <https://spectrum.ieee.org/riskfactor/computing/it/it-failures-2018-all-the-old-familiar-faces> Acesso em: 20/03/2019.

CHERVENAK, A.; VELLANKI, V.; KURMAS, Z. Protecting file systems: A survey of backup techniques. In: *Joint NASA and IEEE Mass Storage Conference*. [S.l.: s.n.], 1998. v. 99.

DANTAS, J.; MATOS, R.; ARAUJO, J.; MACIEL, P. Eucalyptus-based private clouds: availability modeling and comparison to the cost of a public cloud. *Computing*, v. 97, n. 11, p. 1121–1140, Nov 2015.

DANTAS, J. R. *Planejamento de infraestrutura de nuvens computacionais para serviço de VoD streaming considerando desempenho, disponibilidade e custo*. 125 p. Tese (Doutorado) — Universidade Federal de Pernambuco, 2018.

DINH, H. T.; LEE, C.; NIYATO, D.; WANG, P. A survey of mobile cloud computing: architecture, applications, and approaches. *Wireless Communications and Mobile Computing*, v. 13, n. 18, p. 1587–1611, 2011. ISSN 1530-8677. Disponível em: <http://dx.doi.org/10.1002/wcm.1203>.

EATON. *Eaton Downtime Cost Calculator*. 2018. [Online]. <http://powerquality.eaton.com/Products-services/Help-Me-Choose/DowntimeCostCalculator/Default.asp>.

FÁVERO, L. P. L.; BELFIORE, P. P.; SILVA, F. L. d.; CHAN, B. L. *Análise de dados: modelagem multivariada para tomada de decisões*. [S.l.]: Elsevier, 2009.

Ferreira, L.; da Silva Rocha, E.; Monteiro, K. H. C.; Santos, G. L.; Silva, F. A.; Kelner, J.; Sadok, D.; Filho, C. J. A. B.; Rosati, P.; Lynn, T.; Endo, P. T. Optimizing resource availability in composable data center infrastructures. In: *2019 9th Latin-American Symposium on Dependable Computing (LADC)*. [S.l.: s.n.], 2019. p. 1–10.

FILHO, M. C. S.; OLIVEIRA, R. L.; MONTEIRO, C. C.; INáCIO, P. R. M.; FREIRE, M. M. Cloudsim plus: A cloud computing simulation framework pursuing software engineering principles for improved modularity, extensibility and correctness. In: *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*. [S.l.: s.n.], 2017. p. 400–406.

FLYNT, C.; LAKSHMAN, S.; TUSHAR, S. *Linux Shell Scripting Cookbook*. [S.l.]: Packt Publishing, 2017. ISBN 9781785882388.

FONTELA, E.; GABUS, A. *The DEMATEL observer: Dematel 1976 Report*. 1976.

GERMAN, R. *Performance Analysis of Communication Systems with Non-Markovian Stochastic Petri Nets*. New York, NY, USA: John Wiley & Sons, Inc., 2000. ISBN 0471492582.

GHOSH, R.; LONGO, F.; FRATTINI, F.; RUSSO, S.; TRIVEDI, K. S. Scalable analytics for iaas cloud availability. *IEEE Transactions on Cloud Computing*, v. 2, n. 1, p. 57–70, Jan 2014.

GOOGLE. *Google Cloud*. 2020. [Online]. <https://cloud.google.com>.

GOPISETTY, S.; BUTLER, E.; JAQUET, S.; KORUPOLU, M.; NAYAK, T. K.; ROUTRAY, R.; SEAMAN, M.; SINGH, A.; TAN, C.-H.; UTTAMCHANDANI, S.; VERMA, A. Automated planners for storage provisioning and disaster recovery. *IBM Journal of Research and Development*, v. 52, n. 4.5, p. 353–365, jul 2008.

GUEDES, E. A. C. *AVAILABILITY AND CAPACITY MODELING FOR VIRTUAL NETWORK FUNCTIONS BASED ON REDUNDANCY AND REJUVENATION SUPPORTED THROUGH LIVE MIGRATION*. 154 p. Tese (Doutorado) — Federal University of Pernambuco, 2019.

HAIR, J.; BLACK, W.; BABIN, B.; ANDERSON, R.; TATHAM, R. *Análise multivariada de dados - 6ed*. [S.l.]: Bookman, 2009. ISBN 9788577805341.

HALILI, E. H. *Apache JMeter: A practical beginner's guide to automated testing and performance measurement for your websites*. [S.l.]: Packt Publishing Ltd, 2008.

HAPROXY. *HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer*. 2017. [Online]. Disponível em: <http://www.haproxy.org/>. Acessado em: 15/08/2017.

HWANG, C.; YOON, K. *Multiple attribute decision making: methods and applications : a state-of-the-art survey*. [S.l.]: Springer-Verlag, 1981. (Lecture notes in economics and mathematical systems).

INFORMATION TECHNOLOGY INTELLIGENCE CONSULTING CORP. *ITIC 2019 Global Server Hardware, Server OS Reliability Report*. [S.l.], 2019. [Online]. Disponível em: <https://www.lenovo.com/us/en/resources/data-center-solutions/analyst-reports/itic-2019-global-server-hardware-server-os-reliability-report/> Acesso em: 25/11/2019.

JAIN, A. K.; DUBES, R. C. *Algorithms for Clustering Data*. USA: Prentice-Hall, Inc., 1988. ISBN 013022278X.

JAIN, R. *The Art of Computer Systems Performance Analysis*. New York, NY, USA: John Wiley & Sons, 1991.

JAISWAL, V.; SEN, A.; VERMA, A. Rscmap: resiliency planning in storage clouds. In: *Proceedings of the 9th International Conference on Service-Oriented Computing*. Paphos, Cyprus: Springer-Verlag, 2011. p. 505–512.

JANOUSEK, V. *Modelling Objects by Petri Nets*. Tese (Doutorado) — Brno University of Technology, 1998.

JATOTH, C.; GANGADHARAN, G. R.; FIORE, U.; BUYYA, R. Selcloud: A hybrid multi-criteria decision-making model for selection of cloud services. *Soft Comput.*, Springer-Verlag, Berlin, Heidelberg, v. 23, n. 13, p. 4701–4715, jul. 2019. ISSN 1432-7643.

JENSEN, K. Coloured petri nets: A high level language for system design and analysis. In: ROZENBERG, G. (Ed.). *Advances in Petri Nets 1990*. Springer Berlin Heidelberg, 1991, (Lecture Notes in Computer Science, v. 483). p. 342–416. ISBN 978-3-540-53863-9. Disponível em: <http://dx.doi.org/10.1007/3-540-53863-1_31>.

JENSEN, K. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use, Vol. 2*. London, UK, UK: Springer-Verlag, 1995. ISBN 3-540-58276-2.

JOHNSON, T.; MARGALHO, M. Avaliação de desempenho de sistemas computacionais. *Rio de Janeiro: LTC*, 2011.

KAHRAMAN, C.; CEBECI, U.; ULUKAN, Z. Multi-criteria supplier selection using fuzzy ahp. *Logistics Information Management*, Emerald, v. 16, n. 6, p. 382–394, dec 2003. ISSN 0957-6053.

KEESEE, W. R. *A Method of Determining a Confidence Interval for Availability*. Point Mugu, CA: Naval Missile Center, 1965.

KEETON, K.; SANTOS, C.; BEYER, D.; CHASE, J.; WILKES, J. Designing for disasters. In: *Proceedings of the 3rd USENIX Conference on File and Storage Technologies*. Berkeley, CA, USA: USENIX Association, 2004. (FAST '04). Disponível em: <http://dl.acm.org/citation.cfm?id=1096673.1096682>.

KIM, D. S.; MACHIDA, F.; TRIVEDI, K. Availability modeling and analysis of a virtualized system. In: *Dependable Computing, 2009. PRDC '09. 15th IEEE Pacific Rim International Symposium on*. [S.l.: s.n.], 2009. p. 365–371.

KITCHENHAM, B.; CHARTERS, S. *Guidelines for performing Systematic Literature reviews in Software Engineering Version 2.3*. Keele, Staffs, 2007. 65 p.

KUMAR, R. R.; MISHRA, S.; KUMAR, C. A novel framework for cloud service evaluation and selection using hybrid mcdm methods. *Arabian Journal for Science and Engineering*, Springer, v. 43, n. 12, p. 7015–7030, 2018.

KUO, W.; ZUO, M. *Optimal Reliability Modeling: Principles and Applications*. [S.l.]: Wiley, 2003.

LAPRIE, J. *Dependability: basic concepts and terminology in English, French, German, Italian, and Japanese*. [S.l.]: Springer-Verlag, 1992. (Dependable computing and fault-tolerant systems).

LENK, A.; PALLAS, F. *Modeling Quality Attributes of Cloud-Standby-Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. 49–63 p.

LENK, A.; TAI, S. Cloud standby: Disaster recovery of distributed systems in the cloud. In: *Service-Oriented and Cloud Computing: Third European Conference, ESOCC 2014*. Manchester, UK: Springer, Berlin, Heidelberg, 2014. p. 32–46.

LI, X.; LIU, Y.; KANG, R.; XIAO, L. Service reliability modeling and evaluation of active-active cloud data center based on the {it} infrastructure. *Microelectronics Reliability*, 2017.

LIMA, F. R.; CARPINETTI, L. C. R. Uma comparação entre os métodos topsis e fuzzy-topsis no apoio à tomada de decisão multicritério para seleção de fornecedores. *Gestao e Producao*, Brazilian Institute for Information in Science and Technology, v. 22, n. 1, p. 17–34, jan 2015. ISSN 18069649.

LITTLE, J. D. C. A proof for the queuing formula: L = w. *Oper. Res.*, INFORMS, Linthicum, MD, USA, v. 9, n. 3, p. 383–387, jun. 1961. ISSN 0030-364X. Disponível em: <https://doi.org/10.1287/opre.9.3.383>.

LIU, B.; CHANG, X.; HAN, Z.; TRIVEDI, K.; RODRíGUEZ, R. J. Model-based sensitivity analysis of iaas cloud availability. *Future Generation Computer Systems*, v. 83, p. 1 – 13, 2018. ISSN 0167-739X. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0167739X17301796>.

LIU, S.; CHAN, F. T.; RAN, W. Decision making for the selection of cloud vendor: An improved approach under group decision-making with integrated weights and objective/subjective attributes. *Expert Systems with Applications*, v. 55, p. 37 – 47, 2016. ISSN 0957-4174.

LOGICMONITOR. *IT Outage Impact Study*. [S.l.], 2019. [Online]. Disponível em: <https://www.logicmonitor.com/it-outage-impact-survey/>.

LOWE, S. D.; DAVIS, D. M.; GREEN, J. *2016 Disaster Recovery as a Service Attitudes & Adoption Report*. [S.l.], 2016. 23 p. [Online]. Disponível em: <https://www.infrascale.com/wp-content/uploads/pdf/2016-Disaster-Recovery-as-a-Service-Attitudes-and-Adoption-Report.pdf>.

MACHIDA, F.; ANDRADE, E.; KIM, D. S.; TRIVEDI, K. S. Candy: Component-based availability modeling framework for cloud service management using sysml. In: *2011 IEEE 30th International Symposium on Reliable Distributed Systems*. [S.l.: s.n.], 2011. p. 209–218.

MACIEL, P. R.; TRIVEDI, K. S.; MATIAS, R.; KIM, D. S. Dependability modeling. In: *Performance and Dependability in Service Computing: Concepts, Techniques and Research Directions*. [S.l.]: IGI Global, 2012. p. 53–97.

MACIEL, P. R. M.; LINS, R. D.; CUNHA, P. R. *Introduçao às redes de Petri e aplicações*. [S.l.]: UNICAMP - Instituto de Computação, 1996.

MALHOTRA, M.; TRIVEDI, K. S. Power-hierarchy of dependability-model types. *IEEE Transactions on Reliability*, v. 43, n. 3, p. 493–502, Sep 1994.

MANPAGEZ. *BSD System Manager's Manual*. 2019. [Online]. Disponível em: <http://www.manpagez.com/man/8/ping/>. Acesso em: 05/12/2019.

MARSAN, M.; CHIOLA, G. On petri nets with deterministic and exponentially distributed firing times. In: *Advances in Petri Nets 1987*. [s.n.], 1987. v. 266, p. 132–145. ISBN 978-3-540-18086-9. Disponível em: <http://dx.doi.org/10.1007/3-540-18086-9_23>.

MARSAN, M. A. Stochastic petri nets: an elementary introduction. *Advances in Petri Nets 1989*, p. 1–29, 1990.

MARSAN, M. A.; BALBO, G.; CONTE, G.; DONATELLI, S.; FRANCESCHINIS, G. *Modelling with Generalized Stochastic Petri Nets*. 1st. ed. USA: John Wiley & Sons, Inc., 1994. ISBN 0471930598.

MATOS, R.; ANDRADE, E. C.; MACIEL, P. Evaluation of a disaster recovery solution through fault injection experiments. In: *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.: s.n.], 2014. p. 2675–2680.

MAZILU, M. C. Database Replication. *Database Systems Journal*, n. 2, 2010.

MEDINA, A. *2019 in Review: The Biggest Internet Outages of the Year*. 2020. **Network Computing.** [Online]. Disponível em: <https://www.networkcomputing.com/network-security/2019-review-biggest-internet-outages-year> Acesso em: 21/01/2020.

MELO, R. D.; BEZERRA, M. C.; DANTAS, J.; MATOS, R.; FILHO, I.; MACIEL, P. Redundant vod streaming service in a private cloud: Availability modeling and sensitivity analysis. In: *Mathematical Problems in Engineering*. [S.l.: s.n.], 2014. v. 2014, p. 14.

MENDONçA, J.; ANDRADE, E.; ENDO, P. T.; LIMA, R. Disaster recovery solutions for it systems: A systematic mapping study. *Journal of Systems and Software*, v. 149, p. 511 – 530, 2019. ISSN 0164-1212.

MENDONçA, J.; LIMA, R.; QUEIROZ, E.; ANDRADE, E.; KIM, D. S. Evaluation of a backup-as-a-service environment for disaster recovery. In: *2019 IEEE Symposium on Computers and Communications (ISCC)*. [S.l.: s.n.], 2019.

MERLIN, P.; FARBER, D. J. Recoverability of communication protocols–implications of a theoretical study. *Communications, IEEE Transactions on*, v. 24, n. 9, p. 1036–1043, Sep 1976. ISSN 0090-6778.

MICROSOFT. *Recurso de IO: Proteção e Recuperação de Dados - Básico para Padronizado*. 2018. [Online]. Disponível em: <https://technet.microsoft.com/pt-br/library/bb821259.aspx>. Acessado em: 21/01/2018.

MICROSOFT TECHNET. *Selecting the Appropriate Type of Replication*. 2018. [Online]. Disponível em: <https://technet.microsoft.com/en-us/library/ms152565(v=sql.105).aspx>. Acessado em: 12/01/2018.

MURATA, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, vol. 77, n. No. 4, p. 541–580, 1989.

NGUYEN, T. A.; KIM, D. S.; PARK, J. S. Availability modeling and analysis of a data center for disaster tolerance. *Future Generation Computer Systems*, Elsevier B.V., v. 56, p. 27–50, 2016.

OLIVEIRA, D.; DANTAS, J.; ROSA, N.; MACIEL, P.; MATOS, R.; BRINKMANN, A. A dependability and cost optimisation method for private cloud infrastructures. *International Journal of Web and Grid Services*, v. 15, n. 4, p. 367–393, 2019. Disponível em: <https://www.inderscienceonline.com/doi/abs/10.1504/IJWGS.2019.103222>.

OLIVEIRA, D.; MATOS, R.; DANTAS, J.; FERREIRA, J.; SILVA, B.; CALLOU, G.; MACIEL, P.; BRINKMANN, A. Advanced Stochastic Petri Net Modeling with the Mercury Scripting Language. In: *Proceedings of the 11th EAI International Conference on Performance Evaluation Methodologies and Tools*. [S.l.: s.n.], 2017.

OLIVEIRA, D. M. *Análise de Disponibilidade e Consumo Energético em Ambientes de Mobile Cloud Computing*. Dissertação (Mestrado) — Universidade Federal de Pernambuco - UFPE, Recife, Pernambuco, 2014.

OMG. *UML Profile for MARTE: Modeling and Analysis of Real-Time Embedded Systems v.1.1*. 2011.

OMG. *Systems modeling language (sysml) specification*. 2012.

OPRICOVIC, S. *Multicriteria optimization of civil engineering systems*. Tese (Doutorado) — Faculty of Civil Engineering, Belgrade, 1998.

ORACLE. *MySQL 5.6*. 2017. [Online]. Disponível em: <https://dev.mysql.com/doc/refman/5.6/en/>. Acessado em: 15/08/2017.

Oracle. *The Binary Log*. 2020. [Online]. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/binary-log.html>. Acessado em: 15/02/2020.

ORACLE. *MySQL 8*. 2020. [Online]. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/>. Acessado em: 15/05/2020.

Oracle. *Performance Schema Quick Start*. 2020. [Online]. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/performance-schema-quick-start.html>.

Oracle. *Primary-Secondary Replication*. 2020. [Online]. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/group-replication-primary-secondary-replication.html>. Acessado em: 12/01/2020.

Oracle. *The Slave Relay Log*. 2020. [Online]. Disponível em: <https://dev.mysql.com/doc/refman/8.0/en/slave-logs-relaylog.html>. Acessado em: 15/02/2020.

ORACLE. *The BLOB and TEXT Types*. 2020. [Online]. <https://dev.mysql.com/doc/refman/8.0/en/blob.html>.

PADILHA, V. A.; CARVALHO, A. C. P. de Leon Ferreira de. *Mineração de dados em Python*. [S.l.]: ICMC - Universidade de São Paulo, 2017.

PARHAMI, B. From defects to failures: A view of dependable computing. *SIGARCH Comput. Archit. News*, ACM, New York, NY, USA, v. 16, n. 4, set. 1988. ISSN 0163-5964.

PETRI, C. A. *Kommunikation mit Automaten*. Tese (Doutorado) — Universität Hamburg, Hamburg, Alemanha, 1962.

PORTNOY, M. *Virtualization Essentials*. [S.l.]: Wiley, 2012. (Essentials Series). ISBN 9781118176719.

PRESTON, W. C. *Backup & recovery: inexpensive backup solutions for open systems*. [S.l.]: "O'Reilly Media, Inc.", 2007.

REESE, G. *Cloud Application Architectures: Building Applications and Infrastructure in the Cloud*. [S.l.]: O'Reilly Media, Inc., 2009. ISBN 0596156367, 9780596156367.

REGUNATHAN, R.; MURUGAIYAN, A.; LAVANYA, K. A qos-aware hybrid topsis–plurality method for multi-criteria decision model in mobile cloud service selection. In: SPRINGER. *Proceedings of the 2nd International Conference on Data Engineering and Communication Technology*. [S.l.], 2019. p. 499–507.

ROONEY, W. J.; MCBRIDE, G. E.; HANIF, T. Ibm totalstorage productivity center for replication for z/os. *IBM Systems Journal*, v. 47, n. 4, p. 681–694, 2008.

ROY, B. *Multicriteria Methodology for Decision Aiding*. [S.l.]: Springer US, 2013. (Nonconvex Optimization and Its Applications). ISBN 9781475725001.

SAATY, T. *Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World*. [S.l.]: RWS Publications, 1990. ISBN 9781888603132.

SAATY, T. *Theory and Applications of the Analytic Network Process: Decision Making With Benefits, Opportunities, Costs, and Risks*. [S.l.]: RWS Publications, 2005. ISBN 9781888603163.

SAHNER, R.; TRIVEDI, K.; PULIAFITO, A. *Performance and Reliability Analysis of Computer Systems: An Example-Based Approach Using the SHARPE Software Package*. [S.l.]: Springer US, 2012. ISBN 9781461523673.

SAHNER, R.; TRIVEDI, K. S.; PULIAFITO, A. Performance and reliability analysis of computer systems (an example-based approach using the sharpe software. *IEEE Transactions on Reliability*, v. 46, n. 3, p. 441–441, Sep 1997.

SIFAKIS, J. Use of petri nets for performance evaluation. In: *Proceedings of the Third International Symposium on Measuring, Modelling and Evaluating Computer Systems*. Amsterdam, The Netherlands, The Netherlands: North-Holland Publishing Co., 1977. p. 75–93. ISBN 0-444-85058-9. Disponível em: <http://dl.acm.org/citation.cfm?id=647408.724424>.

SILVA, B. *A Framework for Availability, Performance and Survivability Evaluation of Disaster Tolerant Cloud Computing Systems*. 150 p. Tese (Doutorado) — Federal University of Pernambuco, 2016.

SILVA, B.; MATOS, R.; CALLOU, G.; FIGUEIREDO, J.; OLIVEIRA, D.; FERREIRA, J.; DANTAS, J.; LOBO, A.; ALVES, V.; MACIEL, P. 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*. [S.l.: s.n.], 2015.

SOUSA, E.; SILVA, E.; LINS, F.; TAVARES, E.; MACIEL, P. Dependability evaluation of cloud infrastructures. In: *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. [S.l.: s.n.], 2014. p. 1282–1287.

SOUSA, E. T. G. D.; LINS, F. A. A.; TAVARES, E. A. G.; MACIEL, P. R. M. Performance and cost modeling strategy for cloud infrastructure planning. In: *2014 IEEE 7th International Conference on Cloud Computing*. [S.l.: s.n.], 2014. p. 546–553.

SOUZA, D.; MATOS, R.; ARAUJO, J.; ALVES, V.; MACIEL, P. Eucabomber: Experimental evaluation of availability in eucalyptus private clouds. In: *2013 IEEE International Conference on Systems, Man, and Cybernetics*. [S.l.: s.n.], 2013.

SWANSON, M.; BOWEN, P.; PHILLIPS, A.; GALLUP, D.; LYNES, D. *SP 800-34 Rev. 1. Contingency Planning Guide for Federal Information Systems*. 2010. [Online]. Disponível em: <https://csrc.nist.gov/publications/detail/sp/800-34/rev-1/final> Acessado em: 12/01/2018.

TRIVEDI, K.; GROTTKE, M.; ANDRADE, E. Software fault mitigation and availability assurance techniques. *International Journal of Systems Assurance Engineering and Management*, v. 1, p. 340–350, 04 2010.

TRIVEDI, K. S. *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1982. ISBN 0137115644.

TRIVEDI, K. S.; SUN, H.; CAO, Y.; MA, Y. Stochastic petri nets and their applications. In: GOTO, K.; HASEGAWA, T.; TAKAGI, H.; TAKAHASHI, Y. (Ed.). *Performance and QoS of Next Generation Networking*. London: Springer London, 2001. p. 283–298. ISBN 978-1-4471-0705-7.

UNITRENDS. *The State of Cloud and Data Protection 2018*. [S.l.], 2018. Disponível em: <https://bit.ly/2VsiieB>.

Unix Tutorial. *dd command*. 2019. [Online]. Disponível em: <https://www.unixtutorial.org/commands/dd>. Acesso em: 13/03/2020.

XIA, R.; YIN, X.; Alonso Lopez, J.; MACHIDA, F.; TRIVEDI, K. S. Performance and Availability Modeling of ITSystems with Data Backup and Restore. *IEEE Transactions on Dependable and Secure Computing*, v. 11, n. 4, p. 375–389, jul 2014. ISSN 1545-5971.

XIE, M.; DAI, Y.; POH, K. *Computing System Reliability: Models and Analysis*. [S.l.]: Springer US, 2004. (Cell Engineering Series).

YANG, C.-L.; HUANG, C.-Y.; KAO, Y.-S.; TASI, Y.-L. Disaster recovery site evaluations and selections for information systems of academic big data. *Eurasia Journal of Mathematics, Science and Technology Education*, v. 13, n. 8, p. 4553–4589, 2017. ISSN 1305-8215.

YOO, S.-K.; KIM, B.-Y. A decision-making model for adopting a cloud computing system. *Sustainability*, Multidisciplinary Digital Publishing Institute, v. 10, n. 8, p. 2952, 2018.

ZIMMERMANN, A.; BODE, S.; HOMMEL, G. Performance and dependability evaluation of manufacturing systems using petri nets. In: *in: 1st Workshop on Manufacturing Systems and Petri Nets, 17th Int. Conf. on Application and Theory of Petri Nets (Osaka*. [S.l.: s.n.], 1996. p. 235–250.

ZIMMERMANN, A.; KNOKE, M. *TimeNet 4.0: A Software Tool for the Performability Evaluation with Stochastic and Colored Petri Nets. User Manual*. Techn. Univ., Fak. IV, Elektrotechnik und Informatik, 2007. (Forschungsberichte der Fakultät IV - Elektrotechnik und Informatik // Technische Universität Berlin). Disponível em: <https://books.google.com.br/books?id=bCrgygAACAAJ>.

ZIONTS, S. Mcdm—if not a roman numeral, then what? *INFORMS Journal on Applied Analytics*, v. 9, n. 4, p. 94–101, 1979. Disponível em: <https://doi.org/10.1287/inte.9.4.94>.

ZOPOUNIDIS, C.; PARDALOS, P. *Handbook of Multicriteria Analysis*. Springer Berlin Heidelberg, 2010. (Applied Optimization). ISBN 9783540928287. Disponível em: <https://books.google.com.br/books?id=b2Wyj3RgQEIC>.

APÊNDICE A – MAPEAMENTO SISTEMÁTICO SOBRE SOLUÇÕES DE RECUPERAÇÃO DE DESASTRES

Este apêndice apresenta um resumo com alguns dos resultados extraídos a partir do estudo de mapeamento sistemático realizado em Mendonça et al. (2019a). De acordo com Kitchenham e Charters (2007), estudos de mapeamento sistemático "*são projetados para fornecer uma visão geral de uma área de pesquisa, para estabelecer se a evidência de pesquisa existe em um tópico e fornecer uma indicação da quantidade de evidência*". Assim, objetivando obter uma visão geral sobre a área de DR foi executado o estudo de mapeamento sistemático por Mendonça et al. (2019a).

O estudo identificou 567 potenciais através de buscas em quatro bases científicas: ACM Digital Library[1], IEEEXplore[2], ScienceDirect - Elservier[3] e SpringerLink[4]. Seguindo critérios estabelecidos, foram selecionados 49 trabalhos para serem analisados. Após a análise dos 49 trabalhos selecionados, foram extraídos dados para mapear as diferentes características dos trabalhos. Aqui são apresentados algumas das informações relevantes encontradas. A primeira, foi a identificação das razões para se adotar uma solução de DR. A Tabela 52 resume as principais razões citadas nos trabalhos para se adotar uma solução de DR.

Tabela 52 – Razões mencionadas para adoção de soluções de DR.

Motivo	**Quantidade**
Previnir a perda de dados	24 (48.97%)
Manter sistemas ativos	15 (30.61%)
Continuidade do negócio	13 (26.53%)
Evitar perdas financeiras	9 (18.36%)
Aumentar disponibilidade	8 (16.32%)
Prover segurança	4 (8.16%)
Prover integridade dos sistemas	3 (6.12%)
Evitar danos (em geral)	3 (5.88%)
Necessidade do negócio	2 (4.08%)
Proteger reputação da marca	1 (2.04%)
Previnir perda de clientes	1 (2.04%)
Prover recuperação de dados	1 (2.04%)

Analisando os dados extraídos dos estudos selecionados, foi possível identificar as estratégias de DR mais utilizadas. Ao total, foram identificadas nove diferentes estratégias

[1] <https://dl.acm.org/>
[2] <https://ieeexplore.ieee.org/>
[3] <http://www.sciencedirect.com/>
[4] <https://link.springer.com/>

analisadas nos trabalhos. A Figura 40 apresenta as estratégias analisadas ou discutidas nos estudos selecionados, bem como o número de estudos usados por cada estratégia. Vale ressaltar que a proposição dos modelos e a avaliação das estratégias de DR adotadas nesta tese foi baseada no resultado obtido desse estudo, onde foram adotadas as estratégias de replicação de ambiente, *backup* e replicação de BD.

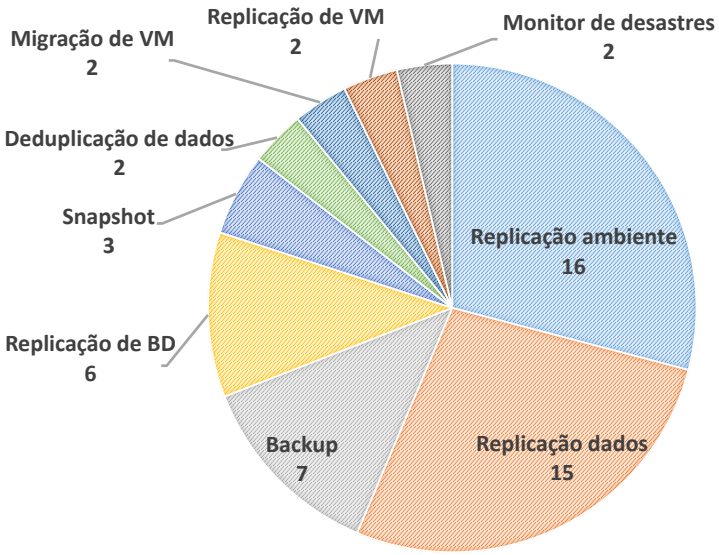


Figura 40 – Estratégias tratadas nos estudos selecionados.

Também foram identificadas as principais abordagens que são empregadas para analisar estratégias de DR. A Figura 41 resume as abordagens mais adotadas pelos estudos selecionados para analisar ou testar novas as estratégias de DR.

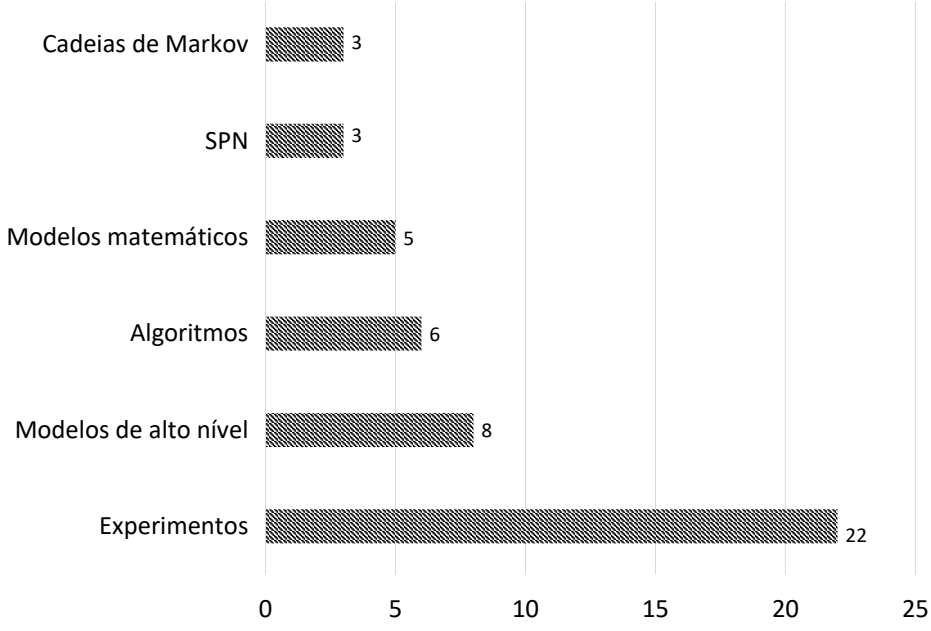


Figura 41 – Abordagens mais utilizadas para analisar estratégias de DR.

Além disso, o estudo também identificou as métricas mais adotadas para analisar estratégias de DR apresentadas ou discutidas nos estudos primários. Foram identificadas 14 métricas diferentes para consideras nos estudos selecionados para avaliar estratégias de DR. A Tabela 53 apresenta estas métricas e respectiva a quantidade de trabalhos que à adotou.

Tabela 53 – Métricas usadas pelos estudos selecionais.

Métrica	**Quantidade**
RTO	17(34.69%)
RPO	16(32.65%)
Custo	10(20.41%)
Disponibilidade	9(18.37%)
Downtime	5(10.2%)
Taxas de E/S	4(8.16%)
Tempo de resposta	3(6.12%)
Uso da CPU	2(4.08%)
Sobrecarga da rede	2(4.08%)
Taxa de transferência de dados	1(2.04%)
Confiabilidade	1(2.04%)
Requisições perdidas	1(2.04%)
TCO	1(2.04%)
TPO	1(2.04%)

APÊNDICE B – DEMONSTRAÇÃO DOS CÁLCULOS DE CUSTO

B.1 CÁLCULO DE CUSTO PARA O ESTUDO DE CASO 1

Abaixo é demonstrado o cálculo de custo do estudo de caso 1.

Custo primeiro ano = $COP + CM + \sum_{i=1}^{1} PoI_i$,

onde,

$COP = (LSR + LEP) \times UA \times T$

$COP = 479 \times 0,00279455 \times 8760 = 11.729,82$

e,

$$PoI = \sum_{i=1}^{1}(CA_i + CO_i) \tag{B.1}$$

$CA_1 = 0$

$CO_1 = (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,99817684 + 0,2 \times (1 - 0,99817684)) \times 8760) = 193,40$

e,

$CM = 0,00279545 \times 150 \times 8760 = 3.673,22$

logo,

Custo primeiro ano = $11.729,82 + 3.673,22 = 15.596,44$

B.2 CÁLCULO DE CUSTO PARA O ESTUDO DE CASO 2

Abaixo é demonstrado o cálculo de custo do estudo de caso 2.

$$
\begin{aligned}
&\text{Custo primeiro ano} = COP + CM + \sum_{i=1}^{2} PoI_i, \\
&\text{onde,} \\
&COP = (LSR + LEP) \times UA \times T \\
&COP = 479 \times 0,00229326 \times 8760 = 9.622,61 \\
&\text{e,} \\
&PoI = \sum_{i=1}^{2} (CA_i + CO_i) \\
&CA_1 = 0 \\
&CA_2 = 1.424,88 \\
&CO_1 = (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,99817684 + 0,2 \times (1 - 0,99817684)) \times 8760) \\
&CO_2 = (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,99885974 + 0,2 \times (1 - 0,99885974)) \times 8760) = 386,90 \\
&\text{e,} \\
&CM = 0,00229326 \times 150 \times 8760 = 3.013,34 \\
&\text{logo,} \\
&\text{Custo primeiro ano} = 9.622,61 + 1.424,88 + 386,90 + 3.013,34 = 14.447,73
\end{aligned}
\tag{B.2}
$$

B.3 CÁLCULO DE CUSTO PARA O ESTUDO DE CASO 3

Abaixo é demonstrado o cálculo de custo do estudo de caso 3. A Equação B.3 detalha o cálculo do custo quando é utilizada a configuração assíncrona. Em seguida, a Equação B.4 detalha o cálculo do custo quando é adotada a configuração semisíncrona.

$$
\begin{aligned}
&\text{Custo primeiro ano} = COP + CM + \sum_{i=1}^{2} PoI_i, \\
&\text{onde,} \\
&COP = (LSR + LEP) \times UA \times T \\
&COP = 479 \times 0,00156653 \times 8760 = 6.573,22 \\
&\text{e,} \\
&PoI = \sum_{i=1}^{2} (CA_i + CO_i) \\
&CA_1 = 0 \\
&CA_2 = 1.424,88 \\
&CO_1 = (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,998859749 + 0,2 \times (1 - 0,998859749)) \times 8760) = 193,5 \\
&CO_2 = (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,998859739 + 0,2 \times (1 - 0,998859739)) \times 8760) = 193,5 \\
&\text{e,} \\
&CM = 0,00156653 \times 150 \times 8760 = 2.058,42 \\
&\text{logo,} \\
&\text{Custo primeiro ano} = 6.573,22 + 1.424,88 + 193,50 + 193,50 + 2.058,42 = 10.443,54
\end{aligned}
\tag{B.3}
$$

$$
\begin{aligned}
&\text{Custo primeiro ano} = COP + CM + \sum_{i=1}^{2} PoI_i, \\
&\text{onde,} \\
&COP = (LSR + LEP) \times UA \times T \\
&COP = 479 \times 0,00158317 \times 8760 = 6.643,04 \\
&\text{e,} \\
&PoI = \sum_{i=1}^{2} (CA_i + CO_i) \\
&CA_1 = 0 \\
&CA_2 = 1.424,88 \\
&CO_1 = (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,998859749 + 0,2 \times (1 - 0,998859749)) \times 8760) = 193,50 \\
&CO_2 = (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,998859739 + 0,2 \times (1 - 0,998859739)) \times 8760) = 193,50 \\
&\text{e,} \\
&CM = 0,00158317 \times 150 \times 8760 = 2.080,29 \\
&\text{logo,} \\
&\text{Custo primeiro ano} = 6.643,04 + 1.424,88 + 193,50 + 193,50 + 2.080,29 = 10.535,22
\end{aligned}
\tag{B.4}
$$

B.4 CÁLCULO DE CUSTO PARA O ESTUDO DE CASO 4

Abaixo é demonstrado o cálculo de custo do estudo de caso 4. A Equação B.5 detalha o cálculo do custo quando é utilizada a configuração *hot-standby*. Em seguida, a Equação B.6 detalha o cálculo do custo quando é adotada a configuração *warm-standby*.

$$
\begin{aligned}
&\text{Custo primeiro ano} = COP + CM + \sum_{i=1}^{3} PoI_i, \\
&\text{onde,} \\
&COP = (LSR + LEP) \times UA \times T \\
&COP = 479 \times 0,00003841 \times 8760 = 161,18 \\
&\text{e,} \\
&PoI = \sum_{i=1}^{3} (CA_i + CO_i) \\
&CA_1 = 0 \\
&CA_2 = 1.424,88 \\
&CA_3 = 1.424,88 \\
&CO_1 = (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,998600657 + 0,2 \times (1 - 0,998600657)) \times 8760) = 193,47 \\
&CO_2 = (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,998991221 + 0,2 \times (1 - 0,998991221)) \times 8760) = 193,53 \\
&CO_3 = (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,998538312 + 0,2 \times (1 - 0,998538312)) \times 8760) = 193,45 \\
&\text{e,} \\
&CM = 0,00003841 \times 150 \times 8760 = 50,47 \\
&\text{logo,} \\
&\text{Custo primeiro ano} = 161,18 + 2.849,76 + 580,45 + 50,48 = 3641,87
\end{aligned}
\tag{B.5}
$$

Custo primeiro ano = $COP + CM + \sum_{i=1}^{4} PoI_i$,

onde,

$COP = (LSR + LEP) \times UA \times T$

$COP = 479 \times 0,00013005 \times 8760 = 545,70$

e,

$$
\begin{aligned}
PoI &= \sum_{i=1}^{3}(CA_i + CO_i) \\
CA_1 &= 0 \\
CA_2 &= 1.424,88 \\
CA_3 &= 1.424,88 \\
CO_1 &= (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,998964134 + 0,2 \times (1 - 0,998964134)) \times 8760) = 193,52 \\
CO_2 &= (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,499571278 + 0,2 \times (1 - 0,499571278)) \times 8760) = 116,14 \\
CO_3 &= (0,165 \times \sum_{i=1}^{1} 0,134 \times (0,998802029 + 0,2 \times (1 - 0,998802029)) \times 8760) = 193,49
\end{aligned}
\tag{B.6}
$$

e,

$CM = 0,00013005 \times 150 \times 8760 = 170,89$

logo,

Custo primeiro ano = $545,70 + 2.849,76 + 503,16 + 170,89 = 4.069,51$

APÊNDICE C – RESULTADOS DA ANÁLISE MULTI-CENÁRIO UTILIZANDO A ESTRATÉGIA DE *BACKUP*

C.1 RESULTADOS DA ANÁLISE MULTI-CENÁRIO

A Tabela 54 apresenta os resultados detalhados das avaliações executadas em diferentes cenários (ver Seção 6.2.4) que adotam o *backup* como estratégia de DR.

Tabela 54 – Resultados dos cenários avaliados utilizando a estratégia de *Backup*.

Cenário	B_i	B_s	Disponibilidade	Indisponibilidade por ano (h)	RPO (h)	RTO (h)	Custo (US$)
B1	12	100	0.997694099441588	20.1996888916887	12.2882045508513	16.9413901189813	14517.3923416079
B2	12	200	0.997563012959583	21.3480064740525	13.994270744499	19.0261214467334	15239.6841009147
B3	12	300	0.997391056548757	22.8543446328883	16.4211634505093	21.9231577647567	16187.1708028225
B4	12	400	0.997233437884943	24.2350841278992	19.1110715268483	23.6558517469889	17055.6559451843
B5	24	100	0.997724780092627	19.9309263885875	24.1268988356677	16.4685371090161	14447.7300000000
B6	24	200	0.997579291181397	21.2054092509624	24.5962950656113	19.4112489740804	15149.990447591
B7	24	300	0.997437766874011	22.4451621836638	25.9527233666663	21.6714872043937	15929.7950422602
B8	24	400	0.997248751517711	24.1009367048519	27.9026872111462	24.23944420898	16971.2772160875
B9	48	100	0.997713931457796	20.0259604297073	48.2725094333108	16.4719393572828	14408.1171390216
B10	48	200	0.997569855221538	21.2880682593275	48.2300504557882	19.0699489971084	15201.9829638527
B11	48	300	0.997423640804365	22.5689065537629	48.4786463914738	21.8442704316597	16007.6302510526
B12	48	400	0.997254419437277	24.0512857294535	49.1393414618976	23.8862296199171	16940.046752562

APÊNDICE D – SCRIPTS PARA OS EXPERIMENTOS DE INJEÇÃO DE FALHAS DO ESTUDO DE CASO 1

Este apêndice apresenta os scripts que foram utilizados para realização dos experimentos de injeção de falhas descritos na Subseção 6.1.4. A Listagem D.1 exibe o script que foi executado pelo injetor de falhas. As Listagens D.2, D.3, D.4, D.5 apresentam os scripts executados no monitor de desastres para verificar a operacionalidade dos servidores web 1 e 2, do balanceador de carga e do servidor de BD1, respectivamente.

Listing D.1 – Script para o injetor de falhas.

```
[General Parameters]

agents:7
[endGeneral]

[agent1]

IP:10.0.0.70
MAC:12345
sshUser:root
sshPass:*******
DistF:Exponential
Param1F:s-95544
DistR:Exponential
Param1R:s-45

Software

SoftwareCommandFailure:/usr/tmp/shutdown-vm-web1.sh&
SoftwareCommandRepair:/usr/tmp/start-vm-web1.sh&

[endAgent1]

[agent2]

IP:10.0.0.70
MAC:12345
sshUser:root
sshPass:*******
DistF:Exponential
Param1F:s-315360
DistR:Exponential
Param1R:s-18

Software

SoftwareCommandFailure:/usr/tmp/shutdown-vm-lb.sh&
SoftwareCommandRepair:/usr/tmp/start-vm-lb.sh&

[endAgent2]

```

```
[agent3]

IP:10.0.0.70
MAC:12345
sshUser:root
sshPass:*******
DistF:Exponential
Param1F:s-95544
DistR:Exponential
Param1R:s-45

Software

SoftwareCommandFailure:/usr/tmp/shutdown-vm-web2.sh&
SoftwareCommandRepair:/usr/tmp/start-vm-web2.sh&

[endAgent3]

[agent4]

IP:10.0.0.70
MAC:12345
sshUser:root
sshPass:*******
DistF:Exponential
Param1F:s-95544
DistR:Exponential
Param1R:s-45

Software

SoftwareCommandFailure:/usr/tmp/shutdown-vm-db1.sh&
SoftwareCommandRepair:/usr/tmp/start-vm-db1.sh&

[endAgent4]

[agent5]

IP:10.0.0.70
MAC:12345
sshUser:root
sshPass:*******
DistF:Exponential
Param1F:s-315360
DistR:Exponential
Param1R:s-144

Software

SoftwareCommandFailure:/usr/tmp/dc-failure.sh&
SoftwareCommandRepair:/usr/tmp/dc-repair-fail.sh&

[endAgent5]

[agent6]

IP:10.0.0.70
```

```
MAC:12345
sshUser:root
sshPass:*******
DistF:Exponential
Param1F:s-630720
DistR:Exponential
Param1R:s-864

Software

SoftwareCommandFailure:/usr/tmp/dc-disaster.sh&
SoftwareCommandRepair:/usr/tmp/dc-recovery.sh&

[endAgent6]

[agent7]

IP:10.0.0.70
MAC:12345
sshUser:root
sshPass:*******
DistF:Exponential
Param1F:s-360000
DistR:Exponential
Param1R:s-36

Software

SoftwareCommandFailure:/usr/tmp/nt-failure.sh&
SoftwareCommandRepair:/usr/tmp/nt-repair.sh&

[endAgent7]
```

Listing D.2 – Script executado no monitor de desastres para monitorar a VM do servidor web 1.

```
#!/bin/bash
#Author: Julio Mendonca
#Script que verifica se o servidor WEB1 esta UP.


SERVER="10.0.0.82"
outfile="/home/host02/Documentos/logs/monitor-dc1-web1-exp1.txt"
touch $outfile

while [ True ]
do
dt=$(date '+%d/%m/%Y %H:%M:%S')
ping -q -c 1 -W 1 $SERVER > /dev/null


if [ $? -eq 0 ]
then
    echo $dt" WEB1_Up">>$outfile
    sleep 1
else
    echo $dt" WEB1_Down">>$outfile
fi
```

```
done
```

Listing D.3 – Script executado no monitor de desastres para monitorar a VM do servidor web 2.

```
#!/bin/bash
#Author: Julio Mendonca
#Script que verifica se o servidor WEB2 esta UP.


SERVER="10.0.0.83"
outfile="/home/host02/Documentos/logs/monitor-dc1-web2-exp1.txt"
touch $outfile

while [ True ]
do
dt=$(date '+%d/%m/%Y %H:%M:%S')
ping -q -c 1 -W 1 $SERVER > /dev/null

if [ $? -eq 0 ]
then
    echo $dt" WEB2_Up">>$outfile
    sleep 1
else
    echo $dt" WEB2_Down">>$outfile
fi

done
```

Listing D.4 – Script executado no monitor de desastres para monitorar a VM do balanceador de carga.

```
#!/bin/bash
#Author: Julio Mendonca
#Script que verifica se o servidor LB1 esta UP.


SERVER="10.0.0.81"
outfile="/home/host02/Documentos/logs/monitor-dc1-lb1-exp1.txt"
touch $outfile

while [ True ]
do
dt=$(date '+%d/%m/%Y %H:%M:%S')
ping -q -c 1 -W 1 $SERVER > /dev/null


if [ $? -eq 0 ]
then
    echo $dt" LB1_Up">>$outfile
    sleep 1
else
    echo $dt" LB1_Down">>$outfile
fi
done
```

Listing D.5 – Script executado no monitor de desastres para monitorar a VM do BD1.

```
#!/bin/bash
#Author: Julio Mendonca
#Script que verifica se o servidor DB1 esta UP.


SERVER="10.0.0.84"
outfile="/home/host02/Documentos/logs/monitor-dc1-db1-exp1.txt"
touch $outfile

while [ True ]
do
dt=$(date '+%d/%m/%Y %H:%M:%S')
ping -q -c 1 -W 1 $SERVER > /dev/null


if [ $? -eq 0 ]
then
    echo $dt" DB1_Up">>$outfile
    sleep 1
else
    echo $dt" DB1_Down">>$outfile
fi
done
```